UNIVERSIDAD AUTÓNOMA DE MADRID

Balancing Flexibility and Robustness in Machine Learning: Semi-parametric Methods and Sparse Linear Models

by
José Miguel Hernández-Lobato

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Escuela Politécnica Superior
Computer Science Department

under the supervision of Alberto Suárez González

November 2010

“When you make the finding yourself — even if you’re the last person on Earth to see the light — you’ll never forget it.”

Carl Sagan

Abstract

Machine learning problems can be addressed by a variety of methods that span a wide range of degrees of flexibility and robustness. In the process of building a model for data, flexibility and robustness are desirable but often conflicting goals. On one side of the spectrum, parametric methods are very robust, in the sense that they are resilient to noise and are not generally misled by spurious regularities, which may be present in the data only by accident. However, their expressive capacity is limited. On the other side, non-parametric methods are very flexible and can in principle learn arbitrarily complex patterns when sufficient amounts of data are available for induction. However, as a result of this high flexibility, they are also more prone to overfitting. In practice, selecting the optimal method to address a specific learning task involves attaining the appropriate balance between flexibility and robustness.

There are some learning problems for which this balance cannot be attained using standard parametric or purely non-parametric approaches in isolation. Semi-parametric methods include both parametric and non-parametric components in the models assumed. The parametric part provides a robust description of some of the patterns in the data. The non-parametric component endows the model with the flexibility necessary to capture additional complex patterns. In this thesis, we analyze several problems in which semi-parametric methods provide accurate models for the data. The first one is the modeling of financial time series. The trends in these series are described by parametric models. The density of the innovations is directly learned from the data in a non-parametric manner. To improve the quality of the approximation, the estimation of the density of the innovations is performed in a transformed space, where the density of the transformed data is close to a Gaussian. A second problem involves developing semi-parametric models to describe arbitrary non-linear dependencies between two random variables. Bivariate Archimedean copulas are re-parameterized in terms of a unidimensional latent function that can be readily approximated using a basis of natural cubic splines. These splines are especially well suited to model the asymptotic tail dependence of the data.

In some learning problems even simple parametric methods are not sufficiently robust to provide accurate descriptions for the data. This investigation also addresses the specific question of how to improve the robustness of linear models by assuming sparsity in the model coefficients. In a Bayesian approach, sparsity can be favored by using specific priors, such as the spike and slab distribution. The advantage of the spike and slab prior is its superior selective shrinkage capacity: Some coefficients (those whose posterior has a large contribution from the spike) are forced to be small, while others (those in which the slab is the predominant contribution to the posterior) are not regularized. In this thesis, linear models with spike and slab priors are used to address problems with a high-dimensional feature space and small number of available training instances. Approximate inference is implemented using Expectation propagation (EP). For the sparse linear regression model, EP is a computationally efficient alternative to MCMC methods, which are asymptotically exact, but often require lengthy computations to converge. Another contribution is the design of a sparse Bayesian classifier for classification problems in which prior information about feature dependencies is available. Finally, a sparse linear model that makes use of a hierarchical spike and slab prior is applied to the problem of identifying transcriptional regulators from gene expression time series.

The semi-parametric methods and the sparse linear models analyzed in this thesis represent configurations of flexibility and robustness that cannot be attained by either standard parametric methods or by fully non-parametric approaches alone. Therefore, the proposed methods fill in some of the gaps left by these standard learning paradigms in the flexibility-robustness spectrum.

Resumen

Los problemas de aprendizaje automático se pueden tratar utilizando una diversidad de métodos que abarcan un amplio rango de grados de flexibilidad y robustez. En la construcción de un modelo para los datos, la flexibilidad y la robustez son objetivos deseables pero a menudo opuestos. A un lado del espectro, los métodos paramétricos son muy robustos, en el sentido de que son resistentes al ruido y que generalmente no les afectan las regularidades espurias que puedan econtrarse en los datos sólo por casualidad. Sin embargo, su capacidad expresiva es limitada. Por otro lado, los métodos no paramétricos son muy flexibles y pueden, en principio, aprender patrones arbitrariamente complejos siempre y cuando se disponga de suficientes datos para la inducción. No obstante, su alta flexibilidad implica que también son más propensos al sobreajuste. En la práctica, la elección del método óptimo para resolver un problema particular de aprendizaje implica alcanzar un equilibrio adecuado entre flexibilidad y robustez.

Existen algunos problemas en los que dicho equilibrio no puede ser alcanzado utilizando sólo enfoques paramétricos o no paramétricos aisladamente. Los métodos semiparamétricos incluyen componentes tanto paramétricas como no paramétricas en los modelos utilizados. La parte paramétrica proporciona una descripción robusta de parte de los patrones en los datos. La componente no paramétrica proporciona flexibilidad para capturar otras regularidades complejas adicionales. En esta tesis se analizan varios problemas en los que los métodos semiparamétricos proporcionan modelos certeros para los datos. El primero es el modelado de series temporales financieras. Las tendencias en estas series se describen paramétricamente. La densidad de las innovaciones se aprende directamente a partir de los datos de un modo no paramétrico. La calidad de la aproximación se mejora realizando la estimación de la densidad de las innovaciones en un espacio transformado, donde la densidad de los datos transformados se parece a una Gaussiana. Un segundo problema trata el desarrollo de modelos semiparamétricos para describir dependencias no lineales arbitrarias entre dos variables aleatorias. Las cópulas Archimedeanas bivariadas se reparametrizan en términos de una función latente unidimensional que se aproxima fácilmente utilizando una base de splines naturales cúbicos. Estos splines son especialmente adecuados para modelar la dependencia asintótica de cola en los datos.

En algunos problemas de aprendizaje incluso los modelos paramétricos más simples no son suficientemente robustos como para proporcionar una descripción certera de los datos. Esta investigación también trata sobre cómo mejorar la robustez de los modelos lineales al asumir dispersidad en los coeficientes del modelo. Bajo un enfoque Bayesiano, dicha dispersidad se favorece utilizando priors específicos, como la distribución de punta y losa. La ventaja del prior de punta y losa es su alto encogimiento selectivo: algunos coeficientes (aquellos en los que la distribución posterior tiene una alta contribución de la punta) se fuerzan a que sean pequeños, mientras que otros (aquellos en los que la losa representa la contribución predominante en la distribución posterior) no son regularizados. En esta tesis, los modelos lineales con priors de punta y losa se utilizan para tratar problemas con un espacio de atributos de dimension alta y un número pequeño de ejemplos de entrenamiento disponibles. La inferencia aproximada se implementa utilizando propagación de expectaciones (PE). En el modelo de regresión disperso y lineal, PE es una alternativa computacionalmente eficiente frente a los métodos MCMC, que son asintóticamente exactos, pero a menudo requieren largos cómputos para converger. Otra contribución es el diseño de un clasificador Bayesiano disperso para problemas de clasificación en los que existe información *a priori* sobre las dependencias entre atributos. Por último, un modelo disperso y lineal basado en un prior jerárquico de punta y losa se utiliza para identificar genes reguladores a partir de series temporales de expresión genética.

Los métodos semiparamétricos y los modelos lineales y dispersos analizados en esta tesis presentan configuraciones de flexibilidad y robustez que no pueden ser alcanzadas ni por métodos paramétricos estándar, ni por enfoques completamente no paramétricos aisladamente. De este modo, los métodos propuestos rellenan algunos de los huecos dejados por estos paradigmas de aprendizaje estándar en el espectro de flexibilidad y robustez.

Acknowledgements

I would like to thank everybody who made this thesis possible. I am very grateful to my supervisor, Alberto Suárez González, for introducing me into the field of machine learning and for his instructive guiding during all these years. I certainly appreciate the freedom he has given me to satisfy my curiosity about many different research topics. Furthermore, his comments and discussions have been not only instructive, but also very helpful.

I express my gratitude to Tom Heskes for giving me the opportunity to join the machine learning group of the Institute for Computing and Information Sciences at Radboud University in Nijmegen for a three-month period in 2006. It is during this stay when I became interested in linear Bayesian models with spike and slab priors and first heard of the algorithm expectation propagation. I am also very grateful to Tjeerd M. H. Dijkstra who introduced me to the fields of Bioinformatics and computational systems biology, his comments about the malaria parasite were valuable and very instructive. I would also like to thank Botond Cseke for his help and friendship during my stay in Nijmegen.

I am grateful to my brother, Daniel Hernández-Lobato, for joining me in the adventure of graduate education, his collaboration has been invaluable for the completion of this dissertation. I would also like to show my deepest gratitude to my parents, Manuel Hernández Bailén and María del Carmen Lobato Gómez, for their support, endless patience and encouragement when it was most required. I appreciate very much their constant love.

I would especially like to thank Cristina Bogdanschi and Alexandra Dumitrescu for their priceless friendship during all these years, you are really great friends! Cristina, you will always have a special place in my heart! I will never forget the time we spent together.

Finally, I would like to thank all the members of the B-408 laboratory, the secretaries of the computer science department, the staff from the cafeteria at the psychology school, and all my really good friends in Collado Villalba.

Contents

Abbreviations

AMISE	**A**symptotic **M**ean **I**ntegrated **E**rror
ARCH	**A**uto**R**egressive **C**onditional **H**eteroskedasticity
BMG	**B**ayesian **M**ixture of **G**aussians copula method
CD	**C**ritical **D**istance
CLA	**CLA**yton parametric Archimedean copula
CML	**C**anonical **M**aximum **L**ikelihood
DIM	The Archimedean copula estimator designed by **DIM**itrova et al.
DMPLE	**D**iscrete **M**aximum **P**enalized **L**ikelihood
DREAM	**D**ialog for **R**everse **E**ngineering **A**ssessments and **M**ethods
EP	**E**xpectation **P**ropagation
ES	**E**xpected **S**hortfall
Exc	**Exc**eedances
FRA	**FRA**nk parametric Archimedean copula
GARCH	**G**eneralized **A**uto**R**egressive **C**onditional **H**eteroskedasticity
GC	The parametric **G**aussian **C**opula
GeD	**GE**ometrically **D**esigned
GHYP	**G**eneralized **HY**perbolic.
GK	Copula estimation method based on **G**aussian **K**ernels.
GL	**G**raph **L**asso
GUM	**GUM**bel parametric Archimedean copula
IG	**I**nverse **G**amma
ISE	**I**ntegrated **S**quared **E**rror
KL	**K**ullback **L**eibler
LAM	The Archimedean coupula estimator designed by **LAM**bert.
Laplace-EP	Linear regression model with **Laplace** prior and **EP**.
LP	**L**inear **P**rogramming
LRMSSP	**L**inear **R**egression **M**odel with **S**pike and **S**lab **P**rior
MCMC	**M**arkov **C**hain **M**onte **C**arlo
MDH	**M**ixture of **D**istributions **H**ipothesis
MFST	**M**etastasis **F**ree **S**urvival **T**ime
MLE	**M**aximum **L**ikelihood **E**stimator
MRF	**M**arkov **R**andom **F**ield
MSE	**M**ean **S**quared **E**error
NBSBC	**N**etwork **B**ased **S**parse **B**ayesian **C**lassifier
NBSVM	**N**etwork **B**ased **S**upport **V**ector **M**achine
NIG	**N**ormal **I**nverse **G**aussian

PIN	**P**air **I**dentification **N**umber
PLL	**P**enalized **L**og **L**ikelihood
PR	**P**recision **R**ecall
ROC	**R**eceiver **O**perating **C**haracteristic
RV	**R**ealized **V**olatilities
RVM	**R**elevance **V**ector **M**achine
SBC	**S**parse **B**ayesian **C**lassifier
SD	**S**tandard **D**eviation
SNP	**S**emi **N**on-**P**arametric
SPAC	**S**emi-**P**arametric bivariate **A**rchimedean **C**opula estimator
SPE	**S**emi-**P**arametric **E**stimator
SS-EP	Linear regression model with **S**pike and **S**lab prior and **EP**.
SS-MCMC	Linear regression model with **S**pike and **S**lab prior and **MCMC**.
SST	**S**kewed **S**tudent's **T** parametric copula
ST	**S**tudent's **T** parametric copula
SVM	**S**upport **Vector** **M**achine
TF	**T**ranscription **F**actor
VaR	**V**alue **at** **R**isk
WMON	**W**orld **M**eteorological **O**rganization **N**umber

To my family

Chapter 1

Introduction

Machine learning is an area of computer science concerned with the problem of how to design computational systems that can discover regularities in empirical data in an automatic manner (Bishop, 2006). The process by which a computational system is configured so that it has the capability to identify these regularity patterns is called *learning*. In supervised learning, the system learns by automatic induction from a set of labeled examples, the training data. When the amount of data available for learning is increased, these systems are expected to identify the underlying patterns more accurately. Therefore, learning systems are said to improve with experience (Mitchell, 1997), which is represented in the form of observed data. A learning method is the process by which a learning system is built. To make automatic induction possible, most learning methods assume that the data are generated by a model of a particular type (for example, a generalized linear model, a neural network or a decision tree) that depends on a set of parameters. This model restricts the regularities that can be identified by the learning system to a subset of candidate patterns. Each of these candidate patterns corresponds to a different value of the model parameters. The value of these parameters is determined by fitting the model to the data available for learning. The ultimate goal is to design systems with good generalization capacity. That is, systems that correctly identify patterns in data instances not seen before. Such systems can be used to draw accurate conclusions and make intelligent decisions about new situations not necessarily encountered in the learning stage. Once an estimate of the model parameters is obtained, the performance of the learning system is evaluated using a *test set*. This test set is composed of additional data instances, which are independent of the ones included in the training set. The performance in the test set is used as a proxy for the actual generalization capacity of the learning system.

The generalization performance of a learning system strongly depends on the complexity of the model assumed (Hastie et al., 2001). If the model is too simple, the system can only capture the actual data regularities in a rough manner. In this case, the system has poor generalization properties and is said to suffer from *underfitting*. By contrast, when the model is too complex, the system can identify accidental patterns in the training data that need not be present in the test set. These spurious patterns can be the result of random fluctuations or of measurement errors during the data collection process. In this case, the generalization capacity of the learning system is also poor. The learning system is said to be affected by *overfitting*.

Most machine learning methods can be classified according to the complexity of the models that they assume as being either *parametric* (less complex) or *non-parametric* (more complex) (Alpaydin, 2004). Parametric methods describe the data using models that depend on a small number of parameters (Wasserman, 2003). These methods make definite assumptions on the functional form of these models. If the types of patterns that can be represented by such form do not correspond to the actual patterns present in the data, these methods can be severely affected by underfitting. No matter how many training data are available for learning, the parametric methods will perform poorly in these cases. An advantage of parametric methods is that they are unlikely to suffer from overfitting. The reason for this is that the patterns that can be described by standard parametric models are generally simple. By contrast, spurious patterns, which are only present by accident in the data, tend to have complex forms. This is the idea behind the principle of Occam's razor for avoiding overfitting: simpler models are preferred if more complex models do not significantly improve the quality of the description for the observations (Domingos, 1999; Duda et al., 2001). Therefore, parametric methods are said to be *robust*. However, their expressive capacity is limited. They do not have the *flexibility* required to learn complex patterns without making strong assumptions about the process that generated the data. Some examples of parametric methods are standard linear regression and classification models (Bishop, 2006), the Kalman filter (Kalman, 1960), Markov random field models (Kindermann and Snell, 1980), ARMA and GARCH processes for the analysis of time series (Brockwell and Davis, 1996) and generally, any method based on a model which is specified using a reduced number of simple mathematical functions (linear, exponential or sinusoidal, for example) and standard parametric probability distributions (Gaussian, Student's t or Bernoulli, for example) (Alpaydin, 2004).

Non-parametric methods make as few assumptions as possible about the characteristics of the data (Wasserman, 2006). In particular, these methods generally assume only that the data exhibit some smoothness. This lack of strong assumptions means that non-parametric methods have the potentiality to learn arbitrarily complex patterns with as much precision as desired, provided that sufficient amounts of training data are available. For this, the data are described using models in which the number of parameters is not fixed beforehand. This number increases with the size of the training set, depending on the complexity of the patterns observed. However, the capacity of non-parametric methods to learn complex patterns makes them more prone to overfitting. Therefore, in general terms, non-parametric methods are more *flexible* but less *robust* than parametric approaches. Some examples of non-parametric methods are Gaussian processes (Rasmussen and Williams, 2005), neural networks (Bishop, 1996), support vector machines (Vapnik, 1995), decision trees (Breiman et al., 1984) and kernel methods for density estimation (Silverman, 1986), among others.

A simple example with simulated data is illustrative of the trade-off between flexibility and robustness in machine learning. Consider the data $\mathcal{D} = \{(x_i, y_i) : i = 1, \ldots, n\}$, where $n = 15$ and each x_i is generated from a uniform distribution in the unit interval $[0,1]$. The corresponding value for y_i is sampled conditioning to the value of x_i according to

$$y_i = f_0(x_i) + e_i = 8x_i^3 - 12x_i^2 + 6x_i - 1 + e_i\,, \tag{1.1}$$

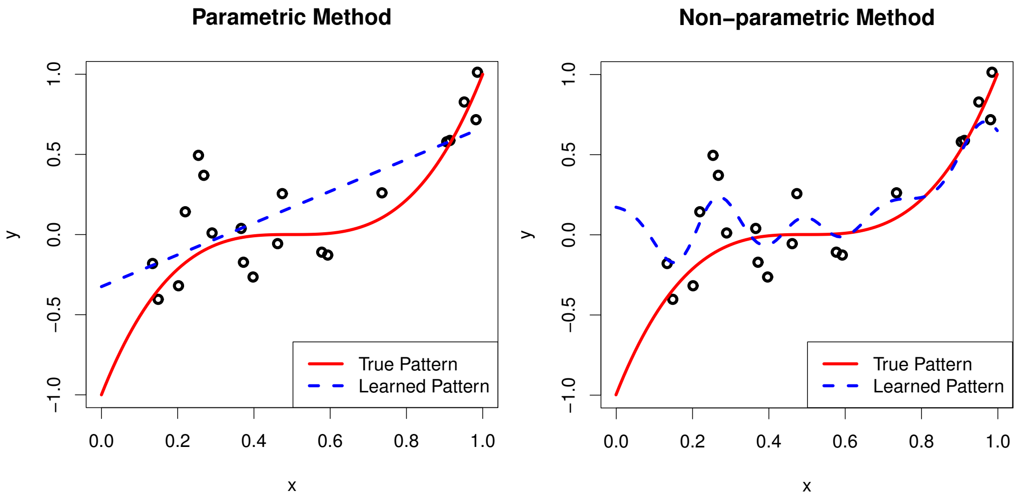

Figure 1.1: Results obtained by the parametric (left) and non-parametric (right) methods on a specific realization of $\mathcal{D}$. Each training instance is displayed using a small black circle. The patterns identified by each learning method are represented by a discontinuous blue line (parametric) or curve (non-parametric). The true pattern f_0 is represented as a continuous red curve.

where e_i is an additive Gaussian noise with zero mean and standard deviation equal to 10^{-1}. The goal of a learning method is to correctly model f_0, the component that describes the regularities in the data, using only the training instances in $\mathcal{D}$. For this task, we consider two machine learning methods. The first one is parametric. It assumes a linear model whose parameters are adjusted by minimizing the average squared prediction error in the training set. The second method is non-parametric. It assumes that the data have been generated by a Gaussian process with a squared exponential covariance function (that is, a Gaussian kernel). The width of this covariance function is estimated on the training set by by 10-fold cross validation. The plots in Figure 1.1 compare the patterns identified by the parametric (left) and the non-parametric method (right) with the actual pattern f_0 in a specific realization of the problem.

The flexibility of the linear model (parametric) is clearly not sufficient to provide an accurate approximation of f_0. In fact, the quality of the fit would not improve even if more training data were available. This problem is especially severe in the regions $0 \leq x \leq 0.1$ and $0.55 \leq x \leq 0.85$. By contrast, the non-parametric model is more flexible and provides a fairly accurate approximation of f_0 in the interval $0.55 \leq x \leq 0.85$. However, the model is not robust. In particular, it predicts a small bump in the region $0.2 \leq x \leq 0.3$, which is only the result of sample fluctuations. Note that the parametric method produces a more accurate fit in this region because the linear dependence which has been assumed is too simple to capture the spurious pattern given by the bump. Finally, the extrapolation given by the non-parametric model in the region $x \leq 0.1$ is also very poor. In particular, the slope of the predicted pattern is negative in that region while the actual f_0 has a large positive slope.

1.1 Balancing Flexibility and Robustness

Flexibility and robustness are often conflicting goals. Consequently, one cannot be improved without deteriorating the other. In practice, selecting the optimal method for a particular learning problem involves achieving a balance between flexibility and robustness. This balance is specific to the problem under analysis. On the one hand, the method should be sufficiently flexible to capture the actual patterns in the data. On the other hand, the method should also be robust, so that it is not misled by spurious patterns which are only observed in the data by accident. This trade-off is discussed in the machine learning literature in terms of the bias/variance dilemma (Geman et al., 1992). The bias measures the alignment of the method with the analyzed problem. Methods having a low bias are well suited to model the particular problem considered. High bias corresponds to poor alignment. The variance measures the specificity of this alignment. High variance denotes a nonspecific alignment (Duda et al., 2001). Flexible methods tend to have high variance and low bias, while robust methods usually have high bias and low variance.

The bounds on the generalization error of classification methods derived in the Vapnik-Chervonenkis theory (Vapnik, 1995) are also illustrative of the compromise between flexibility and robustness in machine learning. These bounds are based on the *probably approximately correct* (PAC) framework (Valiant, 1984). Their value depends on the VC dimension, h, of the family of hypothesis considered (Cherkassky and Mulier, 1998; Vapnik, 1995). Flexible models have high values of h, while robust ones have low values of h. The optimal (lowest possible) value of the bound is obtained by models whose VC dimension is optimally tuned to the classification problem under analysis. Within this framework, these models achieve the best possible trade-off between flexibility and robustness. Illustrations of this compromise also emerge in other approaches to model selection in machine learning such as *minimum description length* MDL, the *Bayesian information criterion* BIC or the Bayesian framework for model selection (Bishop, 2006; Hastie et al., 2001; MacKay, 1992).

Parametric methods are adequate when the size of the training set is not large and the noise in the data acquisition process is moderate, when we have prior knowledge about the types of patterns that are expected in the data, or when these patterns have a simple form. In these cases, the robustness of parametric methods makes up for for their lack of flexibility. By contrast, non-parametric methods are very successful when large amounts of training instances are available for learning and there is no or little knowledge about the specific form of the patterns present in the data. In this situation, the larger flexibility of non-parametric methods compensates for their lack of robustness. Nonetheless, there are some learning tasks in which the appropriate balance between flexibility and robustness cannot be attained by either standard parametric methods or by fully non-parametric approaches in isolation. For example, in some learning problems, the size of the training set is sufficiently large so that a method that is more flexible than standard parametric approaches should be used to avoid underfitting. However, a fully non-parametric description of the underlying patterns may result in severe overfitting problems. These problems can be addressed using *semi-parametric* methods (Gagliardini and Gourieroux, 2007; Gallant and Nychka, 1987; Kosorok, 2009), which combine the robustness of parametric approaches and the flexibility of non-parametric methods. Another case is when the number n of training instances is very small, but the dimensionality of the data d is large ($d \gg n$). In this setting, even the simplest parametric models may be too flexible and can lead to significant overfitting

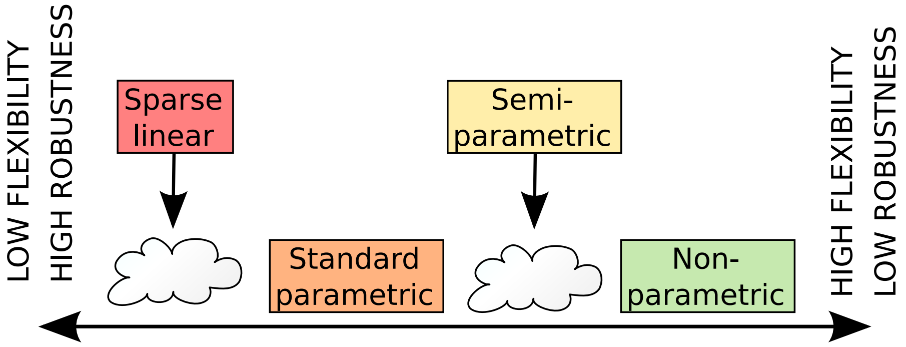

Figure 1.2: Arrangement of the different machine learning methods in terms of their flexibility and robustness. Semi-parametric methods and approaches based on sparse linear models fill in the gaps located on the left of non-parametric and standard parametric methods.

problems. To avoid these problems, we need to improve the robustness of the model at the expense of reducing its flexibility. When the model considered is linear, this can be achieved by assuming that the parameter vector of the model is *sparse* (Johnstone and Titterington, 2009; Seeger, 2008). In sparse linear models, only a few coefficients are expected to take values significantly different from zero.

Figure 1.2 presents a diagram in which different learning methods are arranged according to their specific balance of robustness and flexibility. The methods located on the left part of the diagram are very robust but lack flexibility. Methods located on the right part are very flexible but lack robustness. The diagram illustrates how semi-parametric techniques and approaches based on sparse linear models fill in some of the gaps left by standard parametric and non-parametric methods in the flexibility-robustness spectrum.

1.1.1 Semi-parametric Methods

Semi-parametric methods (Kosorok, 2009) integrate parametric and non-parametric components in the model assumed for the data. The parametric parts are generally used to describe the most salient patterns in the data. The non-parametric components are used to improve the flexibility of the model. These can be used to either refine the description given by the parametric part or to capture other types of regularities that cannot be represented by standard parametric forms. This means that semi-parametric methods are more robust but also less expressive (that is, less flexible) than fully non-parametric approaches.

Semi-parametric methods have been applied to a wide range of problems. For example, the SNP (Semi Non-Parametric) method described by Gallant and Nychka (1987) can be used to model arbitrary probability density functions. In SNP, the density function is approximated using an expansion in Hermite polynomials, which have appealing computational properties (Fenton and Gallant, 1996). SNP has been used to build semi-parametric models of financial time series (Gallant et al., 1997). Hoti and Holmström (2004) present a semi-parametric method for density estimation which is applied to classification problems. This technique combines non-parametric kernel density estimates (Silverman, 1986) with parametric Gaussian distributions. Another example are the semi-parametric regression methods reviewed by Härdle et al. (2004), which combine linear models with smooth non-linear functions. These techniques allow, for instance, to regress y on x_1 and x_2 using the semi-parametric model $\mathrm{E}[y|x_1,x_2] = \alpha + g_1(x_1) + g_2(x_2)$, where g_1 and g_2 are arbitrary non-linear smooth functions. Semi-parametric methods have also been proposed for modeling non-linear dependencies between two random variables. For example, an Archimedean two-dimensional *copula* (Nelsen, 2006) can be described using a single one-dimensional functional parameter that is approximated in a non-parametric manner (Dimitrova et al., 2008; Gagliardini and Gourieroux, 2007; Lambert, 2007). Finally, Bayesian approaches that use a Dirichlet process prior (Blei and Jordan, 2006; Bush and MacEachern, 1996; Ghosh et al., 2010) can also be regarded as semi-parametric methods. The Dirichlet process avoids the specification of parametric prior distributions, which are usually unknown. Additionally, it induces some clustering which is often very useful to identify data instances which share similar characteristics (Blei and Jordan, 2006; Ghosh et al., 2010).

In this thesis, semi-parametric methods are used to address two learning problems in which standard parametric approaches are frequently misspecified and fully non-parametric methods are likely to be affected by overfitting. The first of these problems is the modeling of time series of price variations in financial assets. The second problem involves modeling the dependence structure of two random variables using semi-parametric *copulas*. The following paragraphs describe the contributions of this thesis in the field of semi-parametric methods:

1. Chapter 2 presents a new semi-parametric method for the estimation of financial time-series models of price variations in which the unknown distribution of the innovations (that is, the differences between the observed time-series values and their optimal forecast in terms of past values) is approximated using kernels (Hernández-Lobato et al., 2007). In financial time series, the actual innovations are heavy-tailed (Bollerslev, 1987; Ferenstein and Gasowski, 2004). Standard kernel methods fail to generate a smooth approximation of the density of extreme events when the data are heavy-tailed. To improve the quality of the approximation, the kernel estimation is performed in a transformed space, in which the density of the innovations is close to a Gaussian (Wand et al., 1991). The proposed semi-parametric method is based on an iterative algorithm that alternates between the estimation of the model parameters and the approximation of the innovation density in a non-parametric manner. Experiments with simulated and empirical data show that this method generates accurate estimates of the model parameters and of the density of the innovations (especially at the tails), outperforming other parametric and semi-parametric methods (Forsberg and Bollerslev, 2002; Gallant et al., 1997; Panorska et al., 1995).

2. A novel semi-parametric bivariate Archimedean copula method (SPAC) is proposed in Chapter 3 (Hernández-Lobato and Suárez, 2009). This approach is based on the family of Archimedean copulas (Genest and Rivest, 1993; Nelsen, 2006). Archimedean two-dimensional copulas are specified in terms of a unique one-dimensional function called the Archimedean *generator*. SPAC assumes a non-parametric form for the generator in terms of an auxiliary latent function, which is modeled using a basis of natural cubic splines (de Boor, 1978). This new latent function is especially well suited to modeling

tail dependence (Joe, 1997). Experiments on simulated, financial and precipitation data are performed to compare SPAC with other methods for copula estimation, including parametric Student's t and Gaussian copula models (Malevergne and Sornette, 2006), parametric Archimedean copulas (Genest and Rivest, 1993), other alternative methods for the flexible estimation of Archimedean copulas (Dimitrova et al., 2008; Lambert, 2007), a copula model based on Bayesian mixtures of Gaussians (Attias, 1999; Bishop, 2006) and a fully non-parametric copula method (Fermanian and Scaillet, 2003). The good overall results of SPAC in these experiments are explained by its ability to learn asymmetric dependence structures while limiting the amount of overfitting.

1.1.2 Sparse Linear Models

Sparse linear models assume that the data have been generated by a linear model in which the vector of coefficients is *sparse*. In these models, a small number of coefficients take values that are significantly different from zero. The remaining coefficients are exactly zero. Assuming *sparsity* is a powerful regularization strategy that increases the robustness of the linear model at the expense of reducing its flexibility (Seeger, 2008). This specific balance between flexibility and robustness is very useful to address learning problems with a small number n of training instances and a high-dimensional feature space of dimension d (Johnstone and Titterington, 2009). These types of problems arise in disciplines such as the statistical processing of natural language (Sandler et al., 2008), the analysis of gene expression data (Dudoit and Fridlyand, 2003) or the modeling of fMRI data (van Gerven et al., 2009). The assumption of sparsity is more restrictive than other regularization approaches, such as *ridge* regularization or *weight decay* (Hastie et al., 2001), which only force the model coefficients to be uniformly small, but not necessarily zero. These alternative regularization approaches are less efficient at reducing the random fluctuations generated by irrelevant features. Consequently, they often have poorer performance in the large d and small n scenario (Seeger, 2008).

The simplest way to obtain a sparse linear model is to select a small subset of components of the data (features) and then, learn a standard non-sparse linear model on this reduced subset of features (Miller, 2002). The subsets of features are often identified using greedy methods that include or exclude features according to a specific scoring metric. Because subset selection methods are discrete processes (features are either included or excluded), small changes in the training set can result in the selection of very different subsets of features (Hastie et al., 2001; Tibshirani, 1996). Additionally, learning a standard linear model on the selected features with no further constraints on the model coefficients can yield a predictive model with high variance (low robustness). The next paragraph describes methods for the construction of sparse linear models which are often more stable.

Sparse linear models can be generated using specific penalty terms in the objective function which is minimized to fit the model. The form and strength of these penalties is such that in the minimizer of the penalized objective function includes many coefficients that are exactly zero. Some examples are the lasso (Tibshirani, 1996), whose penalty is proportional to the ℓ_1 norm of the vector of model coefficients, the elastic net (Zou and Hastie, 2005), which employs a penalty proportional to a linear combination of the ℓ_1 and ℓ_2 norms of the coefficient vector or for instance, the F_∞-norm support vector machine proposed by Zou and Yuan (2008), whose penalty term is proportional to the sum of the ℓ_∞ norms of different groups of model parameters. In Bayesian methods (Bishop, 2006), sparsity can be favored by using certain types of prior distributions (Seeger, 2008). Sparsity enforcing priors are probability densities which are peaked at zero and have large probability mass for a wide range of values significantly different from zero. This particular structure induces a bi-separation in the posterior distribution between a reduced number of coefficients whose posterior probability of being different from zero is large and many coefficients which have very small posterior means. Ishwaran and Rao (2005) call this bi-separation effect *selective shrinkage*. Some examples of sparsity enforcing priors are the Laplace (Seeger, 2008), the spike and slab (George and McCulloch, 1997) or the degenerate Student's t (Tipping, 2001) distributions.

This thesis describes the successful application of sparse linear models to several important problems with large d and small n. The set of analyzed problems includes regression tasks, classification problems in which prior information about feature dependencies is available and the problem of identifying regulatory elements in genetic networks. Using a Bayesian approach, sparsity in the linear models is enforced by assuming spike and slab priors for the coefficients of the models. The following paragraphs describe the contributions of this thesis in the field of sparse linear models:

1. Chapter 4 describes a linear regression model, in which sparsity is favored by using spike and slab priors on the coefficients of the model. In these types of models, Bayesian inference is usually performed using Gibbs sampling (George and McCulloch, 1997). This method guarantees the asymptotic convergence to the exact solution of the inference problem. However, the computational costs are often excessively high. As a more efficient alternative, we propose to use the expectation propagation algorithm (EP) (Minka, 2001). The performance of EP is evaluated in different regression tasks: the reverse-engineering of transcription networks, the reconstruction of sparse signals and the prediction of user sentiment. Even though the solution given by EP is only an approximation, its accuracy in the analyzed problems is better or comparable to Gibbs sampling. Furthermore, it has a lower computational cost. The method that assumes spike and slab priors and uses EP for approximate inference also outperforms methods that assume other sparsifying priors, such as the Laplace (Seeger, 2008) and the degenerate Student's t priors (Tipping, 2001). The good performance of the spike and slab model can be ascribed to its superior selective shrinkage capacity.

2. A new network-based sparse Bayesian classifier (NBSBC) is introduced in Chapter 5 (Hernández-Lobato et al., 2010b). This model assumes a spike and slab prior which is combined with a Markov random field (Wei and Li, 2007) to incorporate information about feature dependencies in the model. This information is encoded by a network whose nodes represent features and whose edges connect dependent features. EP is used for approximate inference (Minka, 2001). The performance of NBSBC is evaluated on four classification problems in which prior information about feature dependencies is available. NBSBC improves in these experiments the results of a classifier based on the graph lasso (Jacob et al., 2009), the network-based support vector machine (Zhu et al., 2009), the standard support vector machine (Vapnik, 1995) and a version of NBSBC that ignores

the network of features. An important factor in the good overall results of NBSBC is the superior robustness of this method.

3. In Chapter 6 a sparse hierarchical Bayesian model is used for the discovery of genetic regulators using only time series of gene expression data (Hernández-Lobato et al., 2008). The hierarchy incorporates the prior knowledge that only a reduced number of regulators control the expression of many other genes (Alon, 2006; Hernández-Lobato et al., 2010a). This is implemented using a spike and slab prior, in which the weights of the mixture are assumed to follow a hierarchical Bernoulli model. Also in this case EP is used for efficient approximate inference (Minka, 2001). Applying the method to gene expression data from the malaria parasite (Llinás et al., 2006), we found, among the top ten genes that were identified as likely regulators, four genes with significant homology (in terms of BLASTP hits) to transcription factors in an amoeba, one RNA regulator and three genes of unknown function.

1.2 Publications

This section lists in anti-chronological order the articles published until the completion of this thesis. The publications are classified in three categories. The category *Direct Work* lists documents that have been accepted for publication and correspond to investigation projects described in this thesis. The category *Related Work* includes published work related to this thesis but not described in detail in this document. Manuscripts that have been submitted for publication and are currently under review appear under the heading category *Submitted Work*

Direct Work

- Hernández-Lobato J. M., Hernández-Lobato D. and Suárez A. (2010).
  Network-based Sparse Bayesian Classification
  Pattern Recognition, In Press.

- Hernández-Lobato J. M. and Suárez A. (2009)
  Modeling Dependence with Semiparametric Archimedean Copulas
  In International Workshop on Advances in Machine Learning for Computational Finance, AMLCF 2009.

- Hernández-Lobato J. M., Dijkstra T. and Heskes T. (2008).
  Regulator Discovery from Gene Expression Time Series of Malaria Parasites:
  a Hierarchical Approach
  In Advances in Neural Information Processing Systems 20, Vancouver, British Columbia, Canada, December 3-6, NIPS 2007

- Hernández-Lobato J. M., Hernández-Lobato D. and Suárez A. (2007).
  GARCH Processes with Non-parametric Innovations for Market Risk Estimation
  In International Conference on Artificial Neural Networks, Porto, Portugal, ICANN 2007, Part I, LNCS 4668, pp. 718-727

Related Work

- Hernández-Lobato D., Hernández-Lobato J. M., Helleppute T. and Dupont P. (2010)
  Expectation Propagation for Bayesian Multi-task Feature Selection
  In European Conference on Machine Learning, Barcelona, Spain, ECML PKDD 2010, Part I, LNAI 6321, pp. 522-537

- Hernández-Lobato J. M. and Dijkstra T. (2010)
  Hub Gene Selection Methods for the Reconstruction of Transcription Networks
  In European Conference on Machine Learning, Barcelona, Spain, ECML PKDD 2010, Part I, LNAI 6321, pp. 506-521, 2010

- Hernández-Lobato D., Hernández-Lobato J. M. and Suárez A. (2010)
  Expectation Propagation for Microarray Data Classification
  Pattern Recognition Letters, Volume 31, Issue, 12, pp. 1618-1626

- Hernández-Lobato D. and Hernández-Lobato J. M. (2008)
  Bayes Machines for Binary Classification
  Pattern Recognition Letters, Volume 29, Issue 10, pp. 1466-1473

- Hernández-Lobato D., Hernández-Lobato J. M., Ruiz-Torrubiano R. and Ángel V. (2006)
  Pruning Adaptive Boosting Ensembles by Means of a Genetic Algorithm
  In 7th Intelligent Data Engineering and Automated Learning, Burgos, Spain, IDEAL 2006, LNCS 4224, pp. 995-1002

- Hernández-Lobato J. M. and Suárez A. (2006)
  Competitive and Collaborative Mixtures of Experts for Financial Risk Analysis
  In International Conference on Artificial Neural Networks, Athens, Greece, ICANN 2006, Part II, LNCS 4132, pp. 691-700

Submitted Work

- Hernández-Lobato J. M. and Suárez A.
  Semiparametric Bivariate Archimedean Copulas

Chapter 2

Semi-parametric Models for Financial Time-series

A semi-parametric method is introduced for the estimation of models of financial time series in which the specific form of the density of the innovations is unknown. In financial time series, the actual innovations are heavy-tailed. For this reason, standard kernel density estimators fail to provide a smooth approximation of the density in the tails of the distribution. The quality of the approximation can be improved by performing the kernel estimation in a transformed space, where the distribution of the innovations is close to a Gaussian. The proposed semi-parametric estimator (SPE) is computed using an iterative algorithm that alternates between the estimation of the model parameters and the non-parametric approximation of the density of the innovations. The performance of SPE is assessed in experiments with simulated and empirical financial time series. These time series are assumed to follow an asymmetric GARCH process with unknown innovations, whose distribution is estimated non-parametrically. SPE provides accurate estimates for both the model parameters and the conditional density of asset returns. The improvements obtained are especially significant in the tails of the distribution, which is the region of interest in financial risk analysis.

2.1 Introduction

Time series of price variations in financial markets are very unpredictable. This absence of systematic trends can be understood if markets are assumed to be efficient (Fama, 1970). In an efficient market, all the information about the value of a firm available to the trading agents is reflected in the current stock price. Therefore, variations in price have their origin in new unexpected information that becomes known. To reflect this behavior, the evolution of market prices can be described by stochastic time-series processes with additive random innovations. The innovations are defined as the differences between the current values of the time series and their optimal forecast in terms of past observations. The first stochastic model for financial time series was proposed by Bachelier (1900). Bachelier's model assumes that the innovations in the price process are independent and identically distributed random variables (iidrv) with a

Gaussian density. This model has the drawback that it assigns non-zero probabilities to negative prices. Closer to the empirical behavior of financial time series is the assumption that changes in the price are proportional to price levels. This observation is incorporated in the model of Osborne (1959) and Samuelson (1965), which describes the increments in the logarithm of the price, also referred to as *returns*, as Gaussian iidrv. Given a series of price values $\{P_t\}_{t=0}^n$, the corresponding series of returns $\{Y_t\}_{t=1}^n$ is obtained as

$$Y_t = 100 \log \frac{P_t}{P_{t-1}}, \qquad t = 1, \ldots, n, \tag{2.1}$$

where the factor 100 is included to represent the return in %. Mandelbrot (1963) noticed that the empirical distributions of financial returns are more peaked and have heavier tails than the Gaussian. For this reason, he proposed the family of stable distributions (Nolan, 2002) to model the innovations in the return process. Other alternatives have also been suggested, such as the Student's t distribution (Praetz, 1972) or the mixture of Gaussians model (Kon, 1984).

Another characteristic of asset returns is that they are heteroskedastic; that is, the volatility (standard deviation) of the returns exhibits a time-dependent structure: Large returns (either positive or negative) are often followed by returns that are also large. As a result, these time series present periods of low volatility and periods of high volatility. This effect is frequently called in the literature *volatility clustering* (Cont, 2001). Engle (1982) and Bollerslev (1986) introduced ARCH and GARCH models to capture the heteroskedasticity of financial returns. Additionally, these processes can account for the appearance of heavy tails in the unconditional distribution of returns even when the underlying innovations are Gaussian (Mikosch and Starica, 2000). However, ARCH and GARCH models with Gaussian innovations can only partially explain the leptokurtosis observed in the empirical distribution of returns: When these models are fitted to financial data, the corresponding series of residuals still exhibit heavy tails, even after they have been scaled using the time-dependent volatility given by the model. A possible improvement of these models is to assume that the innovations follow a heavy-tailed parametric distribution. For example, Bollerslev (1987) proposes the Student's t-distribution, Panorska et al. (1995) suggests the stable distribution and Forsberg and Bollerslev (2002) the normal inverse Gaussian distribution.

The models mentioned above assume a parametric form for the density of the innovations. The main advantage of the parametric approach is that the model parameters can be efficiently estimated by maximum likelihood. Furthermore, the risk of overfitting is reduced because of the reduced number of parameters used to characterize the innovations. Nevertheless, if the assumed functional form for the innovations significantly differs from the actual distribution, the predictions made by the model can be rather inaccurate. Another possibility is to resort to semi-parametric time-series models. These models are very flexible because they do not constrain the form of the distribution of the innovations. Instead, this form is directly learned from the data. The quality of the approximation improves as more data becomes available. However, this turns out to be a very challenging task. It is rather difficult to provide accurate non-parametric estimates of the density of extreme outcomes in a heavy-tailed sample because of the scarcity of data in the tails of the distribution. In this chapter, we introduce an estimation method that successfully addresses this difficulty and can be used to construct accurate semi-parametric time-series models for financial returns (Hernández-Lobato et al., 2007).

Several methods have been proposed in the literature for the estimation of financial time-series models when the distribution of the innovations is unknown. The method proposed by Engle and González-Rivera (1991) for the estimation of semi-parametric GARCH processes is very successful in most cases. However, it fails when the actual density of the innovations presents heavy tails. Another method for the estimation of time-series models with a flexible specification for the distribution of the innovations is SNP (semi non-parametric) (Fenton and Gallant, 1996; Gallant and Nychka, 1987). This method approximates the unknown density using an expansion in Hermite polynomials, which have appealing computational properties. The expansion coefficients and the model parameters can be readily estimated by maximum likelihood. However, the approximation given by SNP has exponentially decaying tails, which are not adequate for modeling the density of extreme financial returns. Drost et al. (1997) describe a method for the estimation of the parameters of a time-series model with unknown form for the innovations. This approach focuses only on the model parameters and does not produce accurate estimates of the innovation distribution. In addition to this, it requires to perform a reparameterization of the time-series model, an operation which may not be possible in all cases.

The main difference between the method presented here and previous approaches to the problem is the manner in which the density of the innovations is approximated. Specifically, the proposed method transforms the data before performing the density estimation. Frequently, non-parametric density estimation is implemented by placing kernels of fixed width on each of the points in the sample. However, when the data are heavy-tailed, this method fails to provide a smooth estimate of the density of extreme events. Following Wand et al. (1991), we perform the kernel density estimation in a transformed space where the data are well approximated by a Gaussian distribution. The transformation is based on a fit of a stable distribution (Nolan, 2002) to the original data by maximum likelihood.

The proposed semi-parametric estimator (SPE) is based on iterating the following steps: First, the parametric part of the model is estimated by maximizing a likelihood function. A second stage involves the non-parametric estimation of the density of the innovations using the transformation method described above. This sequence of steps is repeated until convergence. The performance of SPE is evaluated in a series of experiments in which asymmetric GARCH models are fitted to simulated and empirical data (Ding et al., 1993). These experiments show that the efficiency of SPE is very close to the maximum likelihood estimator, which makes use of the actual density of the innovations. SPE also generates a very accurate approximation of the conditional density of financial returns, particularly in the tails of the distribution. These features make SPE especially useful for solving important financial tasks such as the quantification of market risk, the selection of optimal investment portfolios and option pricing.

This chapter is organized as follows. Section 2.2 describes a method for density estimation using kernels in which the approximation of the density is performed in a transformed space and then transformed back into the original space. Subsections 2.2.1 and 2.2.2 analyze the performance of several back-transformed kernel methods for density estimation in experiments with synthetic and empirical data, respectively. Section 2.3 describes an iterative procedure for the estimation of semi-parametric financial time-series models. The performance of this novel semi-parametric method is assessed in subsections 2.3.1 and 2.3.2 using simulated and empirical data. Finally, Section 2.4 summarizes the results and conclusions of this investigation.

2.2 Kernel Density Estimates for Heavy-tailed Data

Kernel methods (Silverman, 1986) are a popular tool for non-parametric estimation of density functions. Given the sample $X_1,\ldots,X_n$, in which each X_i is distributed according to a density function f (that is, $X_i \sim f$) the kernel estimate of f is

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) . \tag{2.2}$$

The parameter h is the *bandwidth* of the kernel. It determines the amount of smoothing in the estimation. A common choice for the kernel function K is the standard Gaussian density. The consistency theorem of kernel density estimators formulated by Parzen (1962) states that if f is a bounded density, in the limit $n \to \infty$, if $nh \to \infty$ and $h \to 0$ then the kernel estimate $\hat{f}$ can approximate the actual f up to any degree of precision in terms of squared error. When both K and f are Gaussian, the value of h that minimizes the AMISE (Asymptotic Mean Integrated Squared Error) is $h_{opt} \approx 1.06 n^{-\frac{1}{5}}\sigma$, where σ is the standard deviation of f (Silverman, 1986). Since σ is generally unknown, it is usually replaced by the empirical standard deviation of the available sample. Although this prescription for the bandwidth parameter works reasonably well even if f is not Gaussian, Sheather (2004) recommends the plug-in method (Sheather and Jones, 1991) for arbitrary non-Gaussian densities.

The method described above for non-parametric density estimation does not perform well when the data $X_1,\ldots,X_n$ are heavy-tailed, as is typically the case for the innovations in time series of financial returns. The difficulties are particularly severe in the approximation of the probability density of extreme outcomes. The origin of this shortcoming is that samples from heavy-tailed distributions often include very few points in the tails. Standard kernel methods tend to assign very low probability to regions with sparse observations. This effect is illustrated in Figure 2.1. The graph on the left shows, in logarithmic scale, a kernel density estimate of the marginal distribution of the daily returns of IBM stocks. The innovations for this time series (whose exact values are unknown) are expected to follow a similar distribution. While the central part of the density function is reasonably well approximated, the kernel estimator does not provide an accurate description of the tails of the distribution. The bumps that appear in the tails correspond to Gaussian kernels centered at sparse extreme observations. The overlap among these kernels is not sufficient to provide a smooth approximation in this region.

To address this problem, Wand et al. (1991) propose to perform the approximation of the density in a transformed space, where using the same value for the smoothing parameter at all locations is expected to be more adequate. Assuming that the correct transformation is known and that $X_1,\ldots,X_n$ is a heavy-tailed sample with density f, then the estimate $\hat{f}$ of the density in the original space is given by the back-transformed kernel density estimator

$$\hat{f}(x) = |g'_\pi(x)| \frac{1}{n}\sum_{i=1}^{n} K_h\left(g_\pi(X_i) - g_\pi(x)\right) , \tag{2.3}$$

where $K_h(\cdot) = h^{-1}K(\cdot/h)$, g_π is the function that maps the data to the transformed space and π is a vector of parameters that specify g_π within a family of monotonic increasing transformations. By arguments similar to those given by Silverman (1986), if the smoothing parameter in the

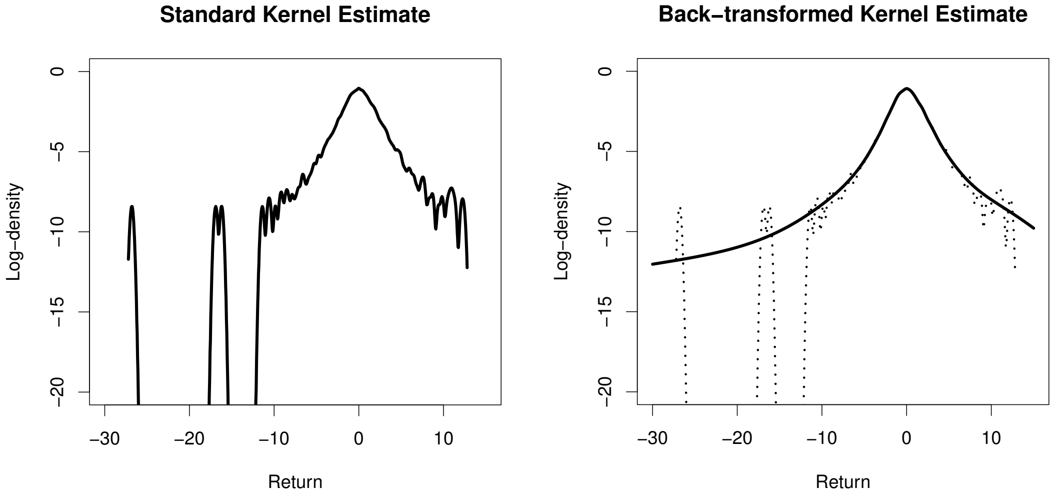

Figure 2.1: Left, logarithm of a standard kernel density estimate for the unconditional distribution of the daily returns of IBM stocks. The sample corresponds to the period from 1985/01/03 to 2008/03/31 and contains 11,665 points. The bandwidth of the kernels is fixed using the plug-in method (Sheather and Jones, 1991). Right, logarithm of the density estimate generated by a back-transformed kernel method for the same sample. A stable distribution is used in the transformation. The bandwidth parameter is in this case $1.06n^{-1/5}$, where $n = 11,665$.

transformed space, h, is optimally chosen the AMISE of the back-transformed kernel density estimator in the transformed space is proportional to

$$\left\{ \int f''_{g_\pi}(x)^2\, dx \right\}^{\frac{1}{5}}, \tag{2.4}$$

where f_{g_π} is the density of the sample in the transformed space, that is, $g_\pi(X_i) \sim f_{g_\pi}$. Thus, an appropriate transformation g_π is such that the transformed density f_{g_π} minimizes (2.4). Terrell (1990) shows that among all densities f with a given known variance, the density that minimizes $\int f''(x)^2\, dx$ is the Beta(4,4) density. Wand et al. (1991) make the observation that the Gaussian density comes very close to attaining this bound. Hence, the criterion used in practice to select the transformation g_π is that the distribution of the data in the transformed space should be close to Gaussian. The transformation $g_\pi(x) = \Phi^{-1}(F(x))$, where F is the cumulative distribution for f and Φ^{-1} is the standard Gaussian quantile function, satisfies $g_\pi(X_i) \sim \mathcal{N}(0,1)$. However, if we knew F then we would also know f because $f = F'$ and the estimation process would not be necessary. Nevertheless, it is possible to approximate F by a parametric distribution $\bar{F}_\pi$ that can describe heavy-tailed data. The parameters π of $\bar{F}_\pi$ can then be estimated from the sample $X_1, \ldots, X_n$, by maximum likelihood

$$\hat{\pi} = \arg\max_{\pi} \sum_{i=1}^{n} \log \bar{F}'_\pi(X_i)\,. \tag{2.5}$$

The final transformation is then $g_{\hat{\pi}}(x) = \Phi^{-1}(\bar{F}_{\hat{\pi}}(x))$. Note that if the actual distribution of the data were known, better results could be obtained by using a transformation based on the Beta(4,4) distribution instead of the standard Gaussian (Bolancé et al., 2008). However, since the true data distribution is never known in practice, the transformation based on the Gaussian quantile function should be accurate enough.

To apply the back-transformed kernel approach to a sample of financial innovations, we consider a family of parametric distributions $\bar{F}_\pi$ that has sufficient flexibility to account for the empirical properties of the marginal distribution of financial returns. Cont (2001) indicates that $\bar{F}_\pi$ should at least have a location parameter, a scale parameter, a parameter describing the decay of the tails and, finally, a parameter that allows each tail to have a different behavior. In this thesis, we consider three families of parametric distributions that fulfill these requirements: The family of normal inverse Gaussian distributions (Barndorff-Nielsen, 1997), the family of generalized hyperbolic distributions (Prause, 1999) and the family of stable distributions (Nolan, 2002). The graph on the right of Figure 2.1 displays a back-transformed kernel density estimate for the daily returns of IBM stocks. The transformation used to compute the density estimate is $g_{\hat{\pi}}(x) = \Phi^{-1}(\bar{F}_{\hat{\pi}}(x))$, where $\bar{F}_{\hat{\pi}}$ is the cumulative probability function of the stable distribution. Figure 2.1 clearly shows that the back-transformed kernel method considerably improves the accuracy of the density estimate, especially in the tails of the distribution.

Finally, another approach for density estimation which can be used to reduce the bumps in the tails of standard kernel methods is the adaptive kernel density estimator (Silverman, 1986). This method constructs a density estimate by placing kernels at the observed data points, but the width of each kernel is allowed to vary from one point to another. A detailed description of this method is given in Appendix A.1. The basic idea is to use broader kernels in regions of low density such as the tails. A consequence of this is that the resulting density estimate is smooth in the tails of the distribution. However, our experiments indicate that, in the analyzed datasets, back-transformed kernel methods often outperform adaptive methods when the actual distribution of the data is heavy-tailed (see the next section).

2.2.1 Experiments with Simulated Data

Experiments with simulated heavy-tailed data are carried out to investigate the improvements that can be achieved by performing the density estimation in a transformed space. In a first group of experiments, samples of size 1000, 2000 and 4000 are independently generated from a Student's t distribution with 5 degrees of freedom, zero mean and unit standard deviation. In a second group of experiments, the samples are generated from a normal inverse Gaussian distribution with parameters $\alpha = 2$, $\beta = -0.1$, $\delta = \alpha^{-2}(\alpha^2 - \beta^2)^{\frac{3}{2}}$, $\mu = -\delta\beta(\alpha^2 - \beta^2)^{-\frac{1}{2}}$, where δ and μ are chosen so that the distribution has zero mean and unit standard deviation. The non-zero value for β means that this distribution is slightly skewed, which is often the case in empirical financial data. Both the Student's t distribution and the NIG distribution exhibit heavy tails and are plausible models for the probability density of financial innovations.

For each sample generated, the density of the data is estimated in the original space with (i) the standard kernel method and (ii) the adaptive kernel method described in Appendix A.1. Subsequently, back-transformed kernel density estimates are computed in different transformed spaces. The transformations considered are based on (iii) normal inverse Gaussian (NIG), (iv) generalized hyperbolic (GHYP) and (vi) stable distributions. For the standard kernel method, the smoothing parameter is determined using the plug-in method of Sheather and Jones (1991). For the back-transformed kernel estimators, this parameter is set to $1.06n^{-\frac{1}{5}}$, where n is the size of

Table 2.1: Average square root of the ISE for the different kernel density estimation methods.

				Back-transformed		
Problem	**Size**	**Standard**	**Adaptive**	**NIG**	**GHYP**	**Stable**
NIG	1000	0.0341	0.0357	0.0323	0.0326	0.0329
	2000	0.0261	0.0276	0.0247	0.0246	0.0253
	4000	0.0199	0.0211	0.0188	0.0189	0.0192
Student's *t*	1000	0.0367	0.0384	0.0330	0.0325	0.0338
	2000	0.0280	0.0300	0.0250	0.0248	0.0257
	4000	0.0214	0.0228	0.0191	0.0191	0.0197

the training sample. The reason for using this rule in back-transformed kernel density estimators is that this bandwidth value would be optimal (in the sense that it minimizes the AMISE in the transformed space) if the transformed data were actually Gaussian. Appendix A.6 describes the computational details for the implementation of these experiments.

Table 2.1 shows the average square root of the integrated squared error (ISE) obtained by the different density estimation methods. The values reported in this table are averages over 1000 simulations. These results show that the back-transformed kernel methods are in all cases more accurate than the standard kernel approach, even when the parametric form of the distribution used in the transformation does not coincide with the actual distribution that was used to generate the data. Finally, the worst performing method is the adaptive kernel estimator, which obtains higher error values than the standard kernel method in all cases.

2.2.2 Experiments with Financial Data

The aim of this chapter is to construct models for heavy-tailed financial time series in which the density of the innovations is described in a non-parametric manner. Because the proposed approach is based on performing the estimation of the density in a transformed space, it is important to determine the impact of selecting a particular transformation function in the quality of the final estimator. Therefore, we investigate the performance of back-transformed kernel methods based on NIG, GHYP and stable transformations using actual financial data. The data used in these experiments consist of time series of 4000 consecutive daily returns from 59 assets included in the Dow Jones Composite Index[1]. The returns are computed using the daily closing prices adjusted for dividends and splits, as published in `http://www.finance.yahoo.com`. The time period considered is from June 3, 1992 to March 31, 2008. Each time series of returns $\{Y_t\}_{t=1}^{4000}$ is assumed to be generated by a stationary lag-one autoregressive process, in which the volatility is assumed to follow an asymmetric GARCH process (Ding et al., 1993)

$$\begin{aligned} Y_t &= \phi_0 + \phi_1 Y_{t-1} + \sigma_t e_t \\ \sigma_t &= \kappa + \alpha(|\sigma_{t-1}e_{t-1}| - \gamma\sigma_{t-1}e_{t-1}) + \beta\sigma_{t-1}\,, \end{aligned} \tag{2.6}$$

[1] AA, AEP, AES, AIG, ALEX, AMR, AXP, BA, BNI, C, CAT, CNP, CNW, CSX, D, DD, DIS, DUK, ED, EIX, EXC, EXPD, FDX, FPL, GE, GM, GMT, HD, HON, HPQ, IBM, INTC, JBHT, JNJ, JPM, KO, LUV, MCD, MMM, MO, MRK, MSFT, NI, NSC, OSG, PCG, PEG, PFE, PG, R, SO, T, UNP, UTX, VZ, WMB, WMT, XOM and YRCW.

where $\kappa > 0$, $\alpha \geq 0$, $\beta \geq 0$, $-1 < \gamma < 1$, $-1 < \phi_1 < 1$. The innovations $\{e_t\}$ are iidrv's sampled form a density f which has zero-mean and unit standard deviation. The dependence of σ_t on the absolute value of the lagged innovation reflects the fact that, in empirical financial data, the absolute value of the returns frequently exhibit higher autocorrelations than higher powers of the returns (Ding et al., 1993; Taylor, 1986). The dependence of σ_t on the lagged innovation reflects the empirical observation that volatility has an asymmetric response to past positive and negative shocks (Cont, 2001).

The parameters of the model are selected by maximizing the logarithm of the conditional likelihood (Brockwell and Davis, 1996), with the assumtion that f is standard Gaussian, that is, $e_t \sim \mathcal{N}(0,1)$. The assumption of Gaussian innovations is violated in practice. However, this estimation method (often referred to as quasi-maximum likelihood estimation) is generally consistent (Bollerslev and Wooldbridge, 1992). In the estimation process, σ_0 is assumed to be equal to the sample standard deviation of the series (denoted by $\hat{\sigma}$), e_0 is assumed to be 0 and finally, $\phi_0 + \phi_1 Y_0$ is taken to be equal to the sample mean of the series of returns (denoted by $\hat{\mu}$). Let $\hat{\boldsymbol{\theta}} = (\hat{\phi}_0, \hat{\phi}_1, \hat{\kappa}, \hat{\alpha}, \hat{\gamma}, \hat{\beta})$ be the estimate of the model parameters obtained after calibration of (2.6) on the series $\{Y_t\}_{t=1}^{4000}$. Then, the series of residuals of the process $\{r_t(\hat{\boldsymbol{\theta}})\}_{t=1}^{4000}$ is given by

$$r_t(\hat{\boldsymbol{\theta}}) = \begin{cases} Y_t - \hat{\phi}_1 Y_{t-1} - \hat{\phi}_0 & t \geq 2 \\ Y_t - \hat{\mu} & t = 1 \end{cases} . \tag{2.7}$$

These residuals should not present significant autocorrelations at lag 1. However, they may still be heteroskedastic. To eliminate the heteroskedasticity in the series of residuals, each $r_t(\hat{\boldsymbol{\theta}})$ is scaled by the corresponding estimate of the volatility $\hat{\sigma}_t(\hat{\boldsymbol{\theta}})$, where

$$\hat{\sigma}_t(\hat{\boldsymbol{\theta}}) = \begin{cases} \hat{\kappa} + \hat{\alpha}(|r_{t-1}(\hat{\boldsymbol{\theta}})| - \hat{\gamma} r_{t-1}(\hat{\boldsymbol{\theta}})) + \hat{\beta}\hat{\sigma}_{t-1}(\hat{\boldsymbol{\theta}}) & t \geq 2 \\ \hat{\kappa} + \hat{\beta}\hat{\sigma} & t = 1 \end{cases} . \tag{2.8}$$

In this manner, we obtain the series of scaled residuals $\{u_t(\hat{\boldsymbol{\theta}})\}_{t=1}^{4000}$ where $u_t(\hat{\boldsymbol{\theta}}) = r_t(\hat{\boldsymbol{\theta}})/\hat{\sigma}_t(\hat{\boldsymbol{\theta}})$. These scaled residuals should be an accurate approximation of the actual innovations in the series of returns.

The plot on the top left of Figure 2.2 displays the series of 4000 returns for the stock AES. The middle-left plot in this figure indicates that this series presents a very small, but significant, autocorrelation at lag 1. The autocorrelations for the absolute values of the returns are larger and remain positive for longer times, as displayed in the bottom-left plot of Figure 2.2. These strong autocorrelations are originated by the time-dependent structure of the volatility in the series of returns. The plot on the top right of Figure 2.2 displays the series of 4000 scaled residuals obtained after calibrating the time-series model (2.6) on the returns of AES. This series of scaled residuals exhibits no significant autocorrelations, as displayed by the middle-right plot in this figure. Finally, the autocorrelations for the absolute values of the series of scaled residuals are also very small (see the bottom-right plot in Figure 2.2). These properties confirm that the heteroskedasticity of the original series has been successfully eliminated and that the resulting scaled residuals are approximately independent.

Once the model described above has been calibrated for a particular asset, the corresponding 4000 scaled residuals are obtained and split on two consecutive blocks of 2000 elements. Then,

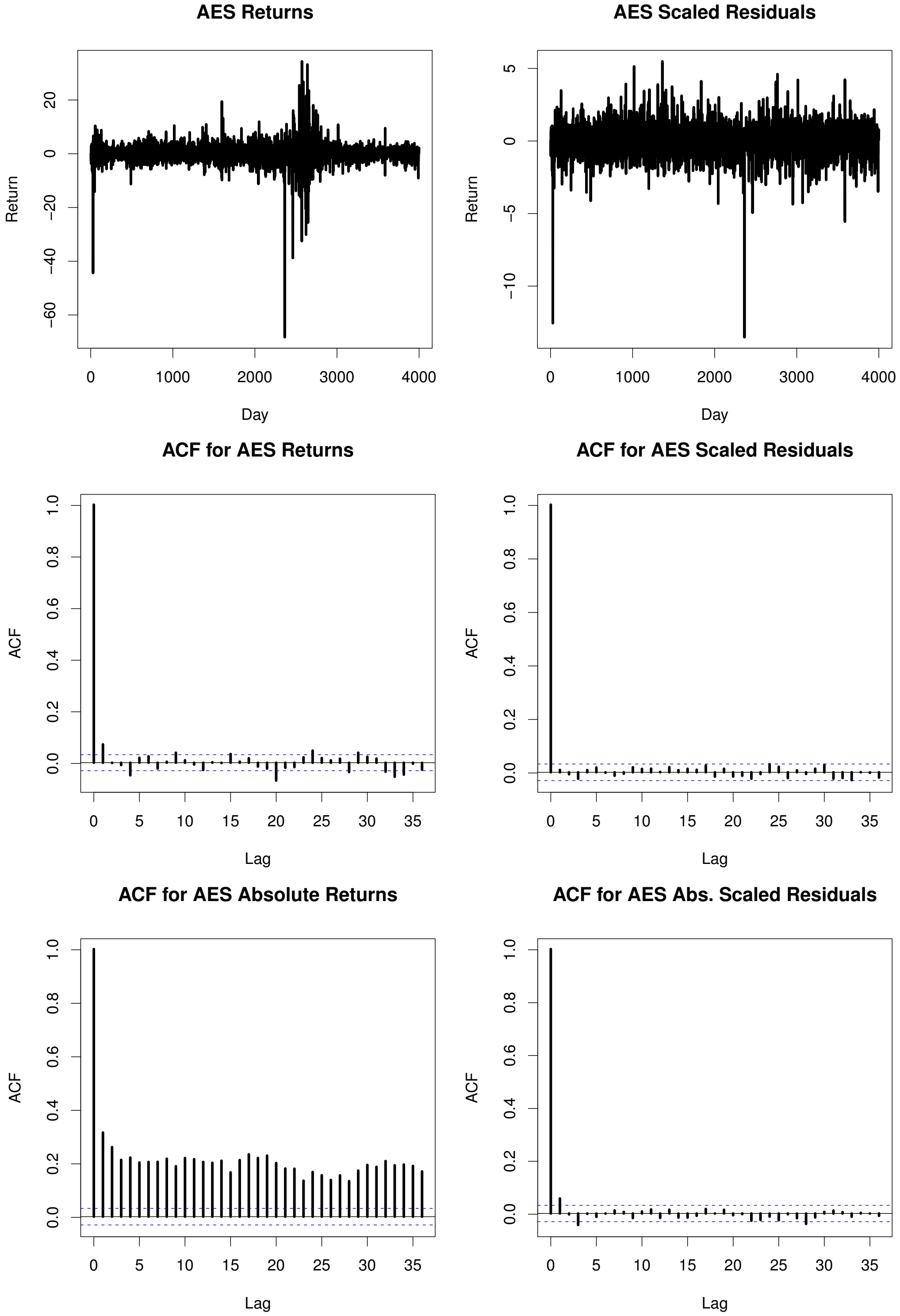

Figure 2.2: Top left, middle left and bottom left: AES returns, empirical autocorrelations for AES returns and empirical autocorrelations for the absolute value of AES returns, respectively. Top right, middle right and bottom right: corresponding plots for the scaled residuals obtained after filtering the returns of AES stocks with the time-series model given by (2.6).

Table 2.2: Average log-likelihood obtained by each back-transformed method on the 59 assets.

NIG	GHYP	Stable
-2777.543	-2776.339	-2762.829

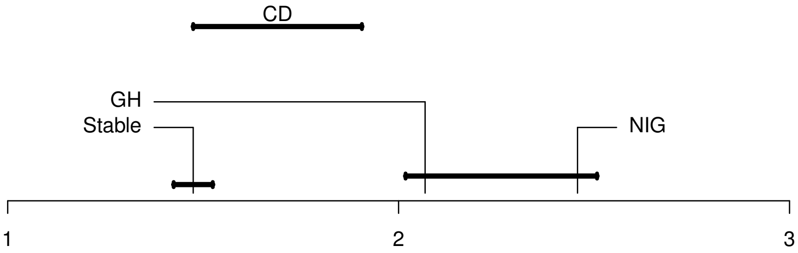

Figure 2.3: All to all comparison of the different back-transformed kernel density estimators by the Nemenyi test. The horizontal axis corresponds the average rank of each method on the 59 problems. Methods whose average ranks are not significantly different at the level $\alpha = 0.05$ appear connected in the figure.

the back-transformed kernel density estimate is constructed using the scaled residuals in the first block. The log-likelihood of this estimate is evaluated in the second block of data. This out-of-sample measure of performance should be unbiased. Table 2.2 displays the average value of the log-likelihood obtained by each estimation technique on the 59 financial assets. According to this performance measure, the best method is the kernel estimator in which the data are transformed using the stable distribution.

To determine whether the differences in performance are statistically significant, we follow the methodology proposed by Demšar (2006). This framework is designed to compare the predictive performance of different methods in a collection of problems: The different methods are ranked according to their performance in the collection of problems considered. Statistical tests are then used to determine whether the differences in average ranks are significant. In our case, a Friedman rank sum test rejects the hypothesis that all the methods have an equivalent performance in the 59 problems that have been analyzed (p-value $= 5.2 \cdot 10^{-7}$). The average ranks of the different estimators with a Nemenyi test at a 95% confidence level are shown in Figure 2.3. The differences in performance between methods whose average ranks differ less than a critical distance (CD) are not significant at this confidence level. The methods whose average ranks are not significantly different appear connected in the figure. These results confirm that the stable transformation is significantly better than the NIG or the GHYP transformations for approximating the conditional density of financial returns.

The only member of the stable family with finite variance is the Gaussian distribution. Other types of stable distributions have infinite variance. However, there is empirical evidence that the actual distributions of financial returns have finite second moments (Cont, 2001). This has often been used to discard the stable family as realistic model for the unconditional distribution of financial returns. Nevertheless, even though the stable density used in the transformation may have infinite variance, the resulting back-transformed kernel approximation has in most cases finite second moment. This result is proved in Appendix A.2.

2.3 Semi-parametric Models for Financial Time-series

In this section, we introduce a novel semi-parametric estimator (SPE) for the calibration of the model parameters and the density of the innovations in a stationary time-series model for the daily returns of financial assets. Consider the time series of financial returns $\{Y_t\}_{t=1}^n$. Assume that this series has been generated by the stationary process

$$Y_t = \mu(\mathcal{F}_{t-1};\boldsymbol{\theta}) + \sigma(\mathcal{F}_{t-1};\boldsymbol{\theta})e_t\,, \qquad t = 1,2,\ldots,n\,, \tag{2.9}$$

where $\boldsymbol{\theta}$ is a d-dimensional vector of parameters, f is the (unknown) density function for the innovations ($e_t \sim f(e)$), which has zero mean and unit standard deviation, $\mathcal{F}_{t-1}$ is a filtration, e_t is independent of $\mathcal{F}_{t-1}$ and $\mu(\mathcal{F}_{t-1};\boldsymbol{\theta})$ and $\sigma(\mathcal{F}_{t-1};\boldsymbol{\theta})$ are known functions which determine the location and the scale of Y_t in terms of $\boldsymbol{\theta}$ and $\mathcal{F}_{t-1}$. The scaled residuals of the model $\{u_t\}_{t=1}^n$ can be computed as a function of $\boldsymbol{\theta}$ using

$$u_t(\boldsymbol{\theta}) = \frac{Y_t - \mu(\mathcal{F}_{t-1};\boldsymbol{\theta})}{\sigma(\mathcal{F}_{t-1};\boldsymbol{\theta})} \tag{2.10}$$

and the corresponding conditional scaled log-likelihood is given by

$$\mathcal{L}_n(\boldsymbol{\theta}, f|Y_1,\ldots,Y_n) = n^{-1}\left[\sum_{t=1}^n \log f(u_t(\boldsymbol{\theta})) - \log\sigma_t(\mathcal{F}_{t-1};\boldsymbol{\theta}))\right]\,. \tag{2.11}$$

The scaling factor n^{-1} has been introduced so that (2.11) does not increase without limit as the number of observations increases. When n becomes very large, we obtain

$$\lim_{n\to\infty} \mathcal{L}_n(\boldsymbol{\theta}, f|Y_1,\ldots,Y_n) = -\mathrm{KL}(f||f_{\boldsymbol{\theta}}) + S(f_{\boldsymbol{\theta}}) + \mathrm{constant}\,, \tag{2.12}$$

where $\mathrm{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler divergence between two density functions, $f_{\boldsymbol{\theta}}$ is the marginal density of the scaled residuals and $S(\cdot)$ represents the differential entropy of a density function. Conditioning to $\boldsymbol{\theta}$, (2.12) is maximized with respect to f when f is equal to $f_{\boldsymbol{\theta}}$. This motivates the following iterative algorithm for maximizing (2.11) with respect to the innovation density f and the model parameters $\boldsymbol{\theta}$. First, assuming that an approximation $\hat{f}$ for the innovation density is available, an estimate $\hat{\boldsymbol{\theta}}$ of the model parameters is obtained by maximizing $\mathcal{L}_n(\boldsymbol{\theta}, \hat{f}|X_1,\ldots,X_n)$ with respect to $\boldsymbol{\theta}$ holding $\hat{f}$ fixed. This step can be performed using standard optimization methods. Second, the current approximation $\hat{f}$ for the innovation density is updated by estimating the marginal density of the scaled residuals $u_1(\hat{\boldsymbol{\theta}}),\ldots,u_n(\hat{\boldsymbol{\theta}})$ using the back-transformed kernel method. Before performing the estimation of the density, the scaled residuals are standardized so that they have zero mean and unit standard deviation. This guarantees that $\hat{f}$ satisfies the constraints imposed by the semi-parametric model on the location and the scale of f (zero mean and unit standard deviation). The update rule for $\hat{f}$ is

$$\hat{f}(e) = |g'_{\hat{\boldsymbol{\pi}}}(e)|\frac{1}{n}\sum_{t=1}^n K_h\left[g_{\hat{\boldsymbol{\pi}}}\left(\frac{u_t(\hat{\boldsymbol{\theta}}) - m(\hat{\boldsymbol{\theta}})}{s(\hat{\boldsymbol{\theta}})}\right) - g_{\hat{\boldsymbol{\pi}}}(e)\right]\,, \tag{2.13}$$

```
Input: a time series $Y_1,\ldots,Y_n$.
Output: a parameter vector $\hat{\boldsymbol{\theta}}$ and a density $\hat{f}$.

1. Initialize $\hat{f}$ to the standard Gaussian density.
2. $\mathcal{L}_{old} \leftarrow \infty$, $\mathcal{L}_{new} \leftarrow -\infty$.
3. while $\mathcal{L}_{new} - \mathcal{L}_{old} <$ tolerance.
    (a) Update $\hat{\boldsymbol{\theta}}$ as the maximizer of $\mathcal{L}_n(\boldsymbol{\theta},\hat{f}|Y_1,\ldots,Y_n)$.
    (b) Update $\hat{f}$ using (2.13).
    (c) $\mathcal{L}_{old} \leftarrow \mathcal{L}_{new}$, $\mathcal{L}_{new} \leftarrow \mathcal{L}_n(\hat{\boldsymbol{\theta}},\hat{f}|Y_1,\ldots,Y_n)$.
4. Return $\hat{\boldsymbol{\theta}}$ and $\hat{f}$.
```

Figure 2.4: Iterative algorithm followed by SPE for maximizing $\mathcal{L}_n(\boldsymbol{\theta},f|Y_1,\ldots,Y_n)$.

where

$$m(\hat{\boldsymbol{\theta}}) = \frac{1}{n}\sum_{t=1}^{n} u_t(\hat{\boldsymbol{\theta}})\,, \qquad\qquad s^2(\hat{\boldsymbol{\theta}}) = \frac{1}{n-1}\sum_{t=1}^{n}\left[u_t(\hat{\boldsymbol{\theta}}) - m(\hat{\boldsymbol{\theta}})\right]^2 . \tag{2.14}$$

In this expression, the transformation $g_{\hat{\pi}}$ is a function based on the stable distribution whose parameters $\hat{\pi}$ are found by maximizing the likelihood of a stable density on the standardized and scaled residuals, as described in Section 2.2. These steps are repeated until the variation in $\mathcal{L}_n(\hat{\boldsymbol{\theta}},\hat{f}|Y_1,\ldots,Y_n)$ between two consecutive iterations is below a specified threshold. In the first iteration of the algorithm, $\hat{f}$ is assumed to be a standard Gaussian density. Figure 2.4 presents the complete pseudocode of SPE. The algorithm typically converges to a fixed point after only a few iterations, usually less than five.

The proposed semi-parametric estimator (SPE) uses similar ideas as the method designed by Engle and González-Rivera (1991) for the estimation of semi-parametric GARCH processes. However, there are two differences between both methods. First, Engle and González-Rivera's method involves only one and a half iterations of the steps shown in Figure 2.4. In particular, they only perform the first iteration completely and the second one is stopped after step *a*. This means that the estimates generated can be sub-optimal. The second and more important difference is that, instead of using back-transformed kernels, Engle and González-Rivera (1991) approximate f using the discrete maximum penalized likelihood estimation (DMPLE) technique of Scott et al. (1980). This method is described in Appendix A.4 and has the same limitations as the standard kernel density estimator. In particular, it does not generate smooth density estimates for the tails of the distribution when the data present extreme events. This is illustrated by the left plot in Figure 2.5, which displays the logarithm of a DMPLE density estimate for the daily returns of IBM stocks. The resulting approximation is very similar to the one generated by a standard kernel method (left plot in Figure 2.1). This failure of DMPLE to accurately model the tails of the distribution also has a negative effect in the estimation of $\boldsymbol{\theta}$, as will be seen in the next section.

2.3.1 Experiments with Simulated Data

We perform a series of experiments with simulated data to evaluate the quality of the estimates of $\boldsymbol{\theta}$ generated by SPE. Two alternative methods are also included in the experiments. The first

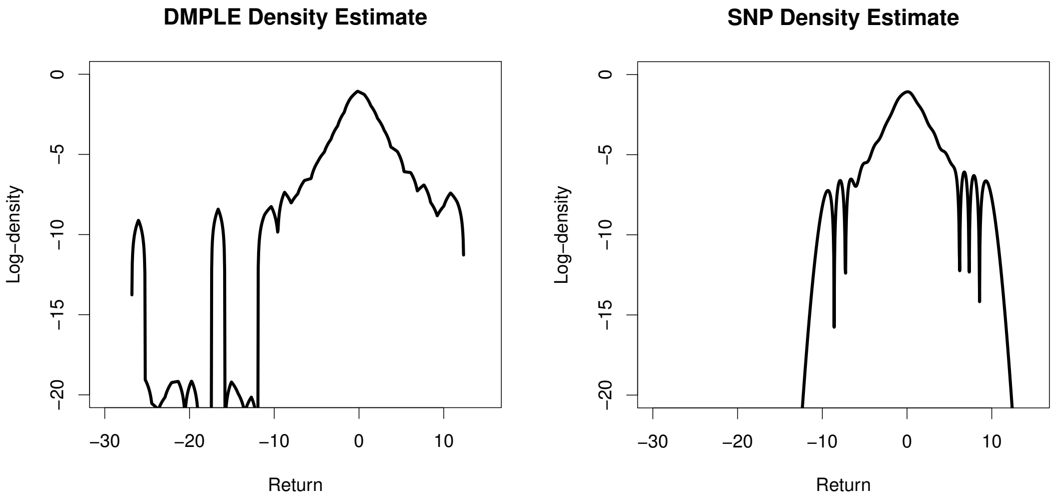

Figure 2.5: Left, logarithm of a DMPLE density estimate for the daily returns of IBM stocks. The configuration for the method (knots and penalty) is the same as the one used by Engle and González-Rivera (1991). Right, logarithm of the SNP density estimate for the same sample. A total of 25 basis functions are used for the expansion in terms of Hermite functions.

one is the maximum likelihood estimator (MLE) which makes use of the actual density of the innovations f. The second one (SPE-DMPLE) is also the method described in the previous section, with the exception that the back-transformed kernel estimator is replaced in this case by the DMPLE method when we have to estimate the density of the standardized and scaled residuals.

For each experiment, a time series with $n = 4000$ elements is simulated from the model described in Section 2.2.2. The innovations of the process follow a heavy-tailed density with known parametric form (either Student's t or NIG). The values of the parameters used for the simulation of the data are $\boldsymbol{\theta} = (\phi_0 = 0.02, \phi_1 = 0.05, \kappa = 0.07, \alpha = 0.05, \beta = 0.9, \gamma = 0.4)$. These specific parameter values are similar to the estimates obtained when the model (2.6) is calibrated on real financial data and f is assumed to be standard Gaussian. For each time series, the estimation of $\boldsymbol{\theta}$ is performed using MLE, SPE and SPE-DMPLE under the assumption that the parametric model of Section 2.2.2 holds. The semi-parametric methods do not make use of the actual density of the innovations f . For SEP-DMPLE, the configuration of the DMPLE method (knots and penalty) is the same as the one used by Engle and González-Rivera (1991). Each experiment is repeated 2500 times. The results reported are the average and the standard deviation (SD) of the estimates of the model parameters generated by each method.

Table 2.3 presents the results for innovations sampled from a Student's t distribution with 5 degrees of freedom, zero mean and unit standard deviation. Table 2.4 presents the results for NIG innovations with parameters $\alpha = 2$, $\beta = -0.1$, $\delta = \alpha^{-2}(\alpha^2 - \beta^2)^{\frac{3}{2}}$, $\mu = -\delta\beta(\alpha^2 - \beta^2)^{-\frac{1}{2}}$, where the values of δ and μ are selected so that the resulting NIG distribution has zero mean and unit standard deviation. The two right-most columns in these tables show the ratios between the empirical standard deviations of the MLE estimates and the empirical standard deviations of the two semi-parametric estimates (SPE and SPE-DMPLE). The MLE estimates are asymptotically optimal. Therefore, the standard deviation of the MLE estimates should be a lower bound of the standard deviation of any other estimate, provided that sufficient amounts of data are used.

Table 2.3: Mean and standard deviation (SD) for the estimates of $\boldsymbol{\theta}$ on simulated data with Student's *t* innovations. The two right-most columns contain the ratios between (1) the SD of the MLE estimates and (2) the SD of the semi-parametric estimates based on back-transformed kernels (SPE) and (1) and (3) the DMPLE approach (SPE-DMPLE).

	MLE		**SPE**		**SPE-DMPLE**		**Ratios**	
	Mean	**SD (1)**	**Mean**	**SD (2)**	**Mean**	**SD (3)**	**(1)/(2)**	**(1)/(3)**
ϕ_0	0.0205	0.0153	0.0204	0.0169	0.0209	0.0268	0.91	0.57
ϕ_1	0.0497	0.0150	0.0497	0.0151	0.0503	0.0240	0.99	0.62
κ	0.0763	0.0273	0.0746	0.0282	0.0818	0.0524	0.97	0.54
α	0.0499	0.0119	0.0487	0.0119	0.0522	0.0188	1.00	0.63
β	0.8943	0.0301	0.8941	0.0317	0.8887	0.0523	0.95	0.58
γ	0.4370	0.1821	0.4370	0.1838	0.4540	0.2628	0.99	0.69

Table 2.4: Mean and standard deviation (SD) for the estimates of $\boldsymbol{\theta}$ on simulated data with NIG innovations. The two right-most columns display the ratios between (1) the SD of the MLE estimates and (2) the SD of the semi-parametric estimates based on back-transformed kernels (SPE) and (1) and (3) the DMPLE approach (SPE-DMPLE).

	MLE		**SPE**		**SPE-DMPLE**		**Ratios**	
	Mean	**SD (1)**	**Mean**	**SD (2)**	**Mean**	**SD (3)**	**(1)/(2)**	**(1)/(3)**
ϕ_0	0.0202	0.0172	0.0204	0.0176	0.0202	0.0197	0.97	0.87
ϕ_1	0.0501	0.0159	0.0500	0.0161	0.0497	0.0218	0.99	0.73
κ	0.0751	0.0241	0.0736	0.0239	0.0761	0.0292	1.01	0.82
α	0.0498	0.0110	0.0487	0.0109	0.0522	0.0131	1.01	0.84
β	0.8956	0.0267	0.8955	0.0270	0.8953	0.0311	0.99	0.86
γ	0.4298	0.1517	0.4301	0.1553	0.4416	0.2039	0.98	0.74

For both types of innovations (Student's *t* and NIG), the standard deviations of the estimates generated by SPE are close to the standard deviations of the MLE estimates. However, the estimates generated by SPE-DMPLE are rather inefficient. They have larger standard deviations than those generated by SPE. The loss of efficiency is stronger when the innovations follow a Student's *t* distribution, which has heavier tails than the NIG distribution.

The reason for the poor efficiency of the SPE-DMPLE estimates is that, similarly to standard kernel methods, the approximation of the density generated by the DMPLE method is very inaccurate in the tails of the distribution when the data are heavy-tailed. In particular, the small bumps that appear in the tails of the approximation (see the left plot in Figure 2.5) reduce the quality of the updates of $\hat{\boldsymbol{\theta}}$ generated during step *a* of the iterative algorithm. These bumps do not allow $\hat{\boldsymbol{\theta}}$ to change significantly between two consecutive iterations and the algorithm finally stops at sub-optimal solutions. This failure of DMPLE to accurately describe the tails of heavy-tailed densities was already noted by Engle and González-Rivera (1991).

2.3.2 Experiments with Financial Data

We now investigate the performance of SPE in the modeling of time series of financial returns. The model described in Section 2.2.2 is assumed to have generated the return observations. The data analyzed are the daily returns of IBM, GM, and the S&P 500 index during the period from January 3rd, 1962 to May 6th, 2008. Each of these series contains 11,665 elements and they are computed using the closing prices adjusted for dividends and splits. The adjusted price values are publicly available at http://www.finance.yahoo.com. The validation of SPE is implemented in a sliding-window experiment. Each series of 11,665 returns is split into 9665 overlapping windows of size 2000. Each window is equal in length to the previous one but its elements are displaced forward one unit in time (a day). The semi-parametric model is estimated on the data within each window and then tested on the first return out of the window. The test process involves the transformation of the return using the conditional probability distribution given by the model and then applying the standard Gaussian quantile function. Under the null hypothesis that the estimated model generated the data, the resulting 9665 *test measurements* are distributed according to a standard Gaussian distribution (Berkowitz, 2001). However, instead of applying standard Gaussianity tests, which generally focus on deviations in the body of the distribution, we use the tests proposed by Kerkhof and Melenberg (2004) for expected shortfall (ES), Value at Risk (VaR) and exceedances (Exc) at the risk level 99%. These statistical tests are particularly sensitive to deviations in the tail corresponding to losses, which is the relevant part of the distribution of financial returns in risk analysis (Dowd, 2005; Jorion, 1997). In particular, they evaluate the capacity of a model to generate accurate risk estimates using standard measures of risk such as *expected shortfall* and *Value at Risk*. A detailed description of these tests and of expected shortfall and Value at Risk is given in Appendix A.3.

The performance of SPE is also compared with three benchmark techniques. Two of them correspond to fully parametric models in which a particular functional form is assumed for the distribution of the innovations. Besides $\boldsymbol{\theta}$, these parametric models include two extra parameters that quantify the heavy-tailedness and the asymmetry of the innovations, respectively. In these models, all the parameters are estimated by maximum likelihood. In the first parametric model, the innovations are assumed to follow a NIG distribution with zero mean and unit standard deviation. This is a plausible model for the conditional distribution of financial returns which is motivated by the *Mixture of Distributions Hypothesis* (MDH) (Clark, 1973) and the idea of *realized volatilities* (RV) (Andersen et al., 2003) as noted by Forsberg and Bollerslev (2002). The MDH postulates that the conditional distribution of financial returns is Gaussian, but with a stochastic (latent) variance. RV approximate this latent variance by the summation of finely sampled squared high-frequency returns. An accurate model for the unconditional distribution of RV and therefore, for the unconditional distribution of the latent return variances, is an inverse Gaussian distribution (Forsberg and Bollerslev, 2002). This, together with the MDH, results in a NIG description for the unconditional distribution of financial returns. The second parametric model which is analyzed assumes stable innovations. In particular, f is considered be a stable density with location and scale parameters 0 and $1/\sqrt{2}$, respectively, where we have used the first parameterization of stable distributions given by Nolan (2002). This model is considered because SPE makes use of a stable distribution to transform the data before performing the density estimation by kernels. The last benchmark method corresponds to the semi-parametric model which is obtained when the unknown density of the innovations is described using the SNP method (Fenton and Gallant, 1996; Gallant and Nychka, 1987). In this case, the density of the innovations is approximated by an expansion in Hermite functions that always satisfies the zero mean and unit standard deviation constraints. A total of 9 functions are used in the expansion because this choice gives the best overall results. The estimation of the SNP model

Table 2.5: *p*-values of the statistical tests at the 99% risk level for each estimator.

Test	Asset	SPE	MLE-NIG	MLE-stable	SNP
	IBM	0.5103	0.000001	0.2057	0.000385
ES	GM	0.3326	0.000044	0.1418	0.000983
	S&P	0.1226	0.000485	0.1855	0.000004
	IBM	0.0985	0.0903	0.00097	0.0991
VaR	GM	0.0934	0.0307	0.00021	0.1152
	S&P	0.1156	0.6523	0.01751	0.3295
	IBM	0.1723	0.2067	0.00706	0.1723
Exc	GM	0.0606	0.0290	0.00008	0.0375
	S&P	0.2459	0.5162	0.01698	0.2901

is performed by maximizing the likelihood of the model parameters. A description of the SNP method is given in Appendix A.5.

Table 2.5 contains the resulting *p*-values of the different goodness-of-fit tests for each method and each financial asset. If we take $\alpha = 0.05$, the model adjusted by SPE cannot be rejected by any test. The parametric model with NIG innovations (MLE-NIG) fails the tests for expected shortfall (ES) with very low *p*-values. Similar results are obtained by the model with stable innovations (MLE-stable), which fails the tests for Value at Risk (VaR) and exceedances (Exc) with very low *p*-values. Finally, the SNP model is strongly rejected by the expected shortfall tests. Note that these tests should be corrected for multiple comparisons to reduce the number of false positives. Nevertheless, the *p*-values obtained by the parametric models or the SNP approach are in many cases so small that they would be rejected even when a correction for multiple testing is applied.

Figure 2.6 displays the Gaussian QQ plots of the test measurements generated by each method on the daily returns of GM. The corresponding plots for the other financial assets are similar. The parametric model MLE-NIG and the SNP method clearly underestimate the loss tail of the conditional return distribution. On the other hand, the parametric model MLE-stable seems to have difficulties describing the conditional return density around the probability level corresponding to the Gaussian quantile -2.5. In this region, the empirical quantiles differ from the Gaussian ones and a small deviation of the points from the straight line can be observed in the plot. Finally, SPE generates an accurate fit of the conditional return distribution, especially in the tails. Similar patterns are observed in the Gaussian QQ plots corresponding to the other two financial assets analyzed, that is, IBM and the S&P 500 index.

The poor results of the semi-parametric method based on the SNP approach have their origin in the limitations of the Hermite functions to accurately model the heavy-tails of the conditional distribution of returns. In particular, although a sufficiently large expansion in terms of Hermite polynomials can approximate any density function up to any degree of precision, in practice, the exponential decay of these functions generates a density estimate with tails that quickly approach zero. This is illustrated by the plot in right part of Figure 2.5, which displays the SNP density estimate for the daily returns of IBM stocks. The tails of the approximation are in this case very light. We have used a total of 25 Hermite functions for the expansion, however, the resulting plot does not depend strongly on the exact number of functions used. By contrast, the

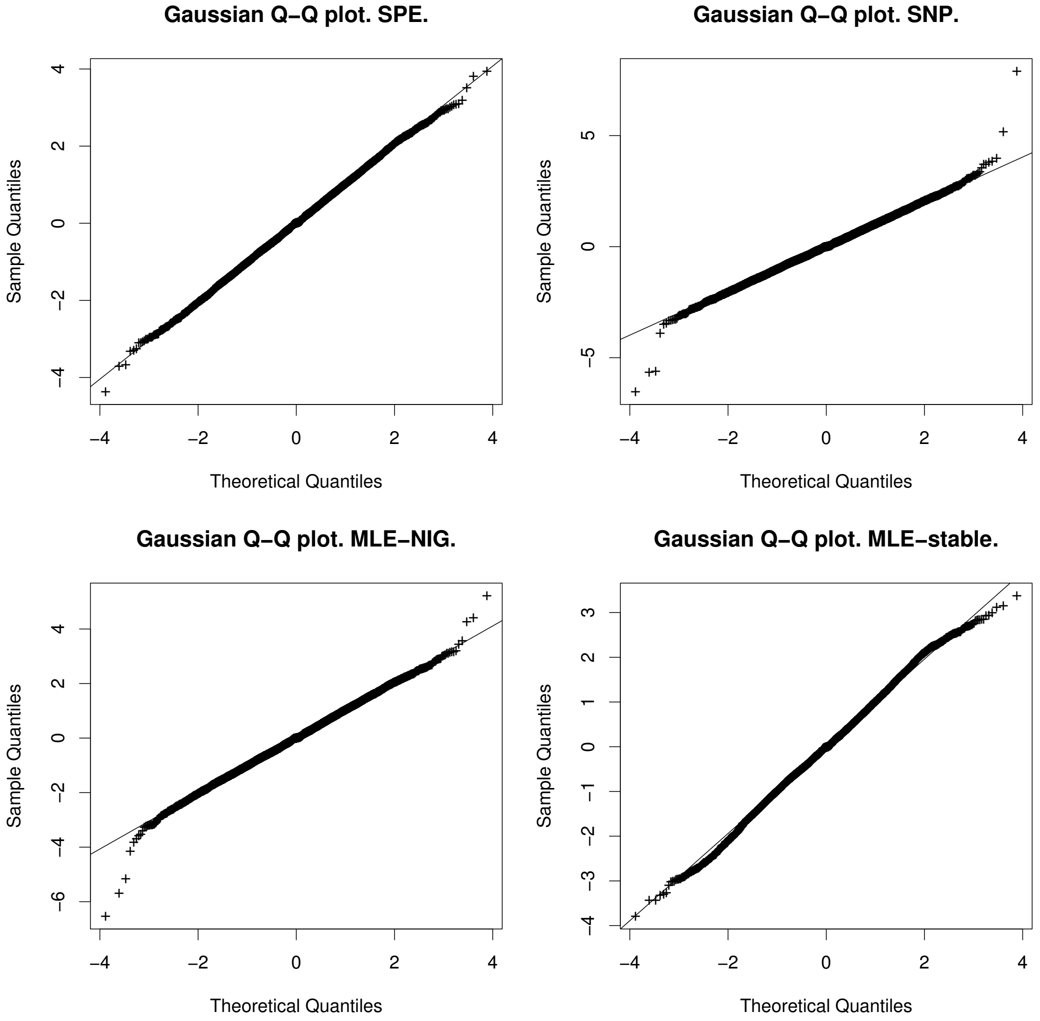

Figure 2.6: Gaussian QQ plots of the test measurements obtained by each model on the returns of GM. Top left, the novel semi-parametric estimator. Top right, the SNP semi-parametric method. Bottom left, the parametric model with f NIG. Bottom right, the parametric model with f stable.

tails of the approximation generated by a back-transformed kernel method are much heavier, as indicated by the right plot in Figure 2.1.

2.4 Summary and Discussion

Financial time-series models generally assume a parametric form for both the trends and the innovations. However, parametric models for the innovations often lack expressive capacity or flexibility to describe the features of the empirical data and, in particular, the distribution of extreme events. This observation motivates the use of a semi-parametric approach, in which the distribution of the innovations is directly learned from the training data. However, because the actual innovations are heavy-tailed, some specific non-parametric method is needed to guarantee that the tails of the return distribution are correctly modeled. Our approach for modeling the density of the innovations is based on the kernel density estimator framework (Silverman, 1986). To improve the quality of the density approximation in the tails, we follow Wand et al. (1991) and perform the estimation of the density in a transformed space, where the transformed data are approximately Gaussian. The transformation function is based on a fit of a stable distribution (Nolan, 2002) to the original data. Experiments on simulated data show the superiority of the back-transformed kernel estimator over standard and adaptive kernel methods (Silverman, 1986) when the distribution of the data is heavy-tailed. Additionally, an iterative algorithm (SPE) is introduced for the estimation of semi-parametric financial time-series models in which the unknown innovation distribution is approximated using a back-transformed kernel method. SPE generates estimates of the model parameters which are very close to the ones obtained by the maximum likelihood method when the actual functional form of the innovations is known. A series of experiments with empirical data show that SPE provides very accurate estimates of the conditional return density, especially in the tails of the distribution.

In this chapter, we have considered unidimensional time-series models for the description of price changes of single financial assets. However, we may be interested in the construction of multivariate models which are able to describe the joint evolution of the prices of several financial stocks. In particular, we would like to extend the proposed semi-parametric method to higher-dimensions. This can be accomplished by using copula functions (Joe, 1997). Copulas allow to link separate univariate models into a joint multivariate model. For this process to be successful, we need accurate copula methods that are able to learn the dependencies present in the data without suffering from significant overfitting problems. The following chapter presents a novel semi-parametric bivariate copula method that can be used for this task.

Chapter 3

Semi-parametric Bivariate Archimedean Copulas

The theory of copulas provides a general framework for the construction of multivariate models with a specific dependence structure and specific univariate marginal distributions. Parametric copulas often lack expressive capacity to capture the complex dependencies that often appear in empirical data. By contrast, non-parametric copulas can have poor generalization performance because of overfitting. As an intermediate approach, we introduce a flexible semi-parametric bivariate Archimedean copula model that provides accurate and robust fits. The Archimedean copula is expressed in terms of a latent function that can be readily represented using a basis of natural cubic splines. The model parameters are determined by maximizing the sum of the log-likelihood function and a term that penalizes non-smooth solutions. The performance of the semi-parametric estimator is analyzed in experiments with simulated and real-world data, and compared to other methods for copula estimation: three parametric copula models, two semi-parametric estimators of Archimedean copulas previously introduced in the literature, two flexible methods that can describe more general and non-Archimedean dependence structures and finally, standard parametric Archimedean copulas. The good overall performance of the proposed semi-parametric approach confirms the capacity of this method to capture complex dependencies in the data while avoiding overfitting.

3.1 Introduction

Many standard univariate models do not have a simple extension to two or higher dimensions. In practice, only a reduced number of parametric distributional families with a closed analytical form are available for modeling multivariate data. Some examples can be found in the family of elliptical distributions (Fang et al., 1990). This family includes the multivariate Gaussian, the multivariate Student's t and the elliptically contoured stable distributions (Nolan, 2002). However, the elliptical family has a limited range of distributional shapes and often cannot provide an accurate fit for empirical multivariate data. One of the main limitations of elliptical distributions is that they cannot model asymmetries in the data. Nevertheless, it is possible to design extensions of the elliptical family that incorporates skewness (Genton, 2004).

The theory of copulas (Joe, 1997) provides a framework for the construction of multivariate models by expressing the distribution of the data in a canonical form that models the marginals separately from the dependence structure of the data. Let $(X_1,\ldots,X_d)^{\mathrm{T}}$ be a continuous random vector that follows distribution F and let F have continuous marginal distributions $F_1,\ldots,F_d$, where F_i is the marginal of X_i. Here, F and $F_1,\ldots,F_d$ are cumulative distribution functions. A theorem due to Sklar (1959) states that there is a unique function C, denoted the copula of F, such that

$$F(x_1,\ldots,x_d) = C\left[F_1(x_1),\ldots,F_d(x_d)\right] . \tag{3.1}$$

Therefore, the joint distribution F is uniquely determined by its marginals $F_1,\ldots,F_d$ and its copula C, where C is a cumulative distribution function with uniform marginals defined in the d-dimensional unit hypercube. Multivariate probabilistic models can be built by first fitting a different univariate model for each marginal and then, learning a copula function that links the univariate specifications in a joint multivariate model. The first step is straightforward and can be implemented using standard methods for modeling univariate data. The second step requires the specification of copula models that are both flexible, so that they are able to capture complex dependence structures in the data, and robust, so that overfitting is avoided. Parametric copula models such as the Gaussian or the Student's t copulas are robust, but they generally lack expressive capacity to represent the complex multivariate dependencies that can be found in real-world data. Non-parametric copula models can approximate arbitrarily complex dependencies when sufficient amounts of data are available (Fermanian and Scaillet, 2003). However, this high flexibility and the absence of any distributional assumption on the underlying copula makes non-parametric methods more prone to overfitting. In this chapter, we adopt a semi-parametric approach based on the family of bivariate Archimedean copulas (Nelsen, 2006) that aims to strike a balance between flexibility and robustness (Hernández-Lobato and Suárez, 2009).

Bivariate Archimedean copulas are a specific class of copulas that are uniquely determined by a unidimensional generator function. Parametric Archimedean copulas assume a particular functional form for this generator, which depends only on a few parameters (Nelsen, 2006). More flexible models can be obtained when the Archimedean generator is expressed in terms of a one-dimensional functional parameter as proposed by Vandenhende and Lambert (2005), Lambert (2007), Gagliardini and Gourieroux (2007) and Dimitrova et al. (2008). Following the latter approach, we express the Archimedean generator in terms of a latent function that is simpler to model than the generator itself. The latent function is then represented using a basis of natural cubic splines. The coefficients of the expansion in the spline basis are computed by maximizing an objective function that includes the log-likelihood of the model and a term that penalizes the curvature of the functional parameter. This form of regularization is particularly convenient because latent functions with low curvature generate smooth copulas, which are less prone to overfitting.

Experiments with simulated data, and data from different domains of application (financial asset returns (Yahoo! Finance, 2008) and precipitation data (Razuvaev et al., 2008)) are carried out to assess the performance of the proposed semi-parametric estimator. In these experiments, the novel approach is compared with alternative methods for bivariate copula estimation. These include estimators of Archimedean copulas based on (a) Bayesian B-splines (Lambert, 2007) and (b) GeD splines (Dimitrova et al., 2008); (c) a flexible copula model based on Bayesian mixtures of Gaussians (Attias, 1999; Bishop, 2006); (d) a non-parametric copula model based on Gaussian kernel density estimators (Duong and Hazelton, 2003; Fermanian and Scaillet, 2003); (e) parametric Gaussian and Student's t copulas (Malevergne and Sornette, 2006); (f) parametric skewed Student's t copulas (Demarta and McNeil, 2005) and finally, (g) standard parametric Archimedean copulas (Nelsen, 2006). The excellent overall performance of the proposed semi-parametric copula method in the problems investigated confirms the capacity of this approach to capture complex dependencies in the data while avoiding overfitting.

The rest of the chapter is organized as follows. Section 3.2 introduces a parameterization of bivariate Archimedean copulas in terms of a novel latent function. Section 3.3 describes the method proposed for the estimation of this function given a bivariate copula sample. Section 3.4 presents a complete experimental evaluation of the new semi-parametric copula and subsections 3.4.1, 3.4.2 and 3.4.3 describe the results obtained for simulated, financial and precipitation data, respectively. Finally, Section 3.5 contains a summary and a discussion.

3.2 Parameterizations of Bivariate Archimedean Copulas

In this section, we introduce a novel specification of bivariate Archimedean copulas in terms of a latent function that can be readily approximated using a finite basis of natural cubic splines. A bivariate Archimedean copula is uniquely determined by its generator ϕ^{-1}, where we have used the notation for the generator given by Gagliardini and Gourieroux (2007). The generator is a function defined in the unit interval which is strictly decreasing, convex and satisfies $\phi^{-1}(0) = \infty$ and $\phi^{-1}(1) = 0$. The bivariate Archimedean copula is defined in terms of ϕ^{-1} as

$$C(u,v) = \phi\left[\phi^{-1}(u) + \phi^{-1}(v)\right], \quad u,v \in [0,1], \tag{3.2}$$

where ϕ is the inverse function of ϕ^{-1}. The corresponding copula density is

$$c(u,v) = \frac{\phi''\left[\phi^{-1}(u) + \phi^{-1}(v)\right]}{\phi'\left[\phi^{-1}(u)\right]\phi'\left[\phi^{-1}(v)\right]}, \quad u,v \in [0,1]. \tag{3.3}$$

Modeling ϕ^{-1} directly presents some difficulties because this function must satisfy rather stringent constraints. For this reason, different authors have proposed to specify the Archimedean copula in terms of a latent function that is easier to model. Lambert (2007) suggests

$$\lambda(w) = \phi^{-1}(w)\left\{\frac{d}{dw}\phi^{-1}(w)\right\}^{-1}, \tag{3.4}$$

where λ satisfies $\lambda(w) < 0$ and $\lambda'(w) < 1$ for $w \in [0,1]$ and $\lambda(0) = \lambda(1) = 0$. Dimitrova et al. (2008) express the copula in terms of

$$K(w) = w - \lambda(w), \tag{3.5}$$

where K satisfies $K(w) > w$ and $K'(w) > 0$ for $w \in [0,1]$, $K(0) = 0$ and $K(1) = 1$. Finally, Gagliardini and Gourieroux (2007) introduce a positive function f defined in the unit interval

$$f(w) = -\frac{\phi''[\phi^{-1}(w)]}{\phi'[\phi^{-1}(w)]} = -\frac{d}{dw}\phi'[\phi^{-1}(w)], \quad w \in [0,1]. \tag{3.6}$$

The latent function f is in one-to-one correspondence with the generator, provided that $f(w) > 0$ for $w \in [0,1]$ and

$$\int_0^1 \frac{1}{F(w)} dw = \infty, \quad \text{where} \quad F(w) = \int_0^w f(w')\, dw' = -\phi'[\phi^{-1}(w)]. \tag{3.7}$$

The generator ϕ^{-1} is expressed in terms of F as

$$\phi^{-1}(w) = \int_w^1 \frac{dw'}{F(w')}. \tag{3.8}$$

Despite its simplicity, this latter specification still presents some modeling difficulties when the Archimedean copula is required to have upper or lower tail dependence (Joe, 1997). Informally speaking, a copula is said to have upper (lower) tail dependence when the limiting probability λ_U (λ_L) that one variable exceeds (is under) a large (small) threshold, given that the other variable is also above (below) this threshold is positive: $\lambda_U > 0$ ($\lambda_L > 0$). Appendix B.1 provides a more formal definition of upper and lower tail dependence.

For the Archimedean copula to have upper (lower) tail dependence, $f(w)$ must exhibit a convergence behavior as $w \uparrow 1$ ($w \downarrow 0$) that is difficult to model using standard approximation techniques. To overcome this shortcoming, we introduce a new latent function g that is in a one-to-one correspondence with f and in which tail dependence is enforced using simpler asymptotic conditions. Before describing the novel parameterization of bivariate Archimedean copulas, it is useful to review some properties of the latent function f given by Gagliardini and Gourieroux (2007). Using the definition (3.2) of the Archimedean copula in terms of ϕ^{-1}

$$\phi^{-1}(C(u,v)) = \phi^{-1}(u) + \phi^{-1}(v). \tag{3.9}$$

Combining this with (3.6) and (3.7) we obtain

$$f(C(u,v)) = -\frac{\phi''[\phi^{-1}(u)+\phi^{-1}(v)]}{\phi'[\phi^{-1}(u)+\phi^{-1}(v)]} \quad \text{and} \quad F(C(u,v)) = -\phi'[\phi^{-1}(u)+\phi^{-1}(v)]. \tag{3.10}$$

Therefore, the copula density in terms of f is

$$c(u,v) = f[C(u,v)]\frac{F[C(u,v)]}{F(u)F(v)}, \tag{3.11}$$

where F and C also depend on f. Given the empirical sample $\mathcal{D} = \{U_i, V_i\}_{i=1}^N$ with $U[0,1]$ marginals, the likelihood of an Archimedean copula parameterized in terms of f is

$$\mathcal{L}(\mathcal{D}|f) = \prod_{i=1}^N c(U_i,V_i) = \prod_{i=1}^N f[C(U_i,V_i)]\frac{F[C(U_i,V_i)]}{F(U_i)F(V_i)}. \tag{3.12}$$

According to corollary B.1.1 in Appendix B.1, when the coefficients of tail dependence are $0 \leq \lambda_L \leq 1$ and $0 \leq \lambda_U \leq 1$ for the lower and the upper tail, respectively, the latent function f in the Archimedean copula satisfies

$$f(w) = w^{\gamma_L} \ell_L(w) \quad \text{when} \quad w \downarrow 0\,, \tag{3.13}$$

$$f(w) = (1-w)^{\gamma_U} \ell_U(w) \quad \text{when} \quad w \uparrow 1\,, \tag{3.14}$$

where $\gamma_L = -\log 2/\log \lambda_L$, $\gamma_U = -\log 2/\log(2-\lambda_U)$ and $\ell_L \in \mathcal{R}^{0,0}$ and $\ell_U \in \mathcal{R}^{0,1}$ are slowly varying functions as $w \downarrow 0$ and as $w \uparrow 1$, respectively. This means that f goes to zero as $w \downarrow 0$ when $\lambda_L > 0$ and that f diverges as $w \uparrow 1$ when $\lambda_U > 0$. Therefore, finite basis of bounded functions, such as B-splines, Gaussian or logistic functions are not useful for modeling f when the data present upper or lower tail dependence.

To address these difficulties, we propose (i) to transform the range of values on which f is defined from the unit interval to the real line using the logistic function and (ii) to enforce the positivity requirement by modeling the logarithm of f instead of f itself. Thus, we introduce the alternative latent function

$$g(x) = \log f[\sigma(x)]\,, \quad x \in \mathbb{R}\,, \tag{3.15}$$

which is in one-to-one correspondence with f

$$f(w) = \exp\left\{g[\sigma^{-1}(w)]\right\}\,, \quad w \in [0,1]\,, \tag{3.16}$$

where σ and σ^{-1} are respectively the logistic function and its inverse, namely

$$\sigma(x) = \frac{1}{1+\exp(-x)}\,, \qquad \sigma^{-1}(w) = \log\left(\frac{1}{1-w}\right)\,. \tag{3.17}$$

Given a two-dimensional dataset $\mathcal{D} = \{U_i, V_i\}_{i=1}^N$ with uniform $U[0,1]$ marginals, the log-likelihood of an Archimedean copula parameterized in terms of g is

$$\log \mathcal{L}(\mathcal{D}|g) = \sum_{i=1}^{N} g\left\{\sigma^{-1}[C(U_i,V_i)]\right\} + \log \frac{F[C(U_i,V_i)]}{F(U_i)F(V_i)}\,, \tag{3.18}$$

where C and F also depend on g.

According to Theorem B.2.1 in Appendix B.2, we can construct an Archimedean copula with upper and lower tail dependence coefficients λ_U and λ_L using a real function g that satisfies

$$g(x) = \gamma_L x + \ell_L(x) \quad \text{when} \quad x \to -\infty\,, \tag{3.19}$$

$$g(x) = -\gamma_U x + \ell_U(x) \quad \text{when} \quad x \to \infty\,, \tag{3.20}$$

where ℓ_L and ℓ_U are additively slowly varying functions as $x \to -\infty$ and as $x \to \infty$, respectively, $\gamma_L = -\log 2/\log \lambda_L$ and $\gamma_U = -\log 2/\log(2-\lambda_U)$. It can be difficult to give and accurate model for ℓ_L and ℓ_U if g is represented in terms of a finite basis of non-linear functions. However, the variation of these functions is very slow in the asymptotic region and they can be well approximated by a constant. Therefore, we restrict our attention to Archimedean copulas with

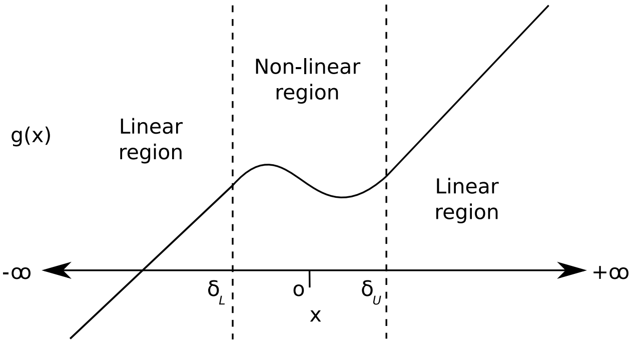

Figure 3.1: Assuming that ℓ_L and ℓ_U are approximately constant, the latent function g behaves linearly for the regions $x < \delta_L < 0$ and $x > \delta_U > 0$.

constant ℓ_L and ℓ_U when $x < \delta_L < 0$ and $x > \delta_U > 0$, respectively. Under this assumption, g can have an arbitrary non-linear shape in the central region $\delta_L < x < \delta_U$ but is restricted to be linear in the regions $x < \delta_L$ and $x > \delta_U$, as sketched in Figure 3.1. These restrictions are readily fulfilled by approximations of g in terms of natural cubic splines with boundary knots at δ_L and δ_U. These splines are linear in the regions beyond the boundary knots. The central non-linear region determines the dependence structure in the body of the copula, while the slopes of g in the linear regions specify the level of dependence in the tails of the bivariate distribution. Not all the possible values for the slopes of g are valid. According to Theorem B.2.2, the slope of g as $x \to -\infty$ must be non-negative to guarantee that ϕ^{-1} is a valid generator.

Modeling g using natural splines has several advantages over using, for instance, bivariate splines to describe the copula function C in a fully non-parametric manner. In particular, a copula C must satisfy very stringent constraints, such as being 2-increasing (Nelsen, 2006) and fulfilling $C(u,0) = C(v,0) = 0$, $C(u,1) = u$ and $C(1,v) = v$ for any u and v in the unit interval. Accounting for these restrictions using bivariate splines whose coefficients are adjusted by a fit to the data can be very difficult. A possible solution is to model the empirical copula using linear spline functions (Shen et al., 2008). However, this method has the disadvantage of the resulting copula function being piecewise linear, which means that the corresponding copula density is piecewise constant and discontinuous. Similar difficulties arise when bivariate splines are used to approximate c, the density of the copula, which must be positive, integrate to one and have uniform marginals. By contrast, the constraints on g are simple to enforce (asymptotic linearity, with a left asymptote of non-negative slope) and the estimation of this function from data requires only to compute a few unidimensional integrals.

Figure 3.2 displays graphs of g for different parametric Archimedean copulas and different levels of dependence in terms of Kendall's tau (Joe, 1997). The copulas considered belong to the Clayton, Gumbel and Frank Archimedean families (Nelsen, 2006). In all cases, g becomes approximately linear for sufficiently large values of $|x|$. The independent copula ($\tau = 0$) is a particular case of these copulas and corresponds to a horizontal straight line. Frank copulas have no tail dependence. This is reflected in the fact that the slope of g vanishes as $x \to \pm\infty$.

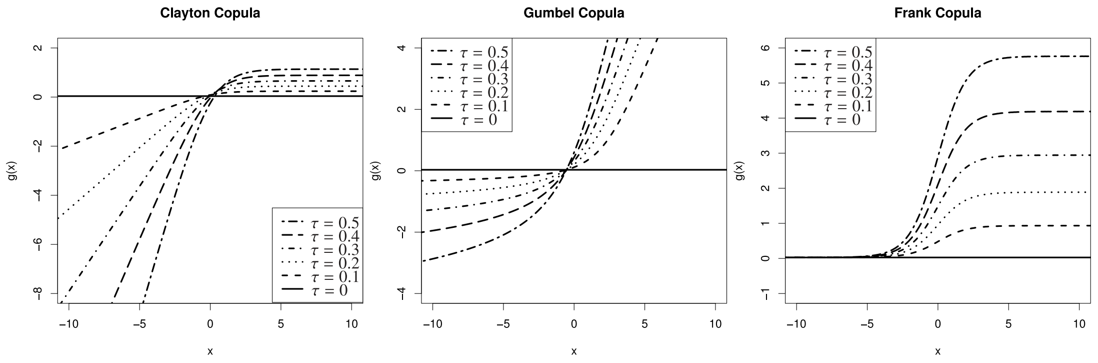

Figure 3.2: Plots of the function g corresponding to different families of Archimedean copulas and different levels of dependence measured in terms of Kendall's tau. The different functions are well described by a central non-linear region and two asymptotically linear regions in the tails.

3.3 Estimation of Bivariate Archimedean Copulas

Consider the problem of approximating the bivariate Archimedean copula that has generated a sample $\mathcal{D} = \{U_i, V_i\}_{i=1}^N$ with uniform $U[0,1]$ marginals. This is an ill-posed inverse problem (O'Sullivan, 1986). In consequence, some constraints must be imposed on the set of candidate models. We assume that the copula is parameterized in terms of a latent function g, which is linear for large values of $|x|$ and has a positive slope as $x \to -\infty$ so that ϕ^{-1} is a valid generator (see Theorem B.2.2). Given that g is assumed to be linear beyond thresholds $x < \delta_L < 0$ and $x > \delta_U > 0$, a basis of natural cubic splines (de Boor, 1978) with boundary knots at δ_L and at δ_U is an appropriate approximation for g. This basis is fully determined by specifying the number and the locations for the knots of the splines. The optimal choices of these parameters strongly depend on the actual form of the copula from which the multidimensional data have been extracted, which is unknown. Nevertheless, reasonable choices can be made using the evidence given by the available data.

The joint density of $w = C(u, v)$ and $z = v$ is given by

$$p(w,z) = \frac{f(w)}{F(z)} \mathbf{I}\{w \leq z\}\,, \tag{3.21}$$

where the term $\mathbf{I}\{w \leq z\}$ appears because any copula function C satisfies $C(u,v) \leq \min(u,v)$ for any u and v (Nelsen, 2006). Because z is uniformly distributed in the unit interval, (3.21) also represents $p(w|z)$, the conditional density of w given z. As noted by Gagliardini and Gourieroux (2007), if the non-negative f is interpreted as an unnormalized density function, (3.21) can be understood as a left-truncated model with threshold z, in which the density of variable w before truncation is proportional to $f(w)$. This interpretation suggests that the knots should be placed at uniformly spaced quantiles of the empirical distribution of $\sigma^{-1}(w)$. However, to compute $\{W_i = C(U_i, V_i)\}_{i=1}^N$ it is necessary to know the true copula C that generated the data, which is precisely the function that is being estimated. A possible solution is to approximate W_i by

$\hat{W}_i = \hat{C}(U_i,V_i)$, where $\hat{C}$ is the empirical copula of the sample $\mathcal{D} = \{U_i,V_i\}_{i=1}^N$:

$$\hat{C}(u,v) = \frac{1}{N+1}\sum_{i=1}^{N}\mathbf{I}\{U_i \leq u, V_i \leq v\} \tag{3.22}$$

and the usual factor N^{-1} is replaced by $(N+1)^{-1}$ to ensure that $\hat{C}(U_i,V_i) < 1$ for $1 \leq i \leq N$. Therefore, the knots are placed at uniform quantiles of the distribution

$$\hat{P}(x) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{I}\left\{\sigma^{-1}(\hat{W}_i) \leq x\right\} . \tag{3.23}$$

The quantile function is

$$\hat{Q}(x) = \inf\left\{y \in \mathbb{R} : \hat{P}(y) \geq x\right\} . \tag{3.24}$$

Note that (3.21) can only be interpreted as a truncation model if $f \in \mathcal{R}^{\beta,1}$ where $\beta > -1$ so that by Karamata's Theorem (Bingham et al., 1987) $F(1) < \infty$. Otherwise, f cannot be normalized. In spite of this, the rule used to determine the location of the knots works well even when $f \in \mathcal{R}^{\beta,1}$ and $\beta \leq -1$.

The optimal choice for the number of knots involves a trade-off between the robustness and the flexibility of the resulting model. If the number of knots is too large, the model is too sensitive to random fluctuations in the training data, leading to poor generalization performance (high flexibility, low robustness). By contrast, if the number of knots is too small, the model may be too rigid to capture complex dependence structures and the quality of the fit can also be poor (low flexibility, high robustness). Similarly as (Lambert, 2007), a large number of knots is chosen so that the model is sufficiently flexible. A regularization approach is adopted to make the approximation robust (Tikhonov and Arsenin, 1977). Following the recommendation of Lambert (2007) we employ 20 knots. Over-fitting is avoided by introducing a penalty on the second derivative of the function to be estimated (Eilers and Marx, 1996; Lambert, 2007; O'Sullivan, 1986; Reinsch, 1967). The estimate of g is obtained by maximizing the penalized log-likelihood

$$\text{PLL}(\mathcal{D}|g,\beta) = \log\mathcal{L}(\mathcal{D}|g) - \beta\int\left\{g''(x)\right\}^2 dx, \tag{3.25}$$

where $\log\mathcal{L}(\mathcal{D}|g)$ is given by (3.18) and $\beta \geq 0$ is a smoothing parameter, whose value is determined using a 10-fold cross validation grid search. The approximation of g in terms of natural cubic splines is

$$g(x) = \sum_{i=1}^{K}\theta_i N_i(x), \tag{3.26}$$

where $\theta_1,\ldots,\theta_K$ are real-valued coefficients and $N_1,\ldots,N_K$ form a K-dimensional set of basis functions that is uniquely determined by the sequence of knots $\xi_1 < \xi_2 < \ldots < \xi_M$, with $\xi_1 = \delta_L$ and $\xi_M = \delta_U$. Appendix B.3 describes how to efficiently compute the $N_1,\ldots,N_K$ non-linear basis functions. Figure 3.3 displays an example of such a functional basis. Once the knots of the splines are fixed, g depends only on the coefficients $\boldsymbol{\theta} = \{\theta_1,...,\theta_K\}$. Therefore, the maximum of (3.25) with respect to g is found by maximizing

$$\text{PLL}(\mathcal{D}|\boldsymbol{\theta},\beta) = \log\mathcal{L}(\mathcal{D}|\boldsymbol{\theta}) - \beta\boldsymbol{\theta}^{\text{T}}\boldsymbol{\Omega}\boldsymbol{\theta} \tag{3.27}$$

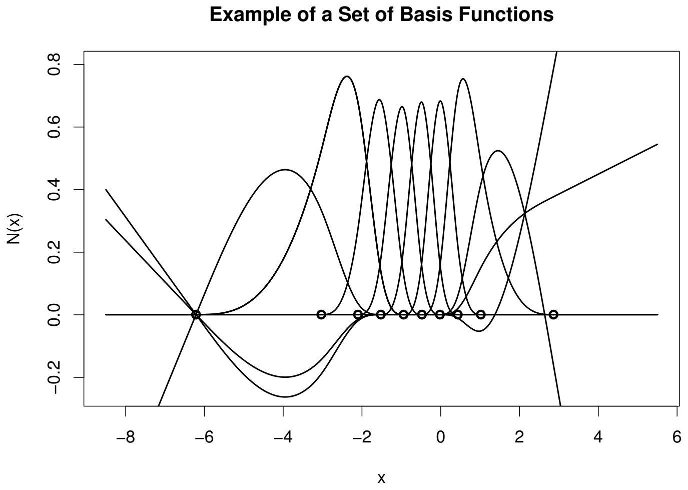

Figure 3.3: A set of basis functions that model the latent function g. The basis is uniquely determined by 10 knots marked with small circles. Once the knots are fixed, the basis can be efficiently computed following the procedure described in Appendix B.3.

with respect to $\boldsymbol{\theta}$, where $\log \mathcal{L}(\mathcal{D}|\boldsymbol{\theta})$ is given by (3.18) with g replaced by the right part of (3.26), and

$$\Omega_{ij} = \int N_i^{"}(x) N_j^{"}(x)\, dx\,. \tag{3.28}$$

The restriction that the slope of g at ξ_1 be larger or equal to zero is enforced by introducing the linear constraint

$$\mathbf{c}^{\mathrm{T}}\boldsymbol{\theta} \geq 0\,, \tag{3.29}$$

where $\mathbf{c}$ is a K-dimensional vector whose components contain the slopes of $N_1,\ldots,N_K$ at ξ_1, that is, $\mathbf{c} = \{N_1'(\xi_1),\ldots,N_K'(\xi_1)\}$. This linear constraint is easily handled by the adaptive barrier method described by Lange (1999). The non-linear optimization of (3.27) with respect to $\boldsymbol{\theta}$ is implemented numerically using the BFGS quasi-Newton algorithm (Press et al., 1992). The technical details concerning the estimation of g are given in Appendix B.5.

The bottleneck of the proposed method for the estimation of semi-parametric bivariate Archimedean copulas is the evaluation of the gradient of $\log \mathcal{L}(\mathcal{D}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ (see Appendix B.4). This operation requires to compute a total of $O(K)$ unidimensional integrals that are not analytically tractable and have to be approximated using numerical methods. Each of these primitives has to be evaluated at $O(N)$ different locations. Thus, the computational cost of the proposed method is $O(MN)$, where $M = K+1$ is the number of knots in the spline basis, N is the number of data instances and we have assumed that maximization of (3.27) involves a constant number of steps of the BFGS algorithm. In practice, the estimation process is feasible in a standard desktop computer with spline basis of tens of knots and datasets that include up to several thousand data instances.

3.4 Experiments

The performance of the proposed semi-parametric estimator for Archimedean copulas (SPAC) is investigated in experiments with simulated data, financial data (Yahoo! Finance, 2008) and precipitation data (Razuvaev et al., 2008). The datasets considered for the experiments have different dependence structures, and different levels of dependence. SPAC is compared with two other methods for modeling Archimedean copulas. The first of these methods (LAM) was proposed by Lambert (2007). This procedure is implemented following the indications of the author. In particular, a gamma hyperprior $\mathcal{G}(a,b)$ is used for τ_λ with parameters $a = b = 10^{-4}$, where τ_λ is the inverse variance of a Gaussian prior for the differences of successive coefficients in the expansion of (3.4) in terms of B-splines. Sampling from the posterior is performed after integrating τ_λ out. Metropolis-Hastings is used to generate samples (Bishop, 2006). Candidate states are sampled from a spherical Gaussian distribution centered at the current state. An initial chain is run for 1000 iterations to determine the scale of this Gaussian distribution so that the acceptance rate is close to 0.3. After this, a final chain is run for 5000 iterations. The second flexible estimator of Archimedean copulas (DIM) is the method designed by Dimitrova et al. (2008). This approach is based on the geometrically designed (GeD) splines and is implemented fixing $\alpha_{\text{exit}} = 0.9$ and $\beta = 0.5$ as recommended by Kaishev et al. (2006).

Additionally, we consider two flexible methods for bivariate copula estimation that are not of the Archimedean type. The first one is the non-parametric estimator described by Fermanian and Scaillet (2003) where Gaussian kernels (GK) are used to estimate the copula density. The second estimator is a Bayesian mixture of Gaussians (BMG) model. Both methods employ a mapping $m : [0,1]^2 \to \mathbb{R}^2$ of the data from the unit square to the Euclidean plane

$$m\left(\mathbf{x} = \{x_1, x_2\}^{\text{T}}\right) = \left\{\Phi^{-1}(x_1), \Phi^{-1}(x_2)\right\}^{\text{T}}, \tag{3.30}$$

where Φ^{-1} is the quantile function of the standard Gaussian distribution. After performing this transformation of the data, GK and BMG build an estimate $\hat{f}$ of the density in $\mathbb{R}^2$. Given $\hat{f}$, the corresponding copula density estimate is

$$\hat{c}(x_1, x_2) = \frac{\hat{f}\left[\hat{F}_1^{-1}(x_1), \hat{F}_2^{-1}(x_2)\right]}{\hat{f}_1\left[\hat{F}_1^{-1}(x_1)\right] \hat{f}_2\left[\hat{F}_2^{-1}(x_2)\right]} \tag{3.31}$$

where $\hat{f}_1$ and $\hat{f}_2$ are the marginal densities of $\hat{f}$ and $\hat{F}_1^{-1}$ and $\hat{F}_2^{-1}$ are the corresponding quantile functions. The inversion of $\hat{F}_1$ and $\hat{F}_2$ is performed by evaluating these functions on a fine grid and then interpolating the tabulated points with splines. GK is implemented using the framework for kernel density estimation developed by Duong (2007), which is included in the package *ks* of the R software environment (Team, 2007). The bandwidth matrix for the bivariate Gaussian kernel is selected using the function *Hpi*. This routine implements the two-stage plug-in method described by Duong and Hazelton (2003). BMG is implemented using the variational framework described by Attias (1999) and Bishop (2006). We employ a maximum of 10 components in the mixture and a Dirichlet prior for the mixing coefficients with $\alpha_0 = 10^{-3}$. This hyper-parameter determines the size of the resulting mixture and the value 10^{-3} is selected because it generates the best test results for this method among other alternatives. The priors for the mean and standard deviation of each Gaussian component are Gaussian-Wishart with hyper-parameters $\boldsymbol{\mu}_0 = \mathbf{0}$, $\nu_0 = 2$, $\beta_0 = 10^{-3}$ and $\mathbf{W}_0 = \mathbf{I}_{2\times 2}$.

The following experimental protocol is followed to evaluate the performance of the different copula estimation methods. For each estimation problem, the available data are split in non-overlapping training and test sets. The copula estimation techniques are applied to the training set and their log-likelihood is then computed on the corresponding test set, which is independent of the data used for model calibration. The independence between the set used for optimizing the models and the set used for evaluation ensures that the quality estimates given by the log-likelihood are not biased. The average log-likelihood obtained by each method on each problem is finally reported.

3.4.1 Experiments with Simulated Data

The accuracy of the different copula estimation methods is evaluated in experiments with data simulated from copulas that belong to the Clayton, Gumbel and Frank Archimedean families (Nelsen, 2006). We also consider samples from Gaussian copulas, Student's *t* copulas with 2 and 5 degrees of freedom and Galambos extreme value copulas (Galambos, 1975), which are not of the Archimedean type. The parameters of the copulas are selected so that Kendall's tau is 0.15, 0.3, 0.45 and 0.6. All the datasets are generated using the routines from the R package *copula*. Each combination of copula family and level of dependence constitutes a different estimation problem. For each problem, we randomly generate 100 pairs of independent training and test sets with 2000 elements each. The average log-likelihood obtained by each method in each problem is reported in Table 3.1. The third column of this table displays the values obtained by the maximum likelihood estimator (MLE), using the exact functional form of the multivariate distribution. These values should be an upper bound of the results obtained by the approximate methods and can be used for reference. For each problem, the results of the best and second-best performing methods are highlighted in boldface and underlined, respectively. The right-most column in Table 3.1 displays the p-value of a paired t test between the results of the best and second best techniques in each problem.

When the data follow an Archimedean dependence model, the best approach is SPAC, which obtains values of the likelihood that are very close to MLE. The second best method is LAM followed by DIM. The poor performance of BMG and GK is explained by the fact that these techniques do not assume an underlying Archimedean form. When the data generating copula are Gaussian, BMG obtains the best results, followed by SPAC. In this case, BMG provides a very good fit because Gaussian mixture models subsume the Gaussian copula. For a Student's *t* copula with two degrees of freedom, BMG always obtains the best performance, generally followed by GK. In this case, the poor results of SPAC are due to the fact that the Student's *t* copula with 2 degrees of freedom captures dependence structures that cannot be accurately represented by an Archimedean copula. When the degrees of freedom are increased to five, these differences between BMG and SPAC are reduced and the ranking is sometimes reversed. Finally, when the data are generated by a Galambos copula, SPAC performs the best, followed by LAM.

Note that GK obtains the worst results in most of the problems considered. The reason for this is that this method tends to overfit to the training data. Additionally, LAM always performs

Table 3.1: Average log-likelihood of each model on the simulated test data.

Copula	τ	**MLE**	**BMG**	**GK**	**DIM**	**LAM**	**SPAC**	***t*-Test**
Clayton	0.15	83.02	76.43	61.18	76.44	77.00	**81.68**	$3.7 \cdot 10^{-25}$
	0.30	314.01	305.57	288.66	306.46	308.55	**312.45**	$3.8 \cdot 10^{-21}$
	0.45	699.67	685.23	665.20	691.02	694.97	**698.09**	$3.7 \cdot 10^{-18}$
	0.60	1274.10	1251.17	1217.80	1263.04	1269.64	**1272.58**	$3.7 \cdot 10^{-20}$
Gumbel	0.15	75.66	69.31	53.48	66.90	69.20	**73.12**	$1.3 \cdot 10^{-12}$
	0.30	279.08	272.37	250.09	270.58	272.96	**276.72**	$3.0 \cdot 10^{-20}$
	0.45	598.32	590.35	563.10	588.32	593.30	**596.19**	$8.1 \cdot 10^{-13}$
	0.60	1119.84	1107.65	1065.61	1108.55	1115.38	**1117.85**	$7.2 \cdot 10^{-13}$
Frank	0.15	51.55	46.68	26.53	42.15	43.97	**49.42**	$1.7 \cdot 10^{-11}$
	0.30	209.18	197.03	177.46	200.44	201.98	**207.02**	$2.2 \cdot 10^{-29}$
	0.45	499.48	484.94	455.70	4890	492.19	**496.93**	$2.5 \cdot 10^{-23}$
	0.60	971.58	958.97	914.28	959.75	964.88	**968.58**	$4.5 \cdot 10^{-10}$
Gaussian	0.15	56.37	**54.91**	34.19	46.65	47.74	53.47	$1.1 \cdot 10^{-04}$
	0.30	226.92	**224.71**	202.88	213.92	217.04	220.73	$5.0 \cdot 10^{-13}$
	0.45	540.28	**538.76**	514.45	520.18	526.68	529.26	$1.5 \cdot 10^{-33}$
	0.60	1060.55	**1058.82**	1026.07	1034.65	1043.11	1044.79	$8.2 \cdot 10^{-45}$
t (2)	0.15	220.26	**206.31**	183.35	110.82	110.28	113.84	$8.9 \cdot 10^{-27}$
	0.30	395.01	**379.67**	348.73	293.08	294.29	296.74	$1.7 \cdot 10^{-26}$
	0.45	716.05	**698.57**	650.15	617.56	622.19	624.51	$8.8 \cdot 10^{-22}$
	0.60	1221.88	**1201.56**	1098.94	1128.26	1134.32	1136.42	$4.6 \cdot 10^{-49}$
t (5)	0.15	87.21	**77.45**	51.46	64.16	65.06	68.71	$4.5 \cdot 10^{-17}$
	0.30	266.42	**256.04**	224.44	249.95	251.38	254.55	$6.8 \cdot 10^{-02}$
	0.45	579.88	570.56	531.03	565.35	569.50	**571.62**	$1.6 \cdot 10^{-01}$
	0.60	1109.88	1098.62	1035.83	1093.79	1100.92	**1103.52**	$3.0 \cdot 10^{-11}$
Galambos	0.15	72.28	67.06	47.41	61.65	64.27	**68.89**	$3.7 \cdot 10^{-04}$
	0.30	273.99	267.93	248.94	264.46	267.02	**270.43**	$2.6 \cdot 10^{-06}$
	0.45	602.64	594.65	571.10	591.69	595.60	**598.82**	$3.4 \cdot 10^{-15}$
	0.60	1119.39	1110.81	1075.81	1107.21	1113.96	**1116.75**	$1.0 \cdot 10^{-12}$

worse than SPAC. This is due to the method used by LAM to regularize the estimate of the copula. Specifically, LAM assumes a prior on the estimate of (3.4) that penalizes the curvature of this function. However, (3.4) is typically a convex function with positive second derivative even for smooth Archimedean copulas. This leads to a biased posterior distribution for τ_λ, which incorrectly assigns too much probability to small values of this regularization parameter (Lambert, 2007). The consequence is that LAM generates copula estimates that overfit, as illustrated in Figure 3.4. By contrast, SPAC does not have this problem since g tends to be a smooth function. Thus, penalizing the curvature of g does not generally introduce a harmful bias. Finally, LAM generally outperforms DIM. The reason for this lies in the specific method used by DIM to estimate the copula function. Specifically, DIM attempts to find a GeD spline that is as close as possible to the estimate of (3.5) given by Genest and Rivest (1993). However, this approach is always less efficient than other methods based on the likelihood (for example, LAM and SPAC).

The performances of the different copula estimation methods are compared to each other using the approach of Demšar (2006). In this comparison framework, all the methods are ranked according to their performance in different tasks. Statistical tests are then applied to determine

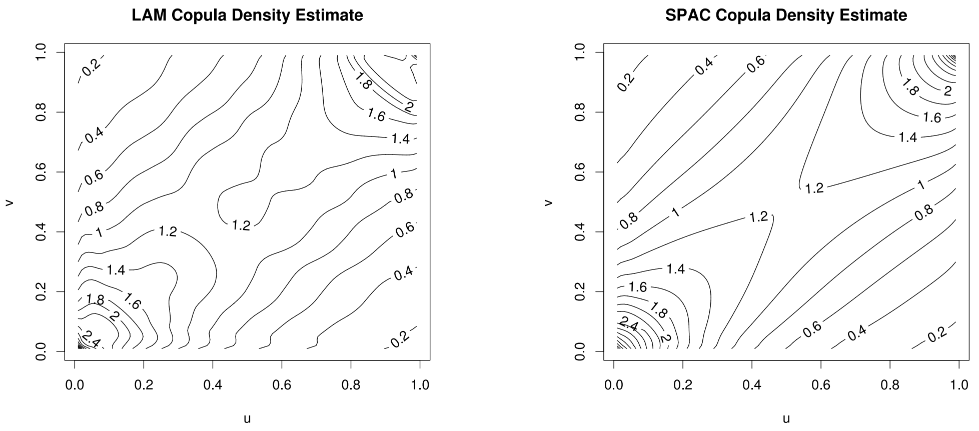

Figure 3.4: Left, copula density obtained by LAM when trained on a sample of size 2000 generated from a Frank copula with dependence level equal to 0.3 in terms of Kendall's tau. The estimate is noisy and overfits the training data. Right, copula density obtained by SPAC when β is fixed to $\exp(3)$ by 10-fold cross validation. In this case the estimate is smooth and has a lower generalization error.

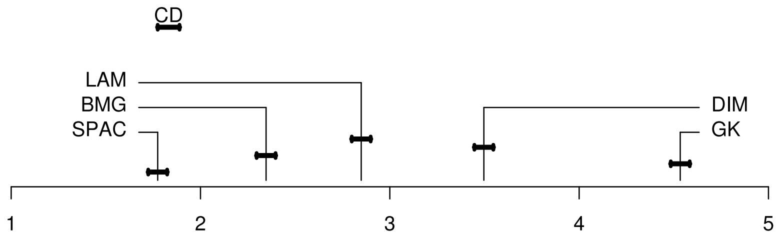

Figure 3.5: All to all comparison of the copula estimation methods by the Nemenyi test on the simulated data. The horizontal axis indicates the average ranking of each method on the set of analyzed tasks. If the differences between the average ranks of two methods is larger than the critical distance (length of the segment labeled CD) the differences in performance are statistically significant with $\alpha = 0.05$. In this case the differences in rank among all methods are statistically significant. The best results are obtained by SPAC (the proposed semi-parametric method), followed by Bayesian mixtures of Gaussians and Lambert's method (Lambert, 2007).

whether the differences among the average ranks of the methods are significant. In the experiments with simulated data, all the datasets are independent. Hence, each train and test episode can be considered as a different task in the testing framework. A Friedman rank sum test rejects the hypothesis that all methods have an equivalent performance in the $7 \times 4 \times 100 = 2800$ tasks under study (p-value = 0). Pairwise comparisons between the average ranks of the different copula estimation methods with a Nemenyi test at a 95% confidence level are summarized in Figure 3.5. The methods whose average ranks differ less than a critical distance (CD) do not show significant differences in performance at this confidence level and appear connected in the figure. The results of this test confirm that the overall performance of SPAC in the problems investigated is superior to the other methods and that the differences in rank are statistically significant.

Table 3.2 shows the average training time in seconds for each estimation method in the experiments with simulated data. The computer used for the experiments is a Dual Quad Core

Table 3.2: Average training time in seconds for each method on the simulated data.

BMG	GK	DIM	LAM	SPAC
275.65	27.99	0.26	221.49	215.50

Intel Xeon 2.5 GHz with 16GB of RAM. All the methods are implemented using the software R (Team, 2007). Note that R is an interpreted programming language and this can have an influence in the running times of the different methods. Nevertheless, the results displayed in Table 3.2 are a good approximation to the actual computational cost of each estimation technique. The most expensive method in terms of computational time is BMG. The reason for this is that BMG needs to be re-run multiple times (40 in our case) from different random initializations to ensure that a global maximum of a lower bound on the model evidence is likely to be found (Bishop, 2006). The training times of SPAC and LAM are similar. The most time-consuming computation in SPAC is the cross-validation process used to determine the value of the regularization parameter β. If an appropriate value of this parameter is known beforehand, the training time of SPAC is often less than 5 seconds. The fastest methods are GK and DIM. However, these are also the techniques that have the worst performance in terms of average rank, as shown in Figure 3.5.

3.4.2 Experiments with Financial Data

The performance of the different methods for bivariate copula estimation is also evaluated on the modeling of financial data. The techniques analyzed are those discussed in the previous subsection. Additionally, we also include in the analysis three parametric dependence models: the Gaussian copula (GC), the Student's *t* copula (ST) and the skewed Student's *t* copula (SST) described by Demarta and McNeil (2005). Maximum likelihood is used to train the parametric models. The data used in the experiments are the daily returns (increments in the logarithm of the price) of 64 components of the Dow Jones Composite Index during the period from April 13th, 2000 to March 31st, 2008. The daily closing price adjusted for dividends and splits, as published in (Yahoo! Finance, 2008), is used to compute each time series of 2000 consecutive returns, one time series per financial asset.

A distinctive characteristic of financial returns is that their distribution is time-dependent (Cont, 2001). For this reason, we focus on the estimation of the copula of their joint conditional distribution (Chen and Fan, 2006; Patton, 2006). The conditional univariate return distribution is assumed to be described by an asymmetric GARCH process (Ding et al., 1993) with an autoregressive component and innovations that follow an unspecified density. This is the time-series model (2.6) used for the experiments of Chapter 2. The model parameters and the unknown density for the innovations are estimated using the semi-parametric method described in the previous chapter (Hernández-Lobato et al., 2007). Once we know the conditional univariate distributions for two financial assets, a sample from the conditional bivariate copula is obtained by mapping each return to the unit interval using the probability integral transform (Joe, 2005). This process generates samples of size 2000 from the conditional copulas of 32 pairs of financial assets. The left-most column of Table 3.3 lists the 32 pairs of financial assets.

Each copula sample is randomly split in 100 pairs of independent training and test sets with 2/3 and 1/3 of the available data instances, respectively. In this manner, 32 estimation problems

Table 3.3: Average log-likelihood of each model on the financial test data.

Assets		τ	**ST**	**SST**	**GS**	**BMG**	**GK**	**DIM**	**LAM**	**SPAC**	t**-Test**
WMB	WMT	0.09	**6.47**	6.38	4.72	4.66	-0.81	2.97	2.78	5.97	$5.6 \cdot 10^{-01}$
KO	LSTR	0.14	11.96	11.08	11.90	11.69	7.52	10.82	12.26	**13.90**	$2.5 \cdot 10^{-12}$
FDX	FE	0.14	13.43	12.73	12.45	13.31	6.09	11.16	11.07	**14.35**	$3.9 \cdot 10^{-08}$
CHRW	CNP	0.14	12.95	12.82	13.09	13.33	9.04	13.28	14.19	**15.63**	$3.9 \cdot 10^{-11}$
EXC	EXPD	0.15	**15.98**	15.44	14.05	14.01	9.89	12.77	14.18	15.41	$2.7 \cdot 10^{-05}$
PEG	PFE	0.15	17.77	17.80	15.10	16.02	10.44	14.80	14.58	**17.80**	$9.8 \cdot 10^{-01}$
OSG	PCG	0.15	16.37	17.57	16.20	15.80	13.18	15.86	16.84	**17.90**	$1.6 \cdot 10^{-02}$
LUV	MCD	0.16	17.66	17.47	17.15	17.14	13.22	16.11	16.38	**18.21**	$2.8 \cdot 10^{-04}$
DIS	DUK	0.15	**20.99**	20.30	17.25	18.10	12.84	15.60	17.27	18.84	$6.8 \cdot 10^{-12}$
NI	NSC	0.17	20.43	19.50	18.70	18.67	14.99	17.69	19.52	**20.66**	$1.6 \cdot 10^{-01}$
AES	AIG	0.16	**21.84**	21.53	20.28	20.22	15.40	19.58	19.66	21.71	$4.0 \cdot 10^{-01}$
PG	R	0.18	**23.46**	22.80	20.24	21.76	16.76	20.10	20.14	22.89	$1.1 \cdot 10^{-03}$
FPL	GE	0.18	23.26	23.10	20.12	21.78	17.16	19.68	20.24	**23.33**	$7.1 \cdot 10^{-01}$
AA	AEP	0.17	23.28	23.33	22.36	22.11	16.52	21.31	21.67	**23.66**	$1.1 \cdot 10^{-02}$
SO	T	0.18	23.54	**24.19**	21.12	22.91	15.58	21.58	22.18	23.88	$2.1 \cdot 10^{-01}$
XOM	YRCW	0.18	23.53	23.24	22.36	22.44	16.05	22.28	22.41	**24.83**	$8.2 \cdot 10^{-13}$
MRK	MSFT	0.19	24.50	23.69	22.81	24.02	20.16	20.71	22.39	**25.65**	$4.1 \cdot 10^{-21}$
MMM	MO	0.18	24.90	24.10	24.57	24.04	19.81	21.57	22.57	**24.93**	$8.6 \cdot 10^{-01}$
D	DD	0.19	26.35	25.97	24.90	24.57	17.25	23.95	24.35	**26.37**	$8.7 \cdot 10^{-01}$
JNJ	JPM	0.18	**29.38**	29.31	23.00	28.82	24.38	24.11	24.65	27.19	$6.5 \cdot 10^{-01}$
ALEX	AMR	0.20	28.75	28.76	28.97	28.57	23.56	27.04	27.62	**29.87**	$4.5 \cdot 10^{-07}$
UTX	VZ	0.22	33.25	32.21	33.11	32.48	24.15	31.06	30.98	**33.88**	$4.4 \cdot 10^{-06}$
CAL	CAT	0.22	35.43	**35.55**	31.31	33.41	25.96	34.18	34.10	35.23	$4.2 \cdot 10^{-01}$
INTC	JBHT	0.24	42.90	42.77	41.09	42.00	42.06	41.11	42.58	**44.22**	$8.0 \cdot 10^{-08}$
GM	GMT	0.24	44.52	44.20	41.60	44.33	41.87	43.22	43.57	**45.21**	$1.5 \cdot 10^{-03}$
AXP	BA	0.25	50.03	51.47	47.40	49.96	46.07	50.23	50.86	**52.06**	$4.5 \cdot 10^{-04}$
HD	HON	0.27	**57.17**	56.13	52.55	54.69	47.07	54.36	55.30	56.84	$5.5 \cdot 10^{-02}$
BNI	C	0.27	60.55	60.43	58.39	58.58	55.56	58.34	60.25	**61.36**	$1.8 \cdot 10^{-06}$
CNW	CSX	0.31	**80.59**	80.09	75.93	77.65	71.23	77.24	79.19	80.36	$3.1 \cdot 10^{-01}$
UNP	UPS	0.32	80.63	79.90	75.21	78.72	74.53	78.49	79.38	**80.86**	$1.8 \cdot 10^{-01}$
HPQ	IBM	0.33	**90.05**	89.27	82.27	88.37	79.22	85.35	87.64	89.44	$7.2 \cdot 10^{-03}$
ED	EIX	0.33	90.99	**93.26**	86.71	93.23	88.80	89.84	91.97	93.15	$9.3 \cdot 10^{-01}$

are created, one for each pair of financial assets considered. The average test log-likelihood obtained by each method on each problem is reported in Table 3.3. The estimated value of Kendall's tau for each copula sample is given in the third column of the table. For each pair of financial assets, the results of the best and second-best performing techniques are highlighted in boldface and underlined, respectively. The right-most column in Table 3.3 displays the p-value given by a paired t test between the results of the best and second best techniques on each problem.

SPAC exhibits the best overall results and has the highest average test log-likelihood on 20 of the 32 problems. The second best method is ST with the highest rank on 9 problems, followed by SST with the largest average log-likelihood on 3 problems. By contrast, the remaining copula estimation methods perform rather poorly. To analyze the differences between SPAC and ST, Figure 3.6 shows the copula density estimates generated by these methods when trained on the complete sample (training and test sets) corresponding to the problem CHRW-CNP. SPAC yields an asymmetric copula estimate where joint losses are more likely than joint gains and both copula tails are fairly light. ST cannot capture this behavior because it is a symmetric

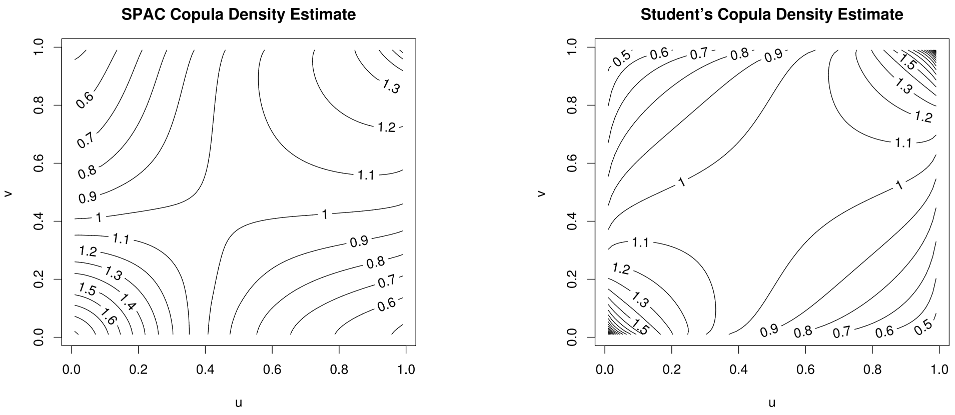

Figure 3.6: Copula density estimates obtained by SPAC (left) and ST (right) when trained on the complete sample (2000 data instances) of problem CHRW-CNP. The SPAC copula has light tails and is asymmetric, with joint losses being more likely than joint gains. The ST copula is radially symmetric with tails heavier than those of SPAC. The parameter β in SPAC is fixed to $\exp(3)$ by 10-fold cross validation. The correlation of the Student's t copula is 0.2 and the degrees of freedom are 34.67.

model. The tails, which are identical due to the symmetry of the model, are heavier than those of SPAC. The capacity of SPAC to construct copula estimates that have asymmetric tails with different degrees of heaviness is the origin of its superior performance in many of the problems considered. In spite of this limitation, ST outperforms SPAC in several problems. The reason for this is that a Student's t copula with a low number of degrees of freedom is able to capture dependencies that cannot be represented by Archimedean copulas. One might think that the SST model, which generalizes the family of Student's t copulas introducing skewness in the model (Demarta and McNeil, 2005) would improve the results because it allows to capture both non Archimedean dependence structures (like ST) and asymmetries between the two tails (like SPAC). Nonetheless, SST usually obtains worse results than SPAC and ST. The reason for this is probably that skewness does not capture well the asymmetries present in the data and consequently, accounting for skewness simply increases the overfitting problems of the method without producing any gain in estimation quality.

The performances obtained by the different methods are compared to each other following the approach proposed by Demšar (2006). This time, we consider that each estimation problem in Table 3.3 represents a different task in the testing framework. This is necessary to guarantee that all the tasks are independent. A Friedman rank sum test rejects the null hypothesis that all methods have an equivalent performance in the 32 tasks under study (p-value = $2.31 \cdot 10^{-32}$). Pairwise comparisons between the average ranks of the different copula estimation methods with a Nemenyi test at a 95% confidence level are summarized in Figure 3.7. Methods whose average ranks differ by less than a critical distance (CD) do not show significant differences in performance at this confidence level and appear connected in the figure. These results confirm that, under the set of analyzed datasets, the differences in performance between SPAC and BMG, LAM, GS, DIM or GK are statistically significant. However, there is not enough statistical evidence to discriminate between SPAC, ST and SST. The reason for this is that the Nemenyi test has little power since it considers many methods at the same time. As a more powerful approach for discriminating between a reduced number of methods, we perform two paired

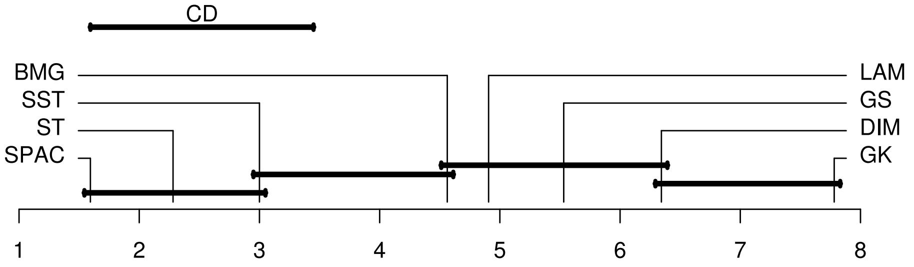

Figure 3.7: All to all comparison of the copula estimation methods by the Nemenyi test on the financial data. The horizontal axis indicates the average rank of each method on the set of analyzed tasks. If the differences between the average ranks of two methods is larger than the critical distance (length of the segment labeled CD), the differences in performance are statistically significant with $\alpha = 0.05$. Otherwise, the differences are not significant and the average ranks of the methods appear connected.

Wilcoxon tests comparing SPAC with ST and SST. The resulting p-values are 0.03 for SPAC vs. ST and 10^{-3} for SPAC vs. SST, indicating that the improvement in performance obtained by SPAC is significant at $\alpha = 0.05$.

SPAC also outperforms in these experiments the results of standard parametric Archimedean copula models corresponding to Gumbel, Clayton and Frank copulas (Nelsen, 2006). The results of these experiments can be found in Appendix B.6.

3.4.3 Experiments with Precipitation Data

We now evaluate the accuracy of the different methods for bivariate copula estimation on the modeling of precipitation data. The data consist of daily precipitation *amounts* collected at 223 stations in the former USSR over the period from 1881 to 2001 (Razuvaev et al., 2008). As reported by Kirshner (2008), the distribution of daily rainfall amounts at a given station is well represented by a non-overlapping mixture with one component corresponding to zero precipitation, and the other component corresponding to positive precipitation. Hence, we focus on modeling the marginal copula of simultaneous positive amounts of precipitation at different pairs of stations. Measurements of precipitation data are discrete because the values are rounded to the nearest tenth of a millimeter. For this reason, the marginal cumulative distribution of positive precipitation measurements at a given station is not a continuous function. Because the copula models analyzed in this work require continuous data marginals, we add to each positive precipitation measurement a random variable uniformly distributed in $[-0.05, 0.05]$. This procedure yields continuous precipitation measurements and does not have a significant impact on the underlying dependence structure of the data.

From the 223 meteorological stations we selected 32 pairs of stations so that the stations in the pair are close to each other but far from stations belonging to a different pair. Stations from different pairs are at least 100 kilometers away from each other. The selected 64 stations are represented in Figure 3.8 with their corresponding pair identification number (PIN). The two left-most columns of Table 3.4 contain, for each PIN, the World Meteorological Organization numbers (WMON) of the respective precipitation stations. For any two stations, we adopt the following protocol to obtain a sample of size 2000 from the copula of simultaneous precipitation

Table 3.4: Average log-likelihood of each model on the precipitation test data.

PIN[a]	WMON[b]	τ	ST	SST	GS	BMG	GK	DIM	LAM	SPAC	t-Test
1	36974–38353	0.08	5.33	5.95	**6.14**	5.07	-3.90	3.34	4.64	5.80	$2.4 \cdot 10^{-01}$
2	30949–30965	0.14	12.90	12.47	10.68	12.82	7.29	10.43	10.94	**13.57**	$2.7 \cdot 10^{-08}$
3	32061–32098	0.16	19.72	**21.04**	20.31	18.67	10.55	17.36	18.38	20.84	$3.3 \cdot 10^{-01}$
4	31735–31829	0.17	24.46	26.86	25.19	25.48	21.16	25.96	27.00	**27.14**	$5.7 \cdot 10^{-01}$
5	38696–38836	0.19	28.41	**29.00**	28.25	27.59	22.37	26.33	26.47	28.60	$8.0 \cdot 10^{-03}$
6	32540–32564	0.21	31.62	36.37	34.53	36.67	33.78	35.14	36.18	**38.85**	$1.2 \cdot 10^{-09}$
7	37235–37472	0.22	37.72	39.92	39.94	37.91	34.84	36.05	37.77	**40.30**	$1.2 \cdot 10^{-01}$
8	38457–38599	0.23	36.95	39.61	38.92	38.49	30.05	38.39	40.00	**42.27**	$2.0 \cdot 10^{-19}$
9	33393–33631	0.23	40.83	**44.06**	42.13	41.58	35.39	39.81	40.65	42.67	$4.3 \cdot 10^{-07}$
10	26406–26422	0.25	45.05	44.68	38.92	44.83	36.08	42.40	45.75	**47.27**	$2.6 \cdot 10^{-10}$
11	29231–29430	0.26	51.35	50.62	44.62	50.22	45.18	50.40	50.70	**53.08**	$5.8 \cdot 10^{-18}$
12	35188–35394	0.29	58.86	58.33	50.40	60.82	55.72	59.31	62.35	**62.88**	$2.3 \cdot 10^{-02}$
13	34731–34747	0.29	61.67	61.32	52.99	63.00	56.54	60.73	61.55	**65.79**	$4.3 \cdot 10^{-13}$
14	33815–33837	0.30	62.39	63.00	58.16	65.04	56.69	66.34	67.25	**69.05**	$1.9 \cdot 10^{-16}$
15	35358–35542	0.29	65.85	65.47	61.54	65.56	61.23	65.53	67.00	**67.79**	$2.6 \cdot 10^{-04}$
16	36034–36177	0.30	68.22	68.18	60.42	67.63	54.06	65.95	67.78	**69.60**	$2.7 \cdot 10^{-11}$
17	28434–28440	0.28	65.30	70.55	65.66	**71.07**	62.43	65.65	67.77	69.28	$1.1 \cdot 10^{-01}$
18	33345–33377	0.31	70.76	70.41	62.10	72.08	64.92	71.93	73.24	**74.57**	$3.3 \cdot 10^{-12}$
19	31594–31707	0.30	70.97	70.90	66.75	70.92	65.91	70.44	72.44	**73.84**	$1.6 \cdot 10^{-08}$
20	34122–34139	0.32	70.15	70.01	59.49	**78.75**	64.10	71.56	73.06	75.88	$5.6 \cdot 10^{-11}$
21	24944–24951	0.30	68.91	72.02	69.33	**74.47**	65.84	71.04	72.78	74.14	$4.5 \cdot 10^{-01}$
22	30054–30253	0.30	66.70	71.51	69.44	75.11	63.57	72.02	74.55	**75.83**	$3.1 \cdot 10^{-02}$
23	31388–31329	0.31	72.11	73.29	70.18	72.52	64.90	73.03	73.47	**74.66**	$6.6 \cdot 10^{-10}$
24	30777–30673	0.31	71.83	73.10	70.28	73.54	70.37	74.96	75.82	**77.17**	$1.1 \cdot 10^{-12}$
25	22820–22837	0.32	76.59	77.83	75.24	77.77	69.27	78.08	79.10	**81.55**	$3.9 \cdot 10^{-29}$
26	26730–26850	0.32	80.30	80.21	72.83	82.82	54.36	80.86	81.54	**84.29**	$6.6 \cdot 10^{-04}$
27	27553–27648	0.32	79.10	79.78	74.87	79.79	76.78	77.22	78.47	**81.50**	$1.2 \cdot 10^{-07}$
28	30823–30925	0.32	78.39	79.24	76.09	79.18	73.73	78.71	80.12	**82.41**	$8.8 \cdot 10^{-22}$
29	23724–23921	0.32	75.75	76.94	72.94	82.38	79.03	80.17	81.52	**84.43**	$1.8 \cdot 10^{-08}$
30	31915–31960	0.30	72.82	86.21	85.78	88.12	75.74	83.65	86.59	**89.62**	$7.6 \cdot 10^{-07}$
31	27037–27333	0.34	90.40	89.85	78.66	**92.24**	78.84	88.92	90.94	92.01	$5.9 \cdot 10^{-01}$
32	30393–31004	0.34	96.37	103.71	100.84	**105.25**	96.40	102.73	104.13	105.18	$8.2 \cdot 10^{-01}$

[a] Pair Identification Number.

[b] World Meteorological Organization Station Numbers.

amounts. First, days with zero rainfall values for at least one of the two stations are removed. Second, precipitation measurements are mapped to uniform pseudo-observations. For this, we use the marginal empirical distribution multiplied by $n/(n+1)$, where n is the number of positive precipitation measurements at the station (Genest et al., 1995). This approach is often called canonical maximum likelihood (CML). Finally, we randomly select 2000 elements from the resulting bivariate sample.

Once a copula sample of size 2000 is available for each pair of precipitation stations, the accuracy of the different methods for bivariate copula estimation is evaluated: Each copula sample is randomly split in 100 pairs of independent training and test sets with 2/3 and 1/3 of the available data instances, respectively. In this manner, we create 32 estimation problems, one for each pair of precipitation stations. The average test log-likelihood obtained by each method on each estimation problem is reported in Table 3.4. The estimated value of Kendall's tau for each copula sample is given in the third column of the table. For each pair of precipitation stations, the results of the best and second-best performing techniques are highlighted in boldface and

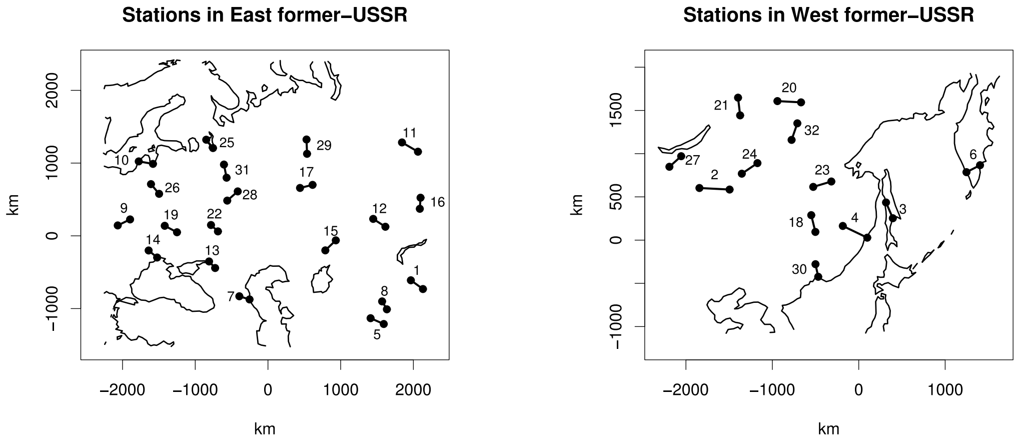

Figure 3.8: Precipitation stations in the former-USSR that are analyzed in the experiments. Each station is marked with a black dot. Stations that constitute a pair are linked with a black line and a corresponding pair identification number (PIN) is displayed. The plots were created using the R package GEOmap designed by Lees (2008).

underlined, respectively. The right-most column in Table 3.4 displays the p-value of a paired t test between the results of the best and second best techniques on each problem.

In this case, SPAC has also the best overall performance with the highest average log-likelihood in 23 of the 32 problems. The second best method is BMG with the best results in 5 problems, followed by LAM with the second rank in 12 problems. SST is the fourth best method with the highest average log-likelihood in 3 problems and the second best results in 2 problems. The other methods perform rather poorly. The good overall results obtained by SPAC are explained by its capacity to capture asymmetric dependence structures. Rainfall data are characterized by a strong positive dependence when precipitation at one of the two stations is very high: If it rains heavily at one station it is very likely that it also rains heavily at the other station. By contrast, the level of dependence is rather moderate when precipitation at one of the stations is low: If it rains slightly at one station, precipitation amounts at the other station are likely to be low, although they can be moderate or high as well. This asymmetric dependence structure is clearly captured by SPAC, as displayed in Figure 3.9. The left part of the figure shows the copula density estimate obtained by SPAC when trained on the complete sample for problem 30054-30253. The upper tail of the copula density is very heavy. This represents the joint occurrence of extreme precipitation amounts. On the other hand, the lower tail is light and the density is broader on the lower left quadrant of the unit square. This indicates that moderate and light rainfalls are less dependent. The right part of Figure 3.9 displays the corresponding copula density estimate obtained by BMG. The lower tail is again less heavy than the upper tail, but BMG is not able to capture the asymmetric dependence structure as accurately as SPAC.

The approach of Demšar (2006) is used to compare the performance of the different copula estimation methods in the problems considered. Each estimation problem in Table 3.4 represents a different task in the testing framework. A Friedman rank sum test rejects the hypothesis that all methods have an equivalent performance in the 32 tasks under study (p-value $= 3.5 \cdot 10^{-29}$). Pairwise comparisons between the average ranks of the different copula estimation methods with a Nemenyi test at a 95% confidence level are summarized in Figure 3.10. Methods whose

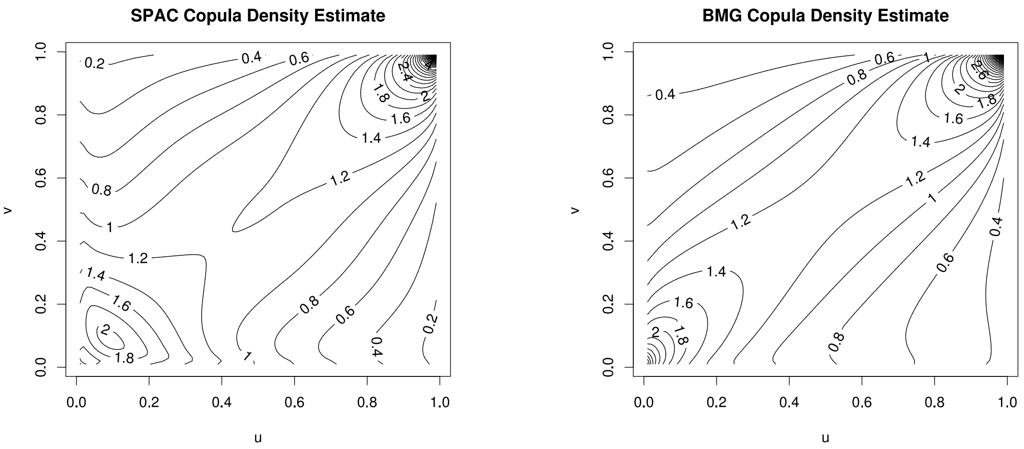

Figure 3.9: Left, copula density estimate obtained by SPAC when trained on the complete sample of problem 30054-30253. The regularization parameter β is fixed to $\exp(2)$ by 10-fold cross validation. Right, copula density estimate obtained by BMG on the same data. The method only selected two components in the Gaussian mixture.

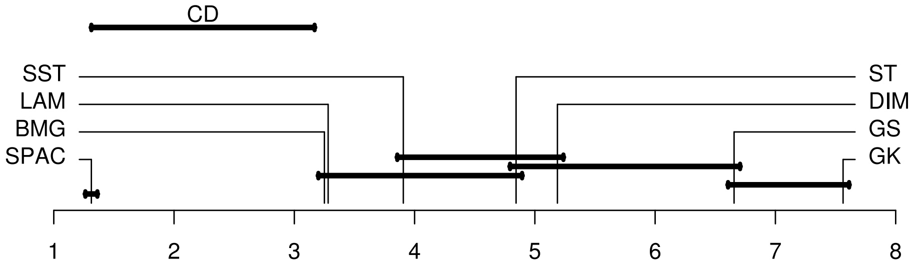

Figure 3.10: All to all comparison of the copula estimation methods by the Nemenyi test on the precipitation data. The horizontal axis indicates the average rank of each method on the set of analyzed tasks. If the differences between the average rank of two methods is larger than the critical distance (length of the segment labeled CD), the differences in performance are statistically significant with $\alpha = 0.05$. Otherwise, the differences are not significant and the average ranks of the methods appear connected.

average ranks differ less than a critical distance (CD) do not show significant differences in performance at this confidence level and appear connected in the graph. The results of Figure 3.10 confirm that, for the precipitation data, SPAC outperforms the other copula estimation techniques. The improvements over the other methods are statistically significant.

The results of SPAC are also compared with those of standard parametric Archimedean copulas corresponding to Gumbel, Clayton and Frank dependence models (Nelsen, 2006). These additional experiments confirm that SPAC is also statistically superior to standard parametric Archimedean copulas in the modeling of the precipitation data. These results are included in Appendix B.6.

3.5 Summary and Discussion

Copulas are useful tools for the construction of multivariate models. They allow to articulate the univariate marginals in a joint model with a specified dependence structure (Joe, 1997). Standard parametric copulas often lack expressive capacity to capture complex dependencies in empirical data. On the other extreme, non-parametric copulas are prone to overfitting. To overcome the limitations of these two approaches and combine their advantages, we have proposed a semi-parametric bivariate copula estimator (SPAC) based on a specification of Archimedean copulas (Nelsen, 2006) in terms of a novel latent function g. This function is defined on the real line and is in a one-to-one relationship with the Archimedean generator. Modeling g instead of the generator is easier because this function has to satisfy less stringent constraints. Additionally, the coefficients of lower and upper tail dependence of the Archimedean copula are in a one-to-one relation with the slopes of $g(x)$ as $x \to -\infty$ and $x \to \infty$, respectively. The function g is asymptotically linear. Therefore, a basis of natural cubic splines is a well-suited choice for its approximation. In addition to this, overfitting can be successfully controlled by penalizing the curvature of the functional parameter. Thus, the coefficients of the expansion of g in terms of natural cubic splines are determined by maximizing an objective function that depends on the likelihood of the model and also includes a smoothing penalty dependent on the curvature of the latent function.

SPAC is evaluated in experiments with simulated, financial and precipitation data. This method is compared with a range of techniques for bivariate copula estimation: Non-parametric copula models, standard parametric copulas and two other flexible estimators of Archimedean copulas (Dimitrova et al., 2008; Lambert, 2007). Experiments with simulated data confirm the excellent performance of SPAC when samples are generated from a copula that belongs to the Archimedean family. Even when the dependence structure of the data is not Archimedean, the quality of SPAC is fairly good. Experiments with financial and precipitation data, confirm the excellent out-of-sample performance of SPAC. The good overall results of this method can be explained by the capacity of SPAC to capture complex and asymmetric dependence structures while limiting the amount of overfitting.

An advantage of SPAC with respect to elliptical copulas is that SPAC can model complex asymmetries in the dependence structure of the data, while elliptical copulas are constrained to have radial symmetry. Elliptical copulas can be extended to model skewness (Demarta and McNeil, 2005; Genton, 2004). However, the asymmetries described by skewness are often not flexible enough to generate an accurate fit to real data. Archimedean copulas are symmetric under the exchange of variables. Therefore, when the variables under analysis play very different roles and are not exchangeable, a fully non-parametric approach should be preferred to SPAC. Extension of SPAC to higher dimensions can be implemented using the techniques described by Aas et al. (2009) and Kirshner (2008). These methods construct a d-dimensional copula using a total of $d(d-1)/2$ bivariate copulas as building blocks. Aas and Kirshner's approaches are probably preferable to standard extensions of Archimedean copulas to dimensions larger than two, which employ a single generator function (McNeil and Nešlehová, 2007). A first reason is that the constraints on the generator become more stringent as the dimension of the copula increases (McNeil and Nešlehová, 2007). Furthermore, the range of dependence structures that can be represented by an Archimedean copula in high dimensions is rather limited.

Chapter 4

Linear Regression Models with Spike and Slab Priors

The problem of Bayesian inference in the linear regression model with a spike and slab prior is considered. When this type of prior is used, exact inference is not tractable analytically or by standard quadrature algorithms. Markov Chain Monte Carlo methods, such as Gibbs sampling, can be used to address the inference problem. However, these methods, which are asymptotically exact, often require long computations to converge. In this chapter, expectation propagation (EP) is proposed as a more efficient alternative for approximate inference. The performance of EP is evaluated in regression tasks characterized by a small number of training instances and a high-dimensional feature space. The problems analyzed include the reverse engineering of transcription networks, the recovery of sparse signals and the prediction of user sentiment. EP outperforms Gibbs sampling in all these problems except in the sentiment prediction task, where both methods obtain comparable results. Furthermore, approximate inference with EP is much faster than Gibbs sampling. In the problems investigated, the linear regression model with a spike and slab prior provides more accurate predictions than other linear models that assume alternative sparsity enforcing priors, such as the Laplace prior and the degenerate Student's t prior. The good overall results obtained with the spike and slab prior can be ascribed to the superior selective shrinkage capacity of this prior distribution.

4.1 Introduction

In many regression problems of practical interest the number of training instances available for induction (n) is small and, simultaneously, the dimensionality of the data (d) is very large. Areas in which these types of problems arise include image analysis (Seeger et al., 2010), genetic microarray studies (Dudoit and Fridlyand, 2003), document processing (Sandler et al., 2008) and fMRI data modeling (van Gerven et al., 2009). To address these regression tasks, one usually assumes a simple multivariate linear model. However, when $d > n$, the calibration problem is under-determined because an infinite number of combinations of the values of the model coefficients can describe the data equally well. In many of these learning tasks only a subset

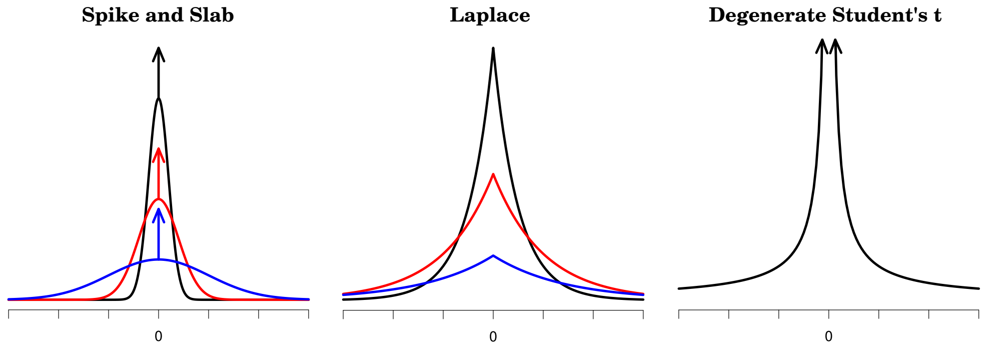

Figure 4.1: Graphs of spike and slab (left), Laplace (middle) and degenerate Student's *t* (right) priors. Spike and slab priors consist of a mixture of a Gaussian density (the slab) and a point probability mass placed at zero (the spike), which is displayed by an arrow pointing upwards. The degenerate Student's *t* is an improper prior that is obtained as the limit of a Student's *t* distribution, in which the number of degrees of freedom approaches zero. This function diverges at the origin and cannot be normalized.

of the measured features are expected to be relevant for prediction. Therefore, the calibration problem can be regularized by assuming that the vector of coefficients is sparse (Johnstone and Titterington, 2009). Different strategies can be used to obtain sparse solutions, in which most of the coefficients of the model are exactly zero. For instance, one can include in the objective function a penalty term proportional to the ℓ_1 norm of the vector of coefficients (Tibshirani, 1996). In a Bayesian approach, sparsity can be favored by using sparsity-enforcing priors for the model coefficients. These priors are characterized by probability densities that are peaked at zero and simultaneously have large probability mass in a wide range of non-zero values. This structure favors a bi-separation in the coefficients of the linear regression model: The posterior distribution of most coefficients is strongly peaked around zero. By contrast, a small subset of coefficients are assigned a large posterior probability of being significantly different from zero (Seeger et al., 2010). The fraction of coefficients whose posterior distribution is peaked at zero is the *degree of sparsity* of the model. Ishwaran and Rao (2005) call the aforementioned bi-separation effect *selective shrinkage*. Ideally, the posterior mean of truly zero coefficients should be shrunk towards zero, and the posterior mean of non-zero coefficients should be barely affected by the prior. Different sparsifying priors have been proposed in the machine learning and statistics literature. Some examples are the Laplace (Seeger, 2008), the degenerate Student's *t* (Tipping, 2001) and the spike and slab (George and McCulloch, 1997) priors. Graphs of the corresponding probability distributions are displayed in Figure 4.1.

Spike and slab priors have some advantages over Laplace and degenerate Student's *t* priors. In particular, spike and slab priors are often more effective in enforcing sparsity because they allow to selectively reduce the magnitude of only a subset of the model coefficients. Both the Laplace prior and the Student's t prior have a single characteristic scale. Consequently, they tend to reduce the magnitude of every coefficient in the model, including those coefficients that should actually be different from zero. The spike and slab distribution is a mixture model with two characteristic scales. This allows to discriminate between coefficients that are better modeled by the slab, which are not shrunk to zero, and coefficients modeled by the spike, which have large posterior probability of being exactly zero. An additional advantage is that the desired degree of sparsity in the posterior distribution is directly related to the weight assigned to the spike. Moreover, spike and slab priors are formulated in terms of a set of latent binary variables that specify whether each coefficient is assigned to the spike or to the slab. The expected value of these latent variables under the posterior distribution gives the probability that the corresponding model coefficients are exactly zero.

A disadvantage of using spike and slab priors is that Bayesian inference becomes a difficult and computationally demanding problem. Since the posterior distribution cannot be expressed in closed form, it needs to be estimated numerically. However, the computational cost of numerical algorithms is excessively large for most problems of practical interest. Therefore, inference in linear models with spike and slab priors is often implemented using Markov chain Monte Carlo (MCMC) methods; in particular, with Gibbs sampling (George and McCulloch, 1997). However, MCMC methods require to simulate very long Markov chains to obtain an accurate approximation of the posterior. The computational cost of Gibbs sampling is $O(p_0^2 d^3 k)$, where p_0 is the expected fraction of non-zero coefficients, d is the dimension of the data and k is the number of samples drawn from the posterior (see Appendix C.1). Typically, accurate inference requires $k \gg d$. This high computational cost makes Gibbs sampling infeasible when d is very large. In this chapter, expectation propagation (EP) (Minka, 2001) is proposed as an efficient alternative to Gibbs sampling. Despite the fact that EP is an approximate method, it has been shown to perform well in a linear classification model with spike and slab priors for microarray data (Hernández-Lobato et al., 2010a). The performance of the linear regression model with spike and slab priors and EP for approximate inference is evaluated in regression problems from different domains of application. The problems analyzed include the reverse engineering of transcription control networks (Gardner and Faith, 2005), the reconstruction of sparse signals (Ji et al., 2008) and the prediction of user sentiment (Blitzer et al., 2007). In these problems, EP outperforms or obtains comparable results to Gibbs sampling at a much smaller computational cost. Additionally, spike and slab priors are more effective than Laplace or Student's t priors. The improved performance of the linear regression model with a spike and slab prior is explained by the superior selective shrinkage capacity of this type of prior distribution.

This chapter is organized as follows: Section 4.2 introduces the linear regression model with a spike and slab prior (LRMSSP). Section 4.3 describes the EP algorithm and its application to the LRMSSP. This section includes a description of the posterior approximation generated by EP (Subsection 4.3.1), the EP update operations (Subsection 4.3.2) and the approximation of the evidence given by EP (Subsection 4.3.3). Section 4.4 presents an exhaustive evaluation of EP in different problems of practical interest: the reverse engineering of transcription networks (Subsection 4.4.1), the reconstruction of sparse signals (Subsection 4.4.2) and the prediction of user sentiment (Subsection 4.4.2.1). Finally, the results and conclusions of this investigation are summarized in Section 4.5.

4.2 The Linear Regression Model with a Spike and Slab Prior

In this section, we describe the linear regression model with a spike and slab prior (LRMSSP). Consider the standard linear regression problem

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \mathbf{e}\,, \tag{4.1}$$

where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\mathrm{T}$ is an $n \times d$ design matrix, $\mathbf{y} = (y_1, \ldots, y_n)^\mathrm{T}$ is a target vector, $\mathbf{w} = (w_1, \ldots, w_d)^\mathrm{T}$ is an unknown vector of regression coefficients and $\mathbf{e}$ is an n-dimensional vector that represents independent additive Gaussian noise with variance σ_0^2 (that is, $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma_0^2 \mathbf{I})$). Given $\mathbf{X}$ and $\mathbf{y}$, the likelihood for $\mathbf{w}$ is

$$\mathcal{P}(\mathbf{y}|\mathbf{w}, \mathbf{X}) = \prod_{i=1}^{n} \mathcal{P}(y_i|\mathbf{w}, \mathbf{x}_i) = \prod_{i=1}^{n} \mathcal{N}(y_i|\mathbf{w}^\mathrm{T}\mathbf{x}_i, \sigma_0^2)\,, \tag{4.2}$$

where $\mathcal{N}(\cdot|\mu, \sigma^2)$ is Gaussian density with mean μ and variance σ^2. When $d > n$, the likelihood function is not strictly concave and infinitely-many values of $\mathbf{w}$ fit the data equally well. A common approach to identify $\mathbf{w}$ in such an under-determined scenario is to assume that only a few components of $\mathbf{w}$ is are different from zero; that is, $\mathbf{w}$ is assumed to be sparse (Johnstone and Titterington, 2009). In a Bayesian approach, sparsity can be favored by assuming a spike and slab prior on $\mathbf{w}$ (George and McCulloch, 1997),

$$\mathcal{P}(\mathbf{w}|\mathbf{z}) = \prod_{i=1}^{d} \left[z_i \mathcal{N}(w_i|0, v_s) + (1 - z_i)\delta(w_i) \right]. \tag{4.3}$$

The slab $\mathcal{N}(w_i|0, v_s)$, is a zero-mean broad Gaussian whose variance v_s is large, and δ, the spike, corresponds to a point probability mass at 0. The prior is expressed in terms of a vector of binary latent variables $\mathbf{z} = (z_1, \ldots, z_d)$ such that $z_i = 0$ when $w_i = 0$ and $z_i = 1$ otherwise. To complete the specification of the prior for $\mathbf{w}$, the distribution of $\mathbf{z}$ is assumed to be a product of Bernoulli terms,

$$\mathcal{P}(\mathbf{z}) = \prod_{i=1}^{d} \mathrm{Bern}(z_i|p_0)\,, \tag{4.4}$$

where p_0 is the expected fraction of components of $\mathbf{w}$ that are different from zero and $\mathrm{Bern}(x|p) = xp + (1 - x)(1 - p)$, $x \in \{0, 1\}$ and $p \in [0, 1]$.

Given $\mathbf{X}$ and $\mathbf{y}$, the uncertainty about the values of $\mathbf{w}$ and $\mathbf{z}$ that were used to generate $\mathbf{y}$ from the design matrix $\mathbf{X}$ according to (4.1) is represented by the posterior distribution $\mathcal{P}(\mathbf{w}, \mathbf{z}|\mathbf{X}, \mathbf{y})$, which can be computed using Bayes' theorem

$$\mathcal{P}(\mathbf{w}, \mathbf{z}|\mathbf{X}, \mathbf{y}) = \frac{\mathcal{P}(\mathbf{y}|\mathbf{w}, \mathbf{X})\mathcal{P}(\mathbf{w}|\mathbf{z})\mathcal{P}(\mathbf{z})}{\mathcal{P}(\mathbf{y}|\mathbf{X})}\,, \tag{4.5}$$

where $\mathcal{P}(\mathbf{y}|\mathbf{X})$ is a normalization constant. This normalization constant is the evidence of the model and can be used to perform model selection (MacKay, 1992). The central operation in the application of Bayesian methods is the computation of marginalizations or expectations with respect to this posterior distribution. For example, given a new feature vector $\mathbf{x}^\mathrm{new}$, one can compute the probability of the associated target y^new using

$$\mathcal{P}(y^\mathrm{new}|\mathbf{X}, \mathbf{y}) = \sum_{\mathbf{z}} \int \mathcal{N}(y^\mathrm{new}|\mathbf{w}^\mathrm{T}\mathbf{x}^\mathrm{new}, \sigma_0^2)\mathcal{P}(\mathbf{w}, \mathbf{z}|\mathbf{X}, \mathbf{y})\, d\mathbf{w}\,. \tag{4.6}$$

Additionally, one can marginalize (4.5) over $w_1, \ldots, w_d$ and $z_1, \ldots, z_d$ except z_i to compute $\mathcal{P}(z_i|\mathbf{X}, \mathbf{y})$, which gives the posterior probability that the i-th component of $\mathbf{w}$ is different from zero. These probabilities can be used to identify the features (columns of $\mathbf{X}$) that are more relevant for predicting the target vector $\mathbf{y}$. Exact Bayesian inference in the LRMSSP involves computing sums and integrals that do not have a closed analytical form. Therefore, they need to be estimated numerically. However, these numerical computations are usually very costly and approximation schemes need to be used in practice. Bayesian inference in spike and slab models is usually carried out using Markov chain Monte Carlo (MCMC) approaches, and, in particular, Gibbs sampling (George and McCulloch, 1997). An efficient implementation of this method for the LRMSS is described in Appendix C.1 (Lee et al., 2003; Tipping and Faul, 2003). However, the average cost of Gibbs sampling is $O(p_0^2 d^3 k)$, where k is the number of samples drawn from the posterior and often $k \gg d$ for accurate inference. This large computational cost makes Gibbs sampling infeasible in problems with a high-dimensional feature space. As a more efficient alternative, we propose to use the expectation propagation algorithm (Minka, 2001). The application of this algorithm to the linear regression problem with a spike and slab prior is described in the following section.

4.3 Expectation Propagation in the LRMSSP

Expectation propagation (EP) is a deterministic algorithm for approximate Bayesian inference. This method approximates the joint distribution of the model parameters and the observed data by a simpler parametric distribution $\mathcal{Q}$ for which the integrals required to calculate expected values, normalization constants and marginal distributions can be expressed in closed form. The posterior distribution of the model parameters is then approximated by the normalized version of this parametric distribution, which we represent by the symbol $\mathscr{Q}$.

For many probabilistic models, the joint distribution of the observed data and the model parameters can be expressed in a factorized form. In the particular case of the LRMSSP, the joint distribution of $\mathbf{w}$, $\mathbf{z}$ and $\mathbf{y}$ given $\mathbf{X}$ can be written as the product of three different terms t_1, t_2 and t_3,

$$\mathcal{P}(\mathbf{w},\mathbf{z},\mathbf{y}|\mathbf{X}) = \prod_{i=1}^{n} \mathcal{P}(y_i|\mathbf{w},\mathbf{x}_i)\mathcal{P}(\mathbf{w}|\mathbf{z})\mathcal{P}(\mathbf{z}) = \prod_{i=1}^{3} t_i(\mathbf{w},\mathbf{z}), \tag{4.7}$$

where $t_1(\mathbf{w},\mathbf{z}) = \prod_{i=1}^{n} \mathcal{P}(y_i|\mathbf{w},\mathbf{x}_i)$, $t_2(\mathbf{w},\mathbf{z}) = \mathcal{P}(\mathbf{w}|\mathbf{z})$ and $t_3(\mathbf{w},\mathbf{z}) = \mathcal{P}(\mathbf{z})$. EP approximates each exact term t_i in (4.7) by a simpler factor $\tilde{t}_i$

$$\mathcal{P}(\mathbf{w},\mathbf{z},\mathbf{y}|\mathbf{X}) \approx \mathcal{Q}(\mathbf{w},\mathbf{z}) = \prod_{i=1}^{3} \tilde{t}_i(\mathbf{w},\mathbf{z}), \tag{4.8}$$

where all the $\tilde{t}_i$ belong to the same family of exponential distributions, except that they need not be normalized. Since exponential distributions are closed under the product operation, $\mathcal{Q}$ has the same functional form as the approximate factors $\tilde{t}_i$. Furthermore, it can be readily normalized to obtain $\mathscr{Q}$. Marginals and expectations over this approximate posterior distribution can also be computed analytically because of its simple parametric form.

Let $\mathcal{Q}^{\setminus i}(\mathbf{w},\mathbf{z})$ be the current approximation to the joint distribution with the i-th approximate term removed:

$$\mathcal{Q}^{\setminus i}(\mathbf{w},\mathbf{z}) = \prod_{j \neq i} \tilde{t}_j(\mathbf{w},\mathbf{z}) = \frac{\mathcal{Q}(\mathbf{w},\mathbf{z})}{\tilde{t}_i(\mathbf{w},\mathbf{z})}\,. \tag{4.9}$$

The EP algorithm proceeds by iteratively updating the approximate factors $\tilde{t}_i$ so that the Kullback-Leibler (KL) divergence between $t_i(\mathbf{w},\mathbf{z})\mathcal{Q}^{\setminus i}(\mathbf{w},\mathbf{z})$ and $\tilde{t}_i(\mathbf{w},\mathbf{z})\mathcal{Q}^{\setminus i}(\mathbf{w},\mathbf{z})$ is as small as possible. The version of the divergence minimized by EP includes a correction factor so that it can be applied to unnormalized distributions (Zhu and Rohwer, 1995). Specifically, each EP update operation minimizes

$$\mathrm{D}_{\mathrm{KL}}(t_i\mathcal{Q}^{\setminus i}\|\tilde{t}_i\mathcal{Q}^{\setminus i}) = \sum_{\mathbf{z}}\int\left[t_i\mathcal{Q}^{\setminus i}\log\frac{t_i\mathcal{Q}^{\setminus i}}{\tilde{t}_i\mathcal{Q}^{\setminus i}} + \tilde{t}_i\mathcal{Q}^{\setminus i} - t_i\mathcal{Q}^{\setminus i}\right]d\mathbf{w}\,, \tag{4.10}$$

with respect to the approximate factor $\tilde{t}_i$. The arguments to $t_i\mathcal{Q}^{\setminus i}$ and $t_i\mathcal{Q}^{\setminus i}$ have been omitted on the right-hand side of this formula to improve readability.

The complete EP algorithm involves the following steps:

1. Initialize all the $\tilde{t}_i$ and $\mathcal{Q}$ to be uniform (non-informative).

2. Repeat until all $\tilde{t}_i$ converge:

   (a) Select a particular factor $\tilde{t}_i$ to be refined. Compute $\mathcal{Q}^{\setminus i}$ dividing $\mathcal{Q}$ by $\tilde{t}_i$.

   (b) Update the $\tilde{t}_i$ so that $\mathrm{D}_{\mathrm{KL}}(t_i\mathcal{Q}^{\setminus i}\|\tilde{t}_i\mathcal{Q}^{\setminus i})$ is minimized.

   (c) Re-compute $\mathcal{Q}$ as the product of the newly computed $\tilde{t}_i$ and $\mathcal{Q}^{\setminus i}$.

The optimization problem in step (b) is convex and has a single global optimum. This global optimum can be found by matching sufficient statistics between $\tilde{t}_i\mathcal{Q}^{\setminus i}$ and $t_i\mathcal{Q}^{\setminus i}$ (Minka, 2001). Upon convergence, the normalized version of $\mathcal{Q}$, that is, $\mathscr{Q}$, is an accurate approximation of the posterior distribution $\mathcal{P}(\mathbf{w},\mathbf{z}|\mathbf{y},\mathbf{X})$. However, EP is not guaranteed to converge and the algorithm may end up oscillating without ever stopping (Minka, 2001). This behavior can be prevented by *damping* the EP updates (Minka and Lafferty, 2002). This is a standard procedure in many applications of the EP algorithm. Let $\tilde{t}_i^{\mathrm{new}}$ be the minimizer of the Kullback-Leibler divergence (4.10). Damping consists in using

$$\tilde{t}_i^{\mathrm{damp}} = [\tilde{t}_i^{\mathrm{new}}]^{\varepsilon}\,[\tilde{t}_i]^{(1-\varepsilon)}\,, \tag{4.11}$$

instead of $\tilde{t}_i^{\mathrm{new}}$ in step (b) of the EP algorithm. The quantity $\tilde{t}_i$ represents in (4.11) the factor before the update. $\varepsilon \in [0,1]$ is a parameter that controls the amount of damping. The original EP update operation (that is, without damping) is recovered in the limit $\varepsilon = 1$. For $\varepsilon = 0$, the approximate term $\tilde{t}_i$ is not modified during step (b).

4.3.1 The Approximation of the Posterior

In our implementation of EP for the LRMSSP, the posterior $\mathcal{P}(\mathbf{w},\mathbf{z}|\mathbf{y},\mathbf{X})$ is approximated by the product of d Gaussian and Bernoulli terms, which is a distribution in the exponential family:

$$\mathscr{Q}(\mathbf{w},\mathbf{z}) = \prod_{i=1}^{d}\mathcal{N}(w_i|m_i,v_i)\mathrm{Bern}(z_i|\sigma(p_i))\,, \tag{4.12}$$

where σ is the logistic function

$$\sigma(x) = \frac{1}{1+\exp(-x)} \tag{4.13}$$

and $\mathbf{m} = (m_1,\ldots,m_d)^{\mathrm{T}}$, $\mathbf{v} = (v_1,\ldots,v_d)^{\mathrm{T}}$ and $\mathbf{p} = (p_1,\ldots,p_d)^{\mathrm{T}}$ are free distributional parameters to be determined by the refinement of the approximate factors. The logistic function is used to improve the numerical stability of the algorithm, especially when the posterior probability of $z_i = 1$ is very close to 0 or 1 for some $i = 1,\ldots,d$. This procedure to stabilize the numerical computations is especially useful in the signal reconstruction experiments of Section 4.4.2.

The approximate factors $\tilde{t}_1$, $\tilde{t}_2$ and $\tilde{t}_3$ in (4.7) have the same form as (4.12), except that they need not be normalized:

$$\tilde{t}_1(\mathbf{w},\mathbf{z}) = \tilde{s}_1 \prod_{i=1}^{d} \exp\left\{-\frac{(w_i-\tilde{m}_{1i})^2}{2\tilde{v}_{1i}}\right\}, \tag{4.14}$$

$$\tilde{t}_2(\mathbf{w},\mathbf{z}) = \tilde{s}_2 \prod_{i=1}^{d} \exp\left\{-\frac{(w_i-\tilde{m}_{2i})^2}{2\tilde{v}_{2i}}\right\} \{z_i\sigma(\tilde{p}_{2i}) + (1-z_i)\sigma(-\tilde{p}_{2i})\}, \tag{4.15}$$

$$\tilde{t}_3(\mathbf{w},\mathbf{z}) = \tilde{s}_3 \prod_{i=1}^{d} \{z_i\sigma(\tilde{p}_{3i}) + (1-z_i)\sigma(-\tilde{p}_{3i})\}, \tag{4.16}$$

where $\{\tilde{\mathbf{m}}_i = (\tilde{m}_{i1},\ldots,\tilde{m}_{id})^{\mathrm{T}}, \tilde{\mathbf{v}}_i = (\tilde{v}_{i1},\ldots,\tilde{v}_{id})^{\mathrm{T}}\}_{i=1}^{2}$, $\{\tilde{\mathbf{p}}_i = (\tilde{p}_{i1},\ldots,\tilde{p}_{id})^{\mathrm{T}}\}_{i=2}^{3}$ and $\{\tilde{s}_i\}_{i=1}^{3}$ are free parameters to be fixed by EP. The positive constants $\{\tilde{s}_i\}_{i=1}^{3}$ are introduced to guarantee that $\tilde{t}_i\mathcal{Q}^{\setminus i}$ and $t_i\mathcal{Q}^{\setminus i}$ have the same normalization constant for $i = 1,2,3$. The parameters of (4.12), $\mathbf{m}$, $\mathbf{v}$ and $\mathbf{p}$, can be obtained from $\tilde{\mathbf{m}}_1$, $\tilde{\mathbf{m}}_2$, $\tilde{\mathbf{v}}_1$, $\tilde{\mathbf{v}}_2$, $\tilde{\mathbf{p}}_2$ and $\tilde{\mathbf{p}}_3$ using the product rule for Gaussian and Bernoulli distributions (see Appendix C.2):

$$v_i = \left[\tilde{v}_{1i}^{-1} + \tilde{v}_{2i}^{-1}\right]^{-1}, \quad m_i = \left[\tilde{m}_{1i}\tilde{v}_{1i}^{-1} + \tilde{m}_{2i}\tilde{v}_{2i}^{-1}\right] v_i, \quad p_i = \tilde{p}_{2i} + \tilde{p}_{3i} \quad i = 1,\ldots,d. \tag{4.17}$$

The first step of the EP method is to initialize the $\tilde{t}_i$ and $\mathcal{Q}$ to be non-informative by setting $\mathbf{p} = \tilde{\mathbf{p}}_{\{2,3\}} = \mathbf{m} = \tilde{\mathbf{m}}_{\{1,2\}} = (0,\ldots,0)^{\mathrm{T}}$ and $\mathbf{v} = \tilde{\mathbf{v}}_{\{1,2\}} = (\infty,\ldots,\infty)^{\mathrm{T}}$. After this, the algorithm runs sequentially over all the approximate factors, updating each $\tilde{t}_i$ so that $\mathrm{D}_{\mathrm{KL}}(t_i\mathcal{Q}^{\setminus i}\|\tilde{t}_i\mathcal{Q}^{\setminus i})$ is minimized. A cycle consists of the sequential update of all the approximate terms. The algorithm stops when the absolute value of the change in the components $\mathbf{m}$ and $\mathbf{v}$ of $\mathcal{Q}$ is less than a threshold $\delta > 0$ between two consecutive cycles. To improve the converge of EP, we use a damping scheme with a parameter ε that is initialized to 1 and then progressively annealed. After each iteration of EP, the value of this parameter is multiplied by a constant $k < 1$. The values selected for these parameters are $\delta = 10^{-4}$ and $k = 0.99$. The results obtained are not sensitive to the specific value of these two constants, provided that δ is sufficiently small and that k is close to 1. In the experiments performed, the EP algorithm converges most of the times in less than 20 cycles. Occasionally, EP takes more than 200 iterations to stop, especially when σ_0 and p_0 are very small and very few training instances are available for induction.

4.3.2 The EP Update Operations

Minimization of $\mathrm{D}_{\mathrm{KL}}(t_i\mathcal{Q}^{\setminus i}\|\tilde{t}_i\mathcal{Q}^{\setminus i})$ with respect to the parameters of the approximate factor $\tilde{t}_i$ is a convex optimization problem with a single global optimum. Since $\mathcal{Q}$ and all the $\tilde{t}_i$ belong to the same family of exponential distributions, the optimum is obtained by finding the parameters of $\tilde{t}_i$ that guarantee that the first and second moments of $\mathbf{w}$ and the first moment of $\mathbf{z}$ are the same for $t_i \mathcal{Q}^{\setminus i}$ and for $\tilde{t}_i \mathcal{Q}^{\setminus i}$ (Bishop, 2006; Minka, 2001). The derivation of the update rules that result from these moment matching constraints is given in Appendix C.3. For the sake of clarity, we present the update operations that do not consider damping. Incorporating the effect of damping is straightforward: The natural parameters of the approximate terms become a convex combination of the parameters before and after the update operation with no damping:

$$\left[\tilde{v}_{ij}^{\text{damp}}\right]^{-1} = \varepsilon \left[\tilde{v}_{ij}^{\text{new}}\right]^{-1} + (1-\varepsilon)\tilde{v}_{ij}^{-1}\,, \tag{4.18}$$

$$\tilde{m}_{ij}^{\text{damp}} \left[\tilde{v}_{ij}^{\text{damp}}\right]^{-1} = \varepsilon \tilde{m}_{ij}^{\text{new}} \left[\tilde{v}_{ij}^{\text{new}}\right]^{-1} + (1-\varepsilon)\tilde{m}_{ij}\tilde{v}_{ij}^{-1}\,, \tag{4.19}$$

$$\tilde{p}_{ij}^{\text{damp}} = \varepsilon \tilde{p}_{ij}^{\text{new}} + (1-\varepsilon)\tilde{p}_{ij}\,, \tag{4.20}$$

where $i = 1,2$ and $j = 1,\ldots,d$. The superscript *new* denotes the value of the parameter given by the full EP update with no damping. The superscript *damp* denotes the value of the parameter given by the damped update rule. The absence of a superscript refers to the parameter value before the EP update operation.

The first approximate term processed by EP is $\tilde{t}_3$. Since the corresponding exact term t_3 has the same functional form as $\tilde{t}_3$, the update operation for this approximate factor is simply $\tilde{\mathbf{p}}_3^{\text{new}} = (\sigma^{-1}(p_0),\ldots,\sigma^{-1}(p_0))^{\text{T}}$, where σ^{-1} is the logit function

$$\sigma^{-1}(x) = \log \frac{x}{1-x}\,. \tag{4.21}$$

Because this update rule does not depend on $\tilde{t}_1$ or $\tilde{t}_2$, we need to update $\tilde{t}_3$ only in the first cycle of the EP algorithm.

The second approximate factor to be processed by EP is $\tilde{t}_2$. During the first iteration of the algorithm, the update rule for $\tilde{t}_2$ is $\tilde{\mathbf{v}}_2^{\text{new}} = (p_0 v_s,\ldots,p_0 v_s)^{\text{T}}$. In successive cycles, the rule is more complex

$$\tilde{v}_{2i}^{\text{new}} = (a_i^2 - b_i)^{-1} - \tilde{v}_{1i}\,, \tag{4.22}$$

$$\tilde{m}_{2i}^{\text{new}} = \tilde{m}_{1i} - a_i(\tilde{v}_{2i}^{\text{new}} + \tilde{v}_{1i})\,, \tag{4.23}$$

$$\tilde{p}_{2i}^{\text{new}} = \frac{1}{2}\log(\tilde{v}_{1i}) - \frac{1}{2}\log(\tilde{v}_{1i} + v_s) + \frac{1}{2}\tilde{m}_{1i}^2 \left[\tilde{v}_{1i}^{-1} - (\tilde{v}_{1i} + v_s)^{-1}\right]\,, \tag{4.24}$$

for $i = 1,\ldots,d$, where a_i and b_i are given by

$$a_i = \sigma(\tilde{p}_{2i}^{\text{new}} + \tilde{p}_{3i})\frac{\tilde{m}_{1i}}{\tilde{v}_{1i} + v_s} + \sigma(-\tilde{p}_{2i}^{\text{new}} - \tilde{p}_{3i})\frac{\tilde{m}_{1i}}{\tilde{v}_{1i}}\,, \tag{4.25}$$

$$b_i = \sigma(\tilde{p}_{2i}^{\text{new}} + \tilde{p}_{3i})\frac{\tilde{m}_{1i}^2 - \tilde{v}_{1i} - v_s}{(\tilde{v}_{1i} + v_s)^2} + \sigma(-\tilde{p}_{2i}^{\text{new}} - \tilde{p}_{3i})\left[\tilde{m}_{1i}^2 \tilde{v}_{1i}^{-2} - \tilde{v}_{1i}^{-1}\right]\,. \tag{4.26}$$

The update rule (4.22) may occasionally generate a negative value for some of the $\tilde{v}_{21},\ldots,\tilde{v}_{2d}$. Negative variances in approximate factors with Gaussian functional forms are common in many EP implementations (Minka, 2001; Minka and Lafferty, 2002). When this happens, the marginals of $\tilde{t}_2$ with negative variances are not density functions. Instead, they are correction factors that compensate for the errors in the corresponding marginals of $\tilde{t}_1$. Negative variances in $\tilde{t}_2$ can lead to erratic behavior and slower convergence rates of the EP algorithm, as noted by Seeger (2008). Furthermore, when some of the components of $\tilde{\mathbf{v}}_2$ become negative, EP may fail to approximate the evidence of the LRMSSP (see the next section). To circumvent these problems, whenever (4.22) generates a negative value for $\tilde{v}_{2i}$, the update rule is modified and the corresponding marginal of $\tilde{t}_2$ is refined by minimizing $\mathrm{D}_{\mathrm{KL}}(t_2 \mathcal{Q}^{\setminus 2} \| \tilde{t}_2 \mathcal{Q}^{\setminus 2})$ under the constraint $\tilde{v}_{2i} \geq 0$. In this case, the update rules for $\tilde{m}_{2i}$ and $\tilde{p}_{2i}$ are still given by (4.23) and (4.24), but the optimal value for $\tilde{v}_{2i}$ is now infinite, as demonstrated in Appendix C.4. Thus, whenever $(a_i^2 - b_i)^{-1} < \tilde{v}_{1i}$ is satisfied, we simply replace (4.22) by $\tilde{v}_{2i}^{\text{new}} = v_\infty$, where v_∞ is a large positive constant.

The last approximate term to be refined by EP is $\tilde{t}_1$. In this case, the update rule consists of two steps. First, $\mathcal{Q}$ is refined by minimizing $\mathrm{KL}(t_1\tilde{t}_2\tilde{t}_3 \| \tilde{t}_1\tilde{t}_2\tilde{t}_3)$. Second, the parameters of $\tilde{t}_1$ are updated by computing the ratio between $\mathcal{Q}$ and $\tilde{t}_2\tilde{t}_3$. The update rule for refining $\mathcal{Q}$ is

$$\mathbf{v}^{\text{new}} = \mathrm{diag}(\mathbf{V})\,, \qquad \mathbf{m}^{\text{new}} = \mathbf{V}\left[\tilde{\mathbf{V}}_2^{-1}\tilde{\mathbf{m}}_2 + \sigma_0^{-2}\mathbf{X}^{\mathrm{T}}\mathbf{y}\right]\,, \qquad \mathbf{p}^{\text{new}} = \tilde{\mathbf{p}}_2 + \tilde{\mathbf{p}}_3\,, \tag{4.27}$$

where $\mathrm{diag}(\cdot)$ extracts the diagonal of a square matrix,

$$\mathbf{V} = (\tilde{\mathbf{V}}_2^{-1} + \sigma_0^{-2}\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1} \tag{4.28}$$

and $\tilde{\mathbf{V}}_2$ is a diagonal matrix such that $\mathrm{diag}(\tilde{\mathbf{V}}_2) = \tilde{\mathbf{v}}_2$. The calculation of $\mathrm{diag}(\mathbf{V})$ is the bottleneck of the EP method. When $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is precomputed and $n \geq d$, the computational cost of this operation is $O(d^3)$. However, when $n < d$, the Woodbury formula provides a more efficient manner to compute $\mathbf{V}$

$$\mathbf{V} = \tilde{\mathbf{V}}_2 - \tilde{\mathbf{V}}_2\mathbf{X}^{\mathrm{T}}\left[\mathbf{I}\sigma_0^2 + \mathbf{X}\tilde{\mathbf{V}}_2\mathbf{X}^{\mathrm{T}}\right]^{-1}\mathbf{X}\tilde{\mathbf{V}}_2\,. \tag{4.29}$$

The cost of EP goes down to $O(n^2d)$ in this case because it is only necessary to compute $\mathrm{diag}(\mathbf{V})$ and not $\mathbf{V}$ itself. However, the use of the Woodbury formula may lead to numerical instabilities when some of the components of $\tilde{\mathbf{v}}_2$ are very large, as reported by Seeger (2008). This limits the size of the constant v_∞ that is used for the update of $\tilde{v}_{2i}$ when (4.22) yields a negative value. We assign to v_∞ the value 100. In practice, the performance of EP does not depend strongly on the precise value of v_∞ as long as it is sufficiently large. Finally, once $\mathcal{Q}$ is refined using (4.27), the update for $\tilde{t}_1$ is obtained as the ratio between $\mathcal{Q}$ and $\tilde{t}_2\tilde{t}_3$ (see Appendix C.3):

$$\tilde{v}_{1i}^{\text{new}} = \left[(v_i^{\text{new}})^{-1} - \tilde{v}_{2i}^{-1}\right]^{-1}, \qquad \tilde{m}_{1i}^{\text{new}} = \left[m_i^{\text{new}}(v_i^{\text{new}})^{-1} - \tilde{m}_{2i}\tilde{v}_{2i}^{-1}\right]\tilde{v}_{1i}^{\text{new}}\,, \tag{4.30}$$

where $i = 1, \ldots, d$.

Finally, note that, although the approximation (4.12) of the posterior does not include any correlations between the components of $\mathbf{w}$, these correlations can be estimated very easily once the EP method has stopped. For this, we only have to compute $\mathbf{V}$ using either (4.28) when $n < d$ or (4.29) when $n \geq d$. If we are not interested in taking into account the correlations in the posterior, we may obtain a more efficient implementation of EP by decomposing t_1 and $\tilde{t}_1$ into the product of n different factors, one factor per data instance (Hernández-Lobato et al., 2008). Under such factorization, $\tilde{t}_1$ is refined by EP in n separate steps which are computationally very efficient. However, in this alternative implementation of the EP algorithm, the approximation of the posterior is less accurate.

4.3.3 The Approximation of the Model Evidence

An advantage of using a Bayesian approach to machine learning is that it provides a natural framework for model comparison and selection (MacKay, 2003). The alternative models are ranked according to the value of the model evidence, which is the normalization constant used to compute the posterior distribution from the joint distribution of the model parameters and the data. According to this framework, one should select the model with the largest value of this normalization constant. In the linear regression setting, the model evidence, $\mathcal{P}(\mathbf{y}|\mathbf{X})$, represents the probability that the targets $\mathbf{y}$ are generated from the design matrix $\mathbf{X}$ using a linear model (4.1) whose coefficient vector $\mathbf{w}$ is randomly sampled from the assumed prior distribution. This procedure naturally considers a balance between the flexibility and the robustness of the model. The model evidence favors models that provide a good fit to the training data and penalizes model complexity (Bishop, 2006; MacKay, 2003).

The exact computation $\mathcal{P}(\mathbf{y}|\mathbf{X})$ in the LRMSSP is generally infeasible because it involves averaging over the 2^d configurations for $\mathbf{z}$ and integrating over $\mathbf{w}$. However, EP can also be used to approximate the model evidence (Minka, 2001)

$$\mathcal{P}(\mathbf{y}|\mathbf{X}) \approx \sum_{\mathbf{z}} \int \tilde{t}_1(\mathbf{w},\mathbf{z})\tilde{t}_2(\mathbf{w},\mathbf{z})\tilde{t}_3(\mathbf{w},\mathbf{z})\,d\mathbf{w}\,. \tag{4.31}$$

This quantity can be computed efficiently because the factors in the approximation have a simple exponential form. Before evaluating (4.31), the constants $\tilde{s}_1$, $\tilde{s}_2$ and $\tilde{s}_3$ in (4.14), (4.15) and (4.16) need to be computed. After EP has converged, these parameters are determined by requiring that $\tilde{t}_i \mathcal{Q}^{\setminus i}$ and $t_i \mathcal{Q}^{\setminus i}$ have the same normalization constant for $i = 1$, 2 and 3

$$\begin{aligned}
\log \tilde{s}_1 =& \frac{1}{2}\mathbf{m}^{\mathrm{T}}(\tilde{\mathbf{V}}_2^{-1}\tilde{\mathbf{m}}_2 + \sigma_0^{-2}\mathbf{X}^{\mathrm{T}}\mathbf{y}) - \frac{n}{2}\log(2\pi\sigma_0^2) - \frac{1}{2}\sigma_0^{-2}\mathbf{y}^{\mathrm{T}}\mathbf{y} - \frac{1}{2}\tilde{\mathbf{m}}_2^{\mathrm{T}}\tilde{\mathbf{V}}_2^{-1}\tilde{\mathbf{m}}_2 \\
& - \frac{1}{2}\log\alpha + \frac{1}{2}\sum_{i=1}^{d}\left\{\log\left[1 + \tilde{v}_{2i}\tilde{v}_{1i}^{-1}\right] + \tilde{m}_{1i}^2\tilde{v}_{1i}^{-1} + \tilde{m}_{2i}^2\tilde{v}_{2i}^{-1} - m_i^2 v_i^{-1}\right\}, && (4.32)\\
\log \tilde{s}_2 =& \sum_{i=1}^{d}\frac{1}{2}\left\{2\log c_i + \log\left[\tilde{v}_{2i} + \tilde{v}_{1i}\right] - \log\tilde{v}_{2i} + \tilde{m}_{1i}^2\tilde{v}_{1i}^{-1} + \tilde{m}_{2i}^2\tilde{v}_{2i}^{-1}\right. \\
& \left. - m_i^2 v_i^{-1} + 2\log\left[\sigma(p_i)\sigma(-\tilde{p}_{3i}) + \sigma(-p_i)\sigma(\tilde{p}_{3i})\right] - 2\log\left[\sigma(\tilde{p}_{3i})\sigma(-\tilde{p}_{3i})\right]\right\}, && (4.33)\\
\log \tilde{s}_3 =& 0\,, && (4.34)
\end{aligned}$$

where $c_i = \sigma(\tilde{p}_{3i})\mathcal{N}(0|\tilde{m}_{1i}, \tilde{v}_{1i} + v) + \sigma(-\tilde{p}_{3i})\mathcal{N}(0|\tilde{m}_{1i}, \tilde{v}_{1i})$, $\alpha = |\mathbf{I} + \sigma_0^{-2}\tilde{\mathbf{V}}_2\mathbf{X}^{\mathrm{T}}\mathbf{X}|$. Logarithms are used to avoid numerical underflow or overflow errors in the practical implementation of EP. The derivation of these formulas is described in Appendix C.3. Sylvester's determinant theorem provides a more efficient representation for α when $n < d$, that is, $\alpha = |\mathbf{I} + \sigma_0^{-2}\mathbf{X}\tilde{\mathbf{V}}_2\mathbf{X}^{\mathrm{T}}|$. Finally, by taking logarithms on both sides of (4.31), $\log \mathcal{P}(\mathbf{y}|\mathbf{X})$ can be approximated as

$$
\begin{aligned}
\log \mathcal{P}(\mathbf{y}|\mathbf{X}) \approx \log \tilde{s}_1 + \log \tilde{s}_2 + \frac{d}{2}\log(2\pi) \\
+ \sum_{i=1}^{d} \frac{1}{2}\left\{ \log v_i + m_i^2 v_i^{-1} - \tilde{m}_{1i}^2 \tilde{v}_{1i}^{-1} - \tilde{m}_{2i}^2 \tilde{v}_{2i}^{-1} \right\} \\
+ \sum_{i=1}^{d} \log \left\{ \sigma(\tilde{p}_{2i})\sigma(\tilde{p}_{3i}) + \sigma(-\tilde{p}_{2i})\sigma(-\tilde{p}_{3i}) \right\},
\end{aligned}
\tag{4.35}
$$

where $\log \tilde{s}_1$ and $\log \tilde{s}_2$ are given by (4.32) and (4.33). This formula has been obtained using the product rules for Gaussian and Bernoulli distributions (see Appendix C.2). Note that α can be negative if some of the components of $\tilde{\mathbf{v}}_2$ are negative. In this case, $\mathbf{I} + \sigma_0^{-2}\tilde{\mathbf{V}}_2\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is not positive definite, $\log \tilde{s}_1$ cannot be evaluated and EP fails to approximate the model evidence. To avoid this, we refine $\tilde{t}_2$ by minimizing $\mathrm{D}_{\mathrm{KL}}(t_2 \mathcal{Q}^{\setminus 2} \| \tilde{t}_2 \mathcal{Q}^{\setminus 2})$ under the constraint that the components of $\tilde{\mathbf{v}}_2$ are positive, as described in the previous section.

4.4 Experiments

In this section, the performance of EP for approximate inference in the LRMSSP is evaluated on several regression problems from different domains of application, using both simulated and real-world data. The problems investigated include the reverse engineering of transcription networks from gene expression data (Gardner and Faith, 2005), the reconstruction of sparse signals from a reduced number of measurements slightly contaminated by noise (Ji et al., 2008) and the prediction of user sentiment from user-written reviews of kitchen appliances and books (Blitzer et al., 2007). These specific learning problems were selected according to the following criteria. First, all the analyzed datasets are characterized by a high-dimensional feature space and small numbers of training instances ($d > n$). Second, only a reduced number of features are expected to be relevant for prediction, so that the optimal solutions should be sparse or nearly sparse. Finally, the selected datasets belong to application domains of current interest; namely, the modeling of gene expression data (Slonim, 2002), compressive sensing (Donoho, 2006) and statistical processing of natural language (Manning and Schütze, 2000).

In these experiments, the LRMSSP with EP for approximate inference is compared with other benchmark techniques for Bayesian inference in sparse linear regression models. These include a) the LRMSSP and Gibbs sampling for approximate inference, which is described in Appendix C.1, b) the sparse linear regression model proposed by Seeger (2008) and c) the relevance vector machine (RVM) of Tipping (2001). Approaches b) and c) use sparsity enforcing priors different from the spike and slab model; namely, Laplace and degenerate Student's t priors, respectively. In b) the posterior distribution is approximated by a multivariate Gaussian distribution whose parameters are determined using EP. Case c) also approximates the posterior by a multivariate Gaussian, whose parameters are determined by a type-II maximum likelihood approach. An interpretation of c) from a variational point of view is given by Wipf et al. (2004). All the methods are efficiently coded in the R environment (Team, 2007) except for the RVM method, which was coded in Matlab by Ji et al. (2008).

4.4.1 Reconstruction of Transcription Regulatory Networks

The LRMSSP can be a useful method for the reconstruction of genetic regulatory networks from gene expression data. Transcription control networks are a specific class of interaction networks in which each node corresponds to a different gene and each connection represents an interaction between two genes at the transcription level (Alon, 2006). Specifically, the directed edge $Z \rightarrow Y$ encodes the information that the protein expressed by gene Z has a direct effect on the transcription rate of gene Y. Michaelis-Menten interaction kinetics and the Hill equation can be used to characterize this network edge as a differential equation (Alon, 2006). Assuming that Z is a transcriptional activator, the equation that describes the regulation kinetics is

$$\frac{d\,[Y]}{dt} = \frac{V_m\,[Z]^{\alpha}}{[Z]^{\alpha} + K_A} - \delta\,[Y]\,. \tag{4.36}$$

When Z is a transcriptional repressor the evolution of $[Y]$ is described by

$$\frac{d\,[Y]}{dt} = \frac{V_m K_R}{K_R + [Z]^{\beta}} - \delta\,[Y]\,. \tag{4.37}$$

In these equations, K_A and K_R are the activation and repression thresholds, respectively, α and β are the Hill coefficients for cooperative binding, V_m is the maximum rate of synthesis, $[\cdot]$ stands for *'concentration of mRNA'* and δ is the rate of mRNA degradation. The concentration of mRNA $[Z]$ is assumed to be a measure of the activity of the protein product of gene Z. When the system achieves a steady-state, and assuming that, in this state, the concentrations of mRNA are far from saturation, the relation between the logarithm of the mRNA concentration of Y and the logarithm of the mRNA concentrations of $Z_1,\ldots,Z_k$, that is, the parents of Y in the transcription network, is approximately linear (Gardner and Faith, 2005):

$$\log\,[Y] \approx \sum_{i=1}^{k} w_i \log\,[Z_i] + \text{constant}\,. \tag{4.38}$$

In this derivation, transcriptional activation and repression are assumed to be simultaneously possible. When Y is a self-regulating gene, $\log[Y]$ is included in the right part of (4.38) with associated coefficient w_{k+1}. However, this autoregulatory term can be eliminated by replacing $w'_i = w_i/(1 - w_{k+1})$ for w_i, where $i = 1,\ldots,k$, and setting $w'_{k+1} = 0$.

This linear model can be readily extended to describe the kinetics of all the transcripts present in a particular biological system. Let $\mathbf{X}$ denote a $d \times n$ matrix whose rows correspond to different genes and whose columns represent values of the logarithm of the concentration of mRNA obtained under different steady-state conditions. The rows of $\mathbf{X}$ are centered so that they have zero mean. Assuming that the components of $\mathbf{X}$ are contaminated with additive Gaussian noise, (4.38) suggests that $\mathbf{X}$ should approximately satisfy

$$\mathbf{X} = \mathbf{W}\mathbf{X} + \sigma_0 \mathbf{E} \tag{4.39}$$

where $\mathbf{W}$ is a $d \times d$ matrix of linear regression coefficients that connects each gene with its transcriptional regulators, $\mathbf{E}$ is a $d \times n$ random matrix whose elements are independent and follow a standard Gaussian distribution and σ_0 is a positive constant that measures the level of

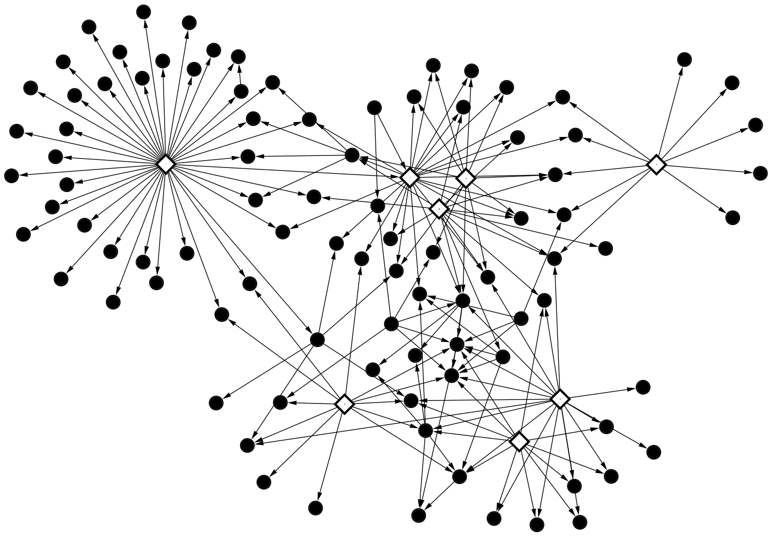

Figure 4.2: A transcription regulatory network with 100 nodes. Each node in the network represents a different gene. Edges represent transcriptional interactions between genes. The network was generated using the software GeneNetWeaver (Marbach et al., 2009). Hub genes are displayed in the network with a diamond shape node.

noise in **X**. The diagonal of **W** can be set to zero because any autoregulatory term in (4.39) can be eliminated using the transformation described above. Assuming (4.39), the likelihood of **W** given **X** and σ_0 is

$$\mathcal{P}(\mathbf{X}|\mathbf{W}) = \prod_{i=1}^{d}\prod_{j=1}^{n}\mathcal{N}(x_{ij}|\mathbf{w}_i\mathbf{x}_j,\sigma_0^2)\,, \tag{4.40}$$

where x_{ij} is the element in the i-th row and j-th column of **X**, $\mathbf{x}_j$ is the j-th column of **X** and $\mathbf{w}_i$ is the i-th row of **W**. To complete a Bayesian description for (4.39), we must specify a prior for **W**. The prior on **W** should reflect the connectivity of the network. Therefore, the element in the i-th row and j-th column of **W** should be non-zero ($w_{ij} \neq 0$) if there is a link from gene j to gene i and $w_{ij} = 0$ otherwise.

Figure 4.2 displays an example of a realistic transcription control network, generated with the application GeneNetWeaver (Marbach et al., 2009). Most genes in the network have only a few parents. There are also a few hub genes that are connected to a large number of nodes (Barabási and Oltvai, 2004; Thieffry et al., 1998). Thus, the connectivity matrix **W** is expected to be sparse. The non-zero elements of **W** are clustered and appear in the columns of this matrix corresponding to hub genes. In Chapter 6 (Hernández-Lobato et al., 2008), we will introduce a linear model that takes into account the *column-wise* sparsity of **W** to identify transcriptional regulators. Taking into account this specific structure of **W** can also improve the performance of network reconstruction methods (Hernández-Lobato and Dijkstra, 2010). Nevertheless, in the current chapter, the components of **W** are assumed to be *a priori* independent. The prior for **W** is then a product of spike and slab factors

$$\mathcal{P}(\mathbf{W}|\mathbf{Z}) = \prod_{i=1}^{d}\prod_{j=1}^{d}\left[z_{ij}\mathcal{N}(w_{ij}|0,v_s) + (1-z_{ij})\delta(w_{ij})\right]\,, \tag{4.41}$$

where $\mathbf{Z}$ is a $d \times d$ matrix of binary latent variables, $z_{ij} = \{0,1\}$ is the element in the i-th row and j-th column of $\mathbf{Z}$ and v_s is the prior variance of the components of $\mathbf{W}$ that are different from zero. Note that $z_{ij} = 1$ whenever there is an edge in the network from gene j to gene i and $z_{ij} = 0$ otherwise. The prior for $\mathbf{Z}$ is given by a product of Bernoulli terms:

$$\mathcal{P}(\mathbf{Z}) = \prod_{i=1}^{d}\prod_{j=1}^{d} \mathrm{Bern}(z_{ij}|p_{ij}), \tag{4.42}$$

where $p_{ij} = p_0$ for $i \neq j$, $p_{ij} = 0$ for $i = j$ and p_0 corresponds to the expected fraction of regulators of a given gene in the network. Finally, the posterior for $\mathbf{W}$ and $\mathbf{Z}$ is obtained using Bayes theorem:

$$\mathcal{P}(\mathbf{W},\mathbf{Z}|\mathbf{X}) = \frac{\mathcal{P}(\mathbf{X}|\mathbf{W})\mathcal{P}(\mathbf{W}|\mathbf{Z})\mathcal{P}(\mathbf{Z})}{\mathcal{P}(\mathbf{X})} = \prod_{i=1}^{d} \frac{\mathcal{P}(\mathbf{x}_i|\mathbf{w}_i)\mathcal{P}(\mathbf{w}_i|\mathbf{z}_i)\mathcal{P}(\mathbf{z}_i)}{\mathcal{P}(\mathbf{x}_i)}, \tag{4.43}$$

where $\mathbf{x}_i$, $\mathbf{w}_i$ and $\mathbf{z}_i$ represent the i-th rows of $\mathbf{X}$, $\mathbf{W}$ and $\mathbf{Z}$, respectively and the right-most part of (4.43) reflects the fact that the posterior factorizes in the rows of $\mathbf{W}$ and $\mathbf{Z}$. The i-th factor in the right part of (4.43) ($i = 1,\ldots,d$) is the posterior distribution of a LRMSSP for the log-concentration of mRNA of gene i. To reconstruct the underlying transcription network, we compute the posterior probability of each possible link. For an edge from gene j to gene i, this probability is given by $\mathcal{P}(z_{ij} = 1|\mathbf{X})$, which is computed by marginalizing (4.43) with respect to $\mathbf{W}$ and all the components of $\mathbf{Z}$ except z_{ij}. Once the posterior probability of each possible connection has been computed, we fix a threshold $0 \leq \gamma \leq 1$ and predict a connection from gene j to gene i whenever $\mathcal{P}(z_{ij} = 1|\mathbf{X}) > \gamma$. However, the marginalization of (4.43) is not practicable. Because (4.43) factorizes into d linear regression problems, we can approximate the posterior of each of these problems using EP. The product of the resulting d approximations generates a final approximation of (4.43), which allows us to compute the posterior edge probabilities very efficiently.

Instead of the spike and slab model, one can use different sparsity enforcing priors for the elements of $\mathbf{W}$. An example is the Laplace distribution, which is often used to enforce sparsity in linear regression problems. For instance, the *lasso* can be given a Bayesian interpretation as the *maximum a posteriori* solution of the linear regression problem when Laplace priors are assumed (Tibshirani, 1996). The Laplace prior for $\mathbf{W}$ is

$$\mathcal{P}(\mathbf{W}) = \left[\prod_{i=1}^{d}\prod_{j=1,j\neq i}^{d} \frac{1}{2b_0}\exp\left\{-\frac{|w_{ij}|}{b_0}\right\}\right]\prod_{k=1}^{d}\delta(w_{kk}), \tag{4.44}$$

where b_0 determines the expected size of the non-diagonal components of $\mathbf{W}$. The diagonal elements of $\mathbf{W}$ are constrained to be zero by the delta functions. Following Steinke et al. (2007), the posterior probability of a connection from gene j to gene i is approximated by the probability of the event $|w_{ij}| > \delta_e$ under the posterior for $\mathbf{W}$, where δ_e is a small positive constant. To compute this probability $\mathcal{P}(\mathbf{W}|\mathbf{X})$ has to be integrated in the set of possible values of $\mathbf{W}$ such that $w_{ij} < -\delta_e$ and $w_{ij} > \delta_e$. Once again, the exact computations are not feasible and we have to resort to approximate inference. The Laplace prior also yields a posterior distribution that factorizes into d linear regression problems. In each of these separate problems, the posterior can be approximated using the EP method described by Seeger (2008).

Another way of favoring sparsity in $\mathbf{W}$ is to use the degenerate Student's t prior, as in the Relevance Vector Machine (RVM) (Tipping, 2001). In this case, the posterior distribution for $\mathbf{W}$ is approximated by solving d different regression problems with RVM. In each of these problems, the log-concentration of mRNA for gene i is expressed as a linear combination of the log-concentration of mRNA of the other genes plus Gaussian noise, for $i = 1,\ldots,d$. The global posterior is approximated as the product of the d multivariate Gaussians that are solutions of the surrogate regression problems. Finally, the posterior probability of a network link from gene j to gene i is approximated by the posterior probability of $|w_{ij}| > \delta_e$ with δ_e a small positive constant as recommended by Steinke et al. (2007).

4.4.1.1 DREAM 4 Multifactorial Sub-challenge

The performance of EP for approximate inference in the LRMSSP is evaluated in the task of reverse engineering transcription control networks following an experimental protocol based on the DREAM 4 (2009) multifactorial sub-challenge. The Dialogue for Reverse Engineering Assessments and Methods (DREAM) is an annual conference where researchers evaluate their methods on a set of common network reconstruction challenges (Stolovitzky et al., 2007). The DREAM 4 multifactorial sub-challenge includes 100 steady-state measurements obtained from networks with 100 genes. The expression of these genes is measured under different perturbed conditions. The perturbations consist in small random changes in the basal activation of the different genes in the network. The network structures and the gene expression measurements are simulated using the program GeneNetWeaver (Marbach et al., 2009). In our experiments, GeneNetWeaver is used to generate 100 networks of size 100 and to sample 100 steady-state measurements from each transcription control network. Figure 4.2 displays one of the networks generated by GeneNetWeaver.

The posterior edge probabilities are approximated for each of the 100 networks generated by GeneNetWeaver. For this task, we use linear regression models with: Spike and slab priors and Gibbs sampling (SS-MCMC), Laplace priors and EP (Laplace-EP), the degenerate Student's t prior and a type-II maximum likelihood approach (RVM) and finally, spike and slab priors and EP (SS-EP). The rows of $\mathbf{X}$ are always standardized so that they have zero mean and unit standard deviation. Each gene is assumed to be regulated on average by a fraction of $p_0 = 0.02$ regulators. For the Laplace model, δ_e and b_0 are fixed following the heuristic rule proposed by Steinke et al. (2007), that is, $\delta_e = 0.1$ and $b_0 = -\delta_e/\log p_0$. In the spike and slab model, v_s is selected so that $\mathbf{W}$ has the same marginal variances, *a priori*, than in the Laplace model, namely, $v_s = 2b_0^2 p_0^{-1}$. Microarray data frequently include a considerable amount of noise. For this reason, we take a conservative approach and select $\sigma_0 = 1$ in all the models. The Gibbs sampling approach draws 10,000 samples from the posterior distribution after a burn-in period of 1000 samples. The performance of the different methods is evaluated using the area under the precision recall (PR) and receiver operating characteristics (ROC) curves which are obtained when γ is varied from 0 to 1 (Davis and Goadrich, 2006). The estimates of the model evidence given by the EP and RVM methods are also used to discriminate among the different Bayesian models. Finally, we also report the training time in seconds of each approach.

Table 4.1 displays the results obtained by each technique in the experiments with the gene expression data generated by GeneNetWeaver. The rows in this table present the average and

Table 4.1: Results for each method in the network reconstruction problem.

	SS-MCMC	**Laplace-EP**	**RVM**	**SS-EP**
AUC-PR	19.0 ± 3.4	14.9 ± 2.3	14.3 ± 3.1	19.4 ± 3.5
AUC-ROC	75.3 ± 3.6	75.1 ± 3.3	64.0 ± 2.6	75.7 ± 3.5
log $\mathcal{P}(\mathbf{X})$		-13,774 ± 123	-12,466 ± 164	-13,450 ± 179
Time	9041 ± 127	4.7 ± 1.2	8.7 ± 2.3	7.4 ± 2.1

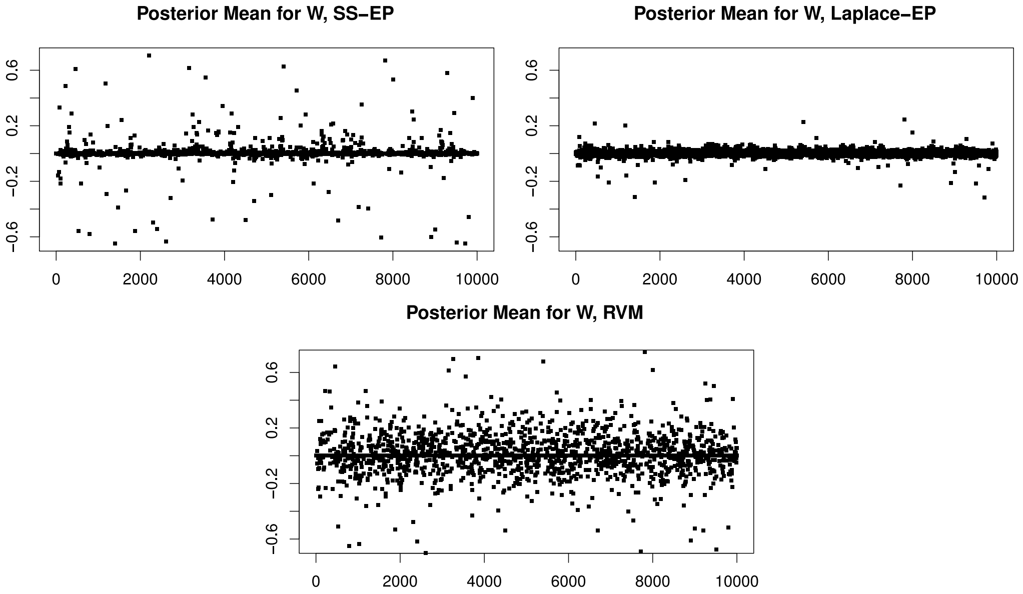

Figure 4.3: Approximations for the posterior mean of **W** given by the spike and slab EP model (top left), the Laplace EP model (top right) and the RVM method (bottom) on a specific instance of the network reconstruction problem. The approximation given by Gibbs sampling (not shown) is very close to the one generated by EP.

the standard deviation of the area under the PR and ROC curves, the logarithm of the model evidence and the training time in seconds for each method. The best reconstruction performance is obtained by SS-EP. The improvements with respect to the other techniques are statistically significant at $\alpha = 5\%$ according to a paired t test. The resulting p-values are below $2 \cdot 10^{-10}$. The evidence of SS-EP is larger than the evidence of Laplace-EP. RVM obtains the highest average evidence in all cases. However, the estimates of the model evidence given by RVM are unreliable high. The reason for this is that the type-II maximum likelihood approach used in RVM generates a posterior approximation in which many of the model coefficients are exactly zero with probability one. The uncertainty in the value of these coefficients is not taken into account and RVM tends to overestimate the the value of the model evidence. Regarding training times, the EP methods and RVM obtain similar results, while the Gibbs sampling approach is of the order of 1000 times slower.

The improved results of the spike and slab model with respect to the other methods reflect the superior selective shrinkage capacity of the spike and slab prior distribution. Figure 4.3 illustrates this by displaying plots of the approximations of the posterior mean for **W** generated by the spike and slab EP model (top left), the Laplace EP model (top right) and the RVM method (bottom) in a specific instance of the network reconstruction task. The corresponding plot for the Gibbs sampling approach is not shown since it cannot be visually distinguished from the one generated by EP. The 100×100 matrices are represented as vectors of dimension $10{,}000$. Each point in the plots represents the posterior mean of a different coefficient. In the spike and slab model, most coefficients are strongly shrunk towards zero while a few of them take very large values. By contrast, in the Laplace model this shrinkage effect is less pronounced for small coefficients while large coefficients are excessively compressed. This result cannot be circumvented by increasing the sparsity level of the Laplace prior, that is, by reducing the value of hyper-parameter b_0, since that generates a general increment in the shrinkage of all the model coefficients, including those coefficients whose values should be large. Finally, the RVM method includes too many coefficients that are considerably different from zero, a clear symptom of overfitting.

4.4.2 Reconstruction of Sparse Signals

The LRMSSP has potential applications in signal processing and in particular, in compressive sensing (Candès, 2006; Donoho, 2006). The goal in compressive sensing is to recover a sparse signal $\mathbf{w} = (w_1, \ldots, w_d)^{\mathrm{T}}$ from a limited set of linear measurements $\mathbf{y} = (y_1, \ldots, y_n)^{\mathrm{T}}$, where $n < d$. The measurements $\mathbf{y}$ are obtained after projecting the signal $\mathbf{w}$ onto an $n \times d$ measurement matrix $\mathbf{X}$, that is, $\mathbf{y} = \mathbf{X}\mathbf{w} + \mathbf{e}$, where $\mathbf{e} = (e_1, \ldots, e_n)^{\mathrm{T}} \sim \mathcal{N}(\mathbf{0}, \sigma_0^2\mathbf{I})$ is additive Gaussian noise. Since $\mathbf{w}$ is sparse, it is possible to reconstruct this vector accurately from $\mathbf{y}$ and $\mathbf{X}$ using fewer measurements than the number of degrees of freedom of the signal, which is the limit imposed by the Nyquist sampling theorem to guarantee the reconstruction of general signals. When $\mathbf{w}$ is not sparse, we may find a $d \times d$ orthonormal matrix $\mathbf{B}$, for example a wavelet basis, such that $\tilde{\mathbf{w}} = \mathbf{B}^{\mathrm{T}}\mathbf{w}$, where $\tilde{\mathbf{w}}$ is sparse or nearly sparse. In this case, the measurement process is performed after projecting the signal onto the columns of $\mathbf{B}$, that is, $\mathbf{y} = \mathbf{X}\mathbf{B}^{\mathrm{T}}\mathbf{w} + \mathbf{e} = \mathbf{X}\tilde{\mathbf{w}} + \mathbf{e}$. Once an estimate of $\tilde{\mathbf{w}}$ is obtained from $\mathbf{y}$ and $\mathbf{X}$, we can approximate $\mathbf{w}$ using $\mathbf{w} = \mathbf{B}\tilde{\mathbf{w}}$. Therefore, even when the signal is not sparse, it may still be possible to reconstruct $\mathbf{w}$ with high precision using less than d samples, provided that this vector is compressible in some basis $\mathbf{B}$.

In summary, the reconstruction of a sparse signal from a reduced number of compressive measurements is a linear regression task in which $\mathbf{y}$ is the target vector, $\mathbf{X}$ is the design matrix and the vector of regression coefficients $\mathbf{w}$ (the signal) is assumed to be sparse. Therefore, the EP algorithm for approximate inference in the LRMSSP introduced in this chapter (SS-EP) can be used to address this problem. The performance of SS-EP is evaluated in a series of experiments on the reconstruction of non-uniform and uniform spike signals. These tasks have been used as benchmarks for comparison in the compressive sensing literature (Ji et al., 2008).

4.4.2.1 Non-uniform Spike Signals

In this experiment, 100 signals of length $d = 512$ are generated by randomly selecting 20 non-zero components in each signal vector. The elements in these positions are then independently sampled from a standard Gaussian distribution. All the other elements in the signal vectors are zero. It is not necessary to determine an appropriate $\mathbf{B}$ because the signals are already sparse. The measurements are performed using a matrix $\mathbf{X}$ whose rows are sampled uniformly from the unit hypersphere. For the reconstruction of the signals, a total of $n = 75$ measurements are used.

Table 4.2: Results for each method in the non-uniform spike signal reconstruction problem.

	SS-MCMC	**Laplace-EP**	**RVM**	**SS-EP**
Error	0.19±0.37	0.82±0.06	0.19±0.36	0.04±0.11
log $\mathcal{P}(\mathbf{y}\vert\mathbf{X})$		19.7±11.2	219±25	122±27
Time	798±198	0.12±0.01	0.07±0.02	0.19±0.11

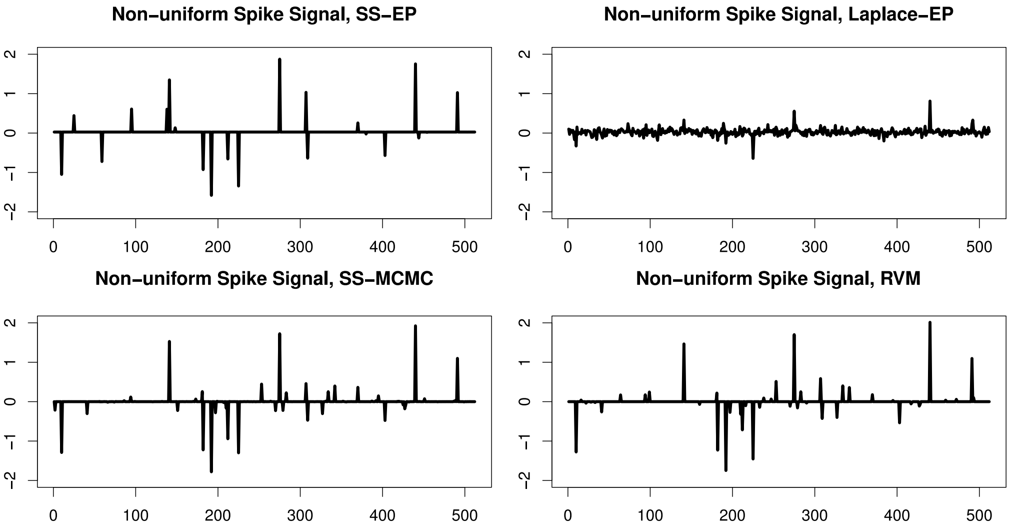

Figure 4.4: Signal estimates generated by each reconstruction method on a particular instance of the non-uniform spike signal problem. The original signal (not shown) cannot be visually distinguished with the approximation generated by EP in the spike and slab model.

Noise in the measurement process follows a zero-mean Gaussian distribution with standard deviation 0.005. The signal is approximated by the posterior mean of $\mathbf{w}$. The following methods for computing the posterior are compared: The LRMSSP with Gibbs sampling (SS-MCMC), the LRMSSP with EP (SS-EP), the linear model with Laplace prior and EP (Laplace-EP) and the RVM. The values of the different hyper-parameters are selected optimally. In the LRMSSP, $p_0 = 20/512$, $v_s = 1$ and $\sigma_0 = 0.005$. In Laplace-EP, the scale parameter is $b_0 = \sqrt{10/512}$. The variance of the noise in Laplace-EP and RVM is 0.005^2. SS-MCMC draws 10,000 samples from the posterior distribution using Gibbs sampling after a burn-in period with 1000 samples. Given an estimate $\hat{\mathbf{w}}$ of a signal $\mathbf{w}_0$, the reconstruction error of $\hat{\mathbf{w}}$ is quantified by $||\hat{\mathbf{w}} - \mathbf{w}_0||_2/||\mathbf{w}_0||_2$, where $||\cdot||_2$ represents the Euclidean norm.

Table 4.2 summarizes the results obtained by each method in the experiments with non-uniform spike signals. The rows in this table display the average and the standard deviation of the signal reconstruction error, the logarithm of the model evidence and the time cost for each method. The best reconstruction performance is obtained by the LRMSSP with EP. The differences with respect to the other methods are statistically significant at the level $\alpha = 5\%$ according to a paired t test. The resulting p-values are all below $3 \cdot 10^{-5}$. The approximation of the model evidence is higher for the spike and slab than for the Laplace prior. Once more, RVM obtains the largest estimate of $\mathcal{P}(\mathbf{y}|\mathbf{X})$. The computational cost of the EP and RVM methods are similar. With the configuration selected, Gibbs sampling is much more costly than the other methods (up to 4000 times slower than EP).

The poor results of Gibbs sampling with respect to EP in this task have their origin in the propensity of the Markov chain to become trapped in sub-optimal modes of the posterior. This is illustrated by the plots in Figure 4.4, which show the signal estimates obtained by the different methods in a particular realization of the experiment. The LRMSSP with EP generates a signal reconstruction which is very accurate and cannot be visually distinguished from the original signal. By contrast, the Gibbs sampling approach generates many spikes of small magnitude that were not present in the original signal. The signal reconstruction given by RVM also presents similar problems. The reason for this is that the optimization process carried out by this method often converges to local and sub-optimal maxima of the type-II likelihood. This happens even when an additional greedy optimization process is used to reduce the impact of local and sub-optimal maxima, as in the implementation of RVM given by Ji et al. (2008). Finally, the Laplace model has the largest error in this problem, as illustrated the top-right plot in Figure 4.4. The Laplace prior produces excessive shrinkage of non-zero coefficients, while the magnitude of the coefficients that should be zero is not sufficiently reduced.

4.4.2.2 Uniform Spike Signals

The uniform spike signals are generated in a similar manner as the non-uniform ones. The only difference is that the non-zero elements of each signal vector are now sampled at random from the set $\{-1,1\}$. The experimental protocol is the same as before. The hyper-parameters of each method are initialized to the same values as in the experiments with non-uniform spike signals. However, we use in this case 100 measurements for the reconstruction of each signal vector because accurate reconstruction of uniform spike signals requires more data.

Table 4.3 presents the results of each method. By far, the most accurate reconstruction is provided by LRMSSP with EP for approximate inference. The differences with respect to the other methods are statistically significant at $\alpha = 5\%$ according to a paired t test. The p-values obtained are all lower than $2.2 \cdot 10^{-16}$. The evidence of the LRMSSP larger than the evidence of the Laplace model. The training times of both EP methods and the RVM approach are similar. Gibbs sampling has a much larger computational cost (85,000 times slower than EP) and obtains the worst performance. Figure 4.5 shows the signal estimates generated by the different methods in a particular realization of the experiment. The Gibbs sampling approach appears to be trapped in some sub-optimal mode of the posterior distribution. Similarly, RVM has converged to a local maximum of the type-II likelihood, which is sub-optimal. By contrast, the signal reconstruction given by the LRMSSP with EP is very accurate. These results show that EP seems to be less affected than Gibbs sampling by the multimodality of the posterior distribution under the spike and slab prior. This is a surprising result because EP is supposed to have problems when the posterior distribution is multimodal (Bishop, 2006). Finally, the behavior of the Laplace method is similar as in the non-uniform case. In this approach, the reduction of the magnitude of the different coefficients is uniform. Consequently, the shrinkage of the coefficients that should be zero is insufficient and the shrinkage of the coefficients that should be non-zero is too large.

In this case the performances of both Gibbs sampling and RVM are markedly worse than in the experiments with non-uniform spike signals. The reason for this is that, with uniform spike signals, it is more difficult to avoid sub-optimal maxima of the type-II likelihood or sub-optimal modes of the posterior distribution. In particular, the starting point of the Markov chain used

Table 4.3: Results for each method in the uniform spike signal reconstruction problem.

	SS-MCMC	Laplace-EP	RVM	SS-EP
Error	1.03 ± 0.61	0.84 ± 0.03	0.66 ± 0.54	0.01 ± 0.01
log $\mathcal{P}(\mathbf{y}\mid\mathbf{X})$		27.8 ± 5.3	248 ± 56	215 ± 5.9
Time	1783 ± 533	0.17 ± 0.02	0.12 ± 0.04	0.2 ± 0.03

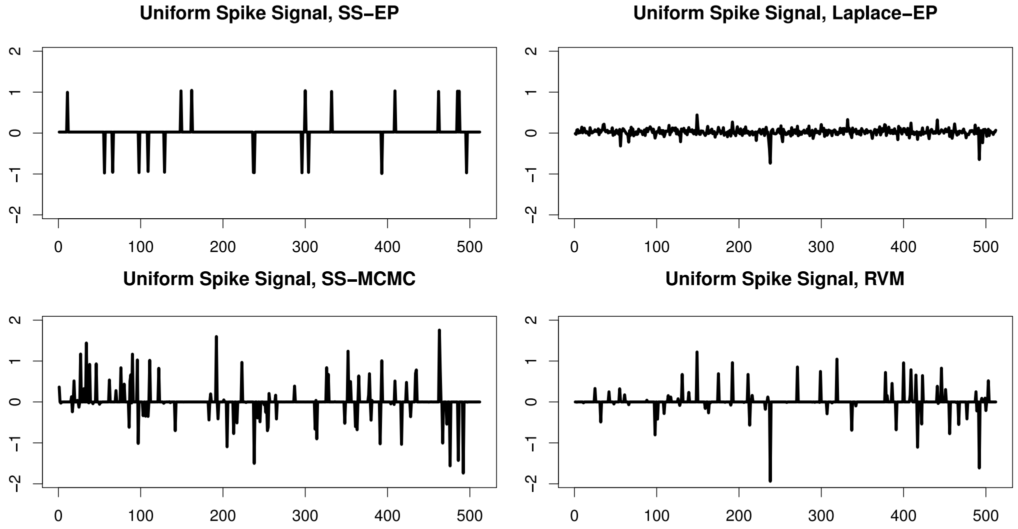

Figure 4.5: Signal estimates generated by each reconstruction method on a particular instance of the uniform spike signal problem. The original signal (not shown) cannot be visually distinguished from the approximation generated by EP in the spike and slab model.

by Gibbs sampling has to be a good initial solution. This solution is determined using a greedy procedure that is described in Appendix C.1. In RVM, the maximization of the type-II likelihood is also implemented using a similar greedy strategy (see the Matlab given by Ji et al. (2008)). When the signal consists of non-uniform spikes, these greedy strategies are very successful in identifying the signal elements which are truly different from zero. However, with uniform spike signals, these greedy processes make more mistakes. Consequently, in this latter case, RVM and SS-MCMC are more likely to get trapped into sub-optimal maxima of the type-II likelihood or sub-optimal modes of the posterior distribution, respectively.

4.4.3 User Sentiment Prediction

The LRMSSP has also applications in the field of natural language processing (Manning and Schütze, 2000). In particular, we consider the problem of sentiment prediction from user-written product reviews. The objective is to predict from the review text of a product, the rating assigned by the user to that product. The data analyzed in these experiments correspond to the sentiment dataset[1] described by Blitzer et al. (2007), which contains review texts and corresponding rating values taken from `www.amazon.com` within four different categories of products. The rating range in this dataset is from 1 to 5 stars. We specifically focus on the product categories *books* and *kitchen appliances* because these two categories generate the hardest and easiest prediction

[1] `http://www.seas.upenn.edu/~mdredze/datasets/sentiment/`

Table 4.4: Number of instances and features within each sentiment dataset.

Dataset	Instances	Features
Books	5501	1213
Kitchen	5149	824

Table 4.5: Results for each method in the books sentiment dataset.

	SS-MCMC	Laplace-EP	RVM	SS-EP
MSE	1.81 ± 0.04	1.84 ± 0.04	2.38 ± 0.16	1.81 ± 0.04
log $\mathcal{P}(\mathbf{y}\|\mathbf{X})$		-743 ± 11	-720 ± 4	-755 ± 17
Time	$155{,}438 \pm 84{,}057$	9.9 ± 0.9	2.1 ± 0.6	11.1 ± 3.4

problems, respectively. Each product review is represented using a vector of features whose components correspond to the unigrams and bigrams (Manning and Schütze, 2000) that appear in at least 100 reviews within the same product category. The feature values are given by the occurrences of these unigrams or bigrams in the text of the review. Table 4.4 shows the total number of instances and features in the resulting datasets.

The performance of the LRMSSP is evaluated in the problem of predicting the user rating from the vector of features which encodes the text of the product review. For this purpose, 20 random partitions of the data into non-overlapping training and test sets are made. The size of the training set is $n = 500$. This particular size is selected because we are interested in evaluating the results of the LRMSSP when the number of features is larger than the number of training instances (that is, $n < d$). During the training process, the data are normalized so that the instance features and the user ratings have zero mean and unit standard deviation on the training set. EP is used to fit a LRMSSP on each training set. The mean squared error (MSE) of the model is evaluated on the corresponding test set. On each of these train/test episodes, the hyper-parameters v_s and p_0 are determined by 10-fold cross-validation. The values considered are uniformly distributed in a 5×5 grid for $v_s \in [0.0025, 0.0125]$ and $p_0 \in [0.1, 0.3]$. The grids have been selected so that they are centered at the points that are selected more often by the cross-validation search, that is, 0.2 for p_0 and 0.0075 for v_s.

The Gibbs sampling method draws 10,000 samples from the posterior after a burn-in period with 1000 samples. In this case, the cross-validation grid search is not feasible because of its large computational cost. Therefore, the hyper-parameters v_s and p_0 take the same values as in the LRMSSP trained with EP. Hyper-parameter b_0 in the Laplace model is selected by running a 10-fold cross-validation grid search in the training set. The grid for b_0 is formed by 10 points uniformly distributed in the interval $[0.021, 0.03]$. The grid is centered at 0.025, the point which is selected more often by the cross-validation search. Finally, the hyper-parameter for the noise σ_0 is fixed to 1 in all the methods. This value provides good overall results for all the techniques.

Table 4.5 displays the results obtained by each approach in the book sentiment dataset. The rows in this table contain the average and the standard deviation of the MSE in the test sets, the logarithm of the model evidence and the training time in seconds for each method. For the two EP methods (SS-EP and Laplace-EP), we do not include in the training time field the time consumed by the cross-validation search. In this manner, we can compare directly the training times of the methods that use a cross-validation search (SS-EP and Laplace-EP) with the training times of the methods that do not perform this search (RVM and SS-MCMC). The best

Table 4.6: Results for each method in the kitchen sentiment dataset.

	SS-MCMC	**Laplace-EP**	**RVM**	**SS-EP**
MSE	1.59 ± 0.02	1.64 ± 0.03	1.91 ± 0.08	1.59 ± 0.02
log $\mathcal{P}(\mathbf{y}\|\mathbf{X})$		-712 ± 9.1	-713 ± 4	-718 ± 14
Time	$40{,}662 \pm 16{,}052$	7.6 ± 0.8	0.9 ± 0.2	9.5 ± 1.7

performing techniques in terms of test MSE are the LRMSSP with EP and Gibbs sampling. The differences in performance between these two methods are not statistically significant at $\alpha = 5\%$ according to a paired t test. The p-value obtained is 0.07. The differences between the LRMSSP with EP and the Laplace-EP and RVM methods are statistically significant according to a paired t test. The resulting p-values are all below $2 \cdot 10^{-8}$. However, the evidence of the Laplace model is slightly larger than the evidence of the LRMSSP, even though the former approach performs worse in test. This failure of the approximation of the model evidence to provide a reliable ranking of these two methods will be analyzed in the following subsection. Regarding training times, the methods that use EP for approximate inference (SS-EP and Laplace-EP) are more or less similar while RVM is slightly faster. The costliest method is the approach based on Gibbs sampling, which is on average 13,000 times slower than EP.

The results for the kitchen dataset are displayed in Table 4.6. Similarly as in the previous dataset, the methods with lowest predictive MSE are SS-EP and SS-MCMC. The differences between them are not statistically significant. A paired t test returns a p-value equal to 0.37. However, the differences between SS-EP and Laplace-EP and RVM are statistically significant according to a paired t test. The resulting p-values are smaller than $3 \cdot 10^{-10}$. In this case, the evidence of Laplace-EP is also larger than the evidence of SS-EP, even though Laplace-EP performs significantly worse in test. This result will be analyzed in the next subsection. Finally, the training cost of SS-MCMC is again larger than the cost of SS-EP (about 4000 times larger).

Figure 4.6 is useful for understanding the good results of the LRMSSP on the sentiment prediction problem. This figure shows the posterior means for $\mathbf{w}$ generated by each method on specific training instances of the sentiment datasets (left, books; right, kitchen). For the spike and slab model with EP (top plots), the posterior means of most of the model coefficients are shrunk towards zero while a few of these coefficients have posterior means that are significantly different from zero. The effect is stronger for the kitchen dataset. When a Laplace prior is used, this selective shrinkage process is less effective (middle plots). The reduction of the magnitude of the coefficients that are close to zero is not enough. By contrast, the Laplace prior causes a reduction of the magnitude of non-zero coefficients that is too large. The posterior means produced by the RVM method (bottom plots) include too many components that are significantly different from zero, which is a clear mark of overfitting.

4.4.3.1 Accuracy of the EP Approximation of the Model Evidence

In both sentiment prediction tasks (books and kitchen), SS-EP has lower generalization error than Laplace-EP. However, the model evidence is larger for the latter model. Hence, using the model evidence to select the more accurate method can be misleading in these problems. The origin of these results might be the failure of EP to generate accurate approximations of the model evidence. To further investigate this issue we carry out a series of experiments with

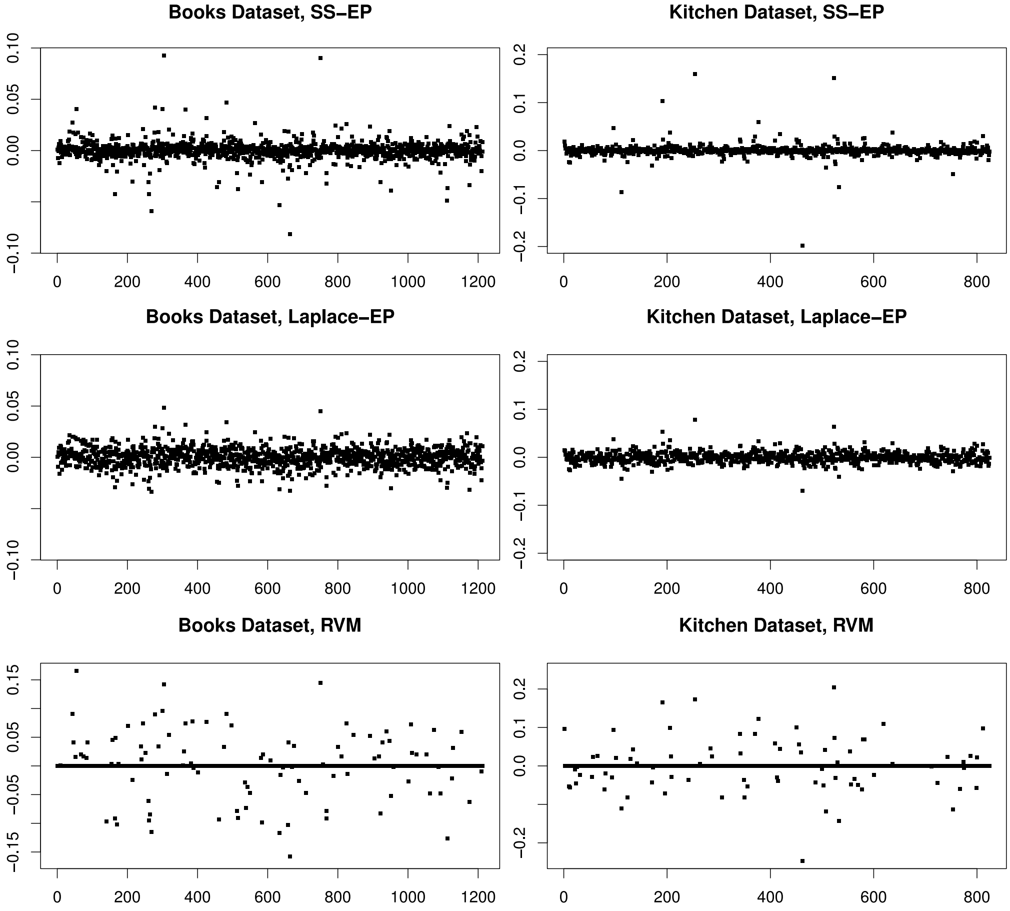

Figure 4.6: Posterior mean for **w** generated by the spike and slab model with EP (top), the Laplace model (middle) and RVM (bottom) in a particular training instance of the books (left) and kitchen (right) datasets. The posterior means generated by the Gibbs sampling approach (not shown) cannot be visually distinguished from the ones generated by the EP method.

simulated data. In these experiments, the approximations of the evidence generated by SS-EP and Laplace-EP are compared with the exact values of these normalization constants obtained by numerical methods.

For each prior distribution (spike and slab and Laplace), we generate 100 design matrices with $n = 4$ and $d = 4$, where the components of each design matrix follow independent standard Gaussian distributions. The hyper-parameters of the priors are fixed to the most frequent values obtained in the cross-validation searches described above: $p_0 = 0.2$, $v_s = 0.0075$ and $b_0 = 0.025$. As in the experiments with sentiment data, the standard deviation of the noise in the targets is fixed to 1. For each design matrix, we sample a 4-dimensional vector of coefficients from the corresponding prior distribution. Using this vector and the associated design matrix, we draw a vector of target values according to the linear model given by (4.1). Following this process, we finally obtain 100 pairs of design matrices and associated target vectors for each of the two prior distributions. Here, we have selected d to be relatively small because, otherwise, numerical methods would not be feasible, especially for the Laplace model.

Table 4.7: Squared error in the log-evidence for each method in the simulated data.

Method	**Error in log $\mathcal{P}(\mathbf{y}\|\mathbf{X})$**
SS-EP	$8.2 \cdot 10^{-12}$
Laplace-EP	$1.1 \cdot 10^{-09}$

Table 4.7 shows the average squared error in the approximation of the logarithm of the evidence for SS-EP (top row) and Laplace-EP (bottom row). Both methods perform very well, generating very accurate approximations of the evidence. The reported error for Laplace-EP is larger than the error of SS-EP because the numerical method used for calculating the evidence in the Laplace model achieves less precision. This method requires to compute a multi-dimensional integral on $\mathbf{w}$ using a numerical grid quadrature approach. By contrast, in the spike and slab model, the integrals can be computed analytically when we condition on $\mathbf{z}$. Consequently, in this case, the numerical method only needs to sum over all the possible configurations for $\mathbf{z}$, which is always more accurate than using numerical quadrature techniques. This experiment is repeated with $\{n = 8, d = 8\}$ and $\{n = 16, d = 16\}$ only for the spike and slab model because the numerical method for the Laplace prior is not feasible for these high-dimensional data. In these cases, the average squared errors in the approximations of the log-evidence generated by EP are $3.7 \cdot 10^{-11}$ and $4 \cdot 10^{-10}$, respectively. These results suggest that the squared error in the EP estimate of $\log \mathcal{P}(\mathbf{y}|\mathbf{X})$ is multiplied by a factor of ≈ 10 when n and d are doubled.

The conclusion of these experiments is that the discrepancies between the predictive error and the model evidence in the sentiment datasets are unlikely to originate by a lack of precision of the EP estimates for $\log \mathcal{P}(\mathbf{y}|\mathbf{X})$. An alternative explanation is that the analyzed models impose some assumptions on the structure of the data (for instance, the assumption linearity of the targets with respect to the design matrix) which does not hold in practice. If such were the case, the model evidence can be an unreliable tool for model selection (Bishop, 2006).

4.5 Summary and Discussion

Many regression problems of practical interest have a feature space whose dimension, d, is significantly larger than n, the number of available data instances. Under these conditions, the learning process is often implemented assuming a sparse linear model to reduce overfitting (Johnstone and Titterington, 2009). In a Bayesian framework, sparsity can be favored by using specific priors, such as the Laplace (Seeger, 2008), the degenerate Student's t (Tipping, 2001) and the spike and slab (George and McCulloch, 1997) priors. These priors induce a bi-separation in the posterior distribution between a few coefficients whose probability of being different from zero is large and many coefficients that have very small posterior means. Ishwaran and Rao (2005) call this bi-separation effect *selective shrinkage*.

Spike and slab priors are more suited to enforce sparsity than Laplace and Student's t priors because they consider two different classes of coefficients: The spike is the prior distribution of the coefficients that are zero in the true model. The slab is the prior distribution of the coefficients that are actually different from zero. The Laplace prior is less flexible (it has a single scale parameter) and does not allow to discriminate between groups of coefficients. As a result, the Laplace prior produces a more uniform reduction of the magnitude of the coefficients and it is less effective than the spike and slab distribution for enforcing sparsity. The use of a type-II maximum-likelihood approach in the model that assumes the degenerate Student's t prior often results in significant overfitting problems, specially when the dimensionality of the feature space is very large.

A disadvantage of spike and slab priors is that they make approximate inference a difficult and computationally demanding problem. Inference in the linear regression model with a spike and slab prior is commonly implemented using Markov Chain Monte Carlo (MCMC) methods such as Gibbs sampling (George and McCulloch, 1997). However, the high computational cost of Gibbs sampling makes this method infeasible when d is very large. As a more efficient alternative, we have proposed to use the expectation propagation (EP) algorithm (Minka, 2001). The cost of EP is $O(n^2d)$ when the number of training instances n is smaller than d. EP has been evaluated in regression problems with $n < d$ from fields of application of practical interest: the reverse engineering of transcription networks, the reconstruction of sparse signals given a reduced number of linear measurements and the prediction of sentiment from user-written product reviews. In these tasks, EP outperforms or obtains comparable results to Gibbs sampling at a much lower computational cost. Another advantage of EP with respect to Gibbs sampling is that the performance of EP seems to be less affected by the multi-modality of the posterior distribution.

In the analyzed problems, EP outperforms other sparse Bayesian linear regression models that assume Laplace and degenerate Student's t priors. The good overall results of the model based on the spike and slab prior are explained by the superior selective shrinkage capacity of this specific prior distribution: In the posterior distribution computed by EP, most of the model coefficients have a large probability of being close to zero. Only for a small subset of coefficients is the posterior probability centered around values that are significantly different from zero. By contrast, the Laplace prior produces a more uniform reduction of the magnitude of the model coefficients. As a result, the reduction of the magnitude of the coefficients that should be zero is insufficient while, at the same time, the reduction of the size of the coefficients that should be different from zero is too large. The accuracy of the method that assumes the degenerate Student's t prior is rather poor in all the analyzed problems. The reason for this poor performance is that the resulting model includes an excessive number of coefficients which are significantly different from zero: A clear symptom of overfitting. This result has already been noted by Qi et al. (2004).

The superior performance of a model that assumes a particular prior distribution over a different prior ultimately depends on the actual distribution of the data. In particular, when the true prior of the coefficients is a Laplace distribution, the model that assumes a prior of this form should perform better than any other approach. For the same reason, when, in the actual model, many coefficients are exactly zero, spike and slab priors should be preferred because they can assign non-zero probability to such solutions (the spike). Other priors that favor sparsity (such as the Laplace prior) do not have this characteristic.

A drawback of EP is that it is not guaranteed to converge. In our implementation of this algorithm, different methods have been used to improve the convergence of EP, such as the restriction of the components of $\tilde{\mathbf{v}}_2$ to be positive and the use of an annealing process for the damping parameter ε. In all the experiments described above, SS-EP generally stops after less than 20 iterations. However, in some specific cases, SS-EP may take more than 200 iterations to stop, especially when σ_0 and p_0 are very small and the amount of training data is very reduced. This occurs more often in the signal reconstruction problems, when the number of available measurements is so small that no accurate reconstruction of the signal is possible. By contrast, the Laplace-EP method exhibits better convergence properties and does not seem to be affected by this shortcoming.

Chapter 5

Network-based Sparse Bayesian Classification

In some classification problems there is prior information about the joint relevance of groups of features. This knowledge can be encoded in a network whose nodes correspond to features and whose edges connect features that should be either both excluded or both included in the final predictive model. In this chapter, we introduce a novel network-based sparse Bayesian classifier (NBSBC) that makes use of the information about feature dependencies encoded in such network to improve its prediction accuracy, especially in problems with a high-dimensional feature space and a limited amount of training data. Approximate Bayesian inference is efficiently implemented in this model using expectation propagation. The NBSBC method is validated on four real-world classification problems from different domains of application: phonemes, handwritten digits, precipitation records and gene expression measurements. A comparison with state-of-the-art methods (support vector machine, network-based support vector machine and graph lasso) show that NBSBC has excellent predictive performance, obtaining the best accuracy in three of the four analyzed problems and ranking second in the modeling of the precipitation data. NBSBC also yields accurate and robust rankings of the individual features according to their relevance to the solution of the classification problem considered. The accuracy and stability of these estimates is an important factor in the good overall performance of this method.

5.1 Introduction

In some supervised learning problems it can be difficult to build robust and reliable classifiers when n, the number of instances available for induction, is small and d, the dimension of the vector of features that characterizes each instance, is large. This ”large d, small n” paradigm arises in microarray studies (Dudoit and Fridlyand, 2003), image analysis (Seeger et al., 2010), astronomy (Johnstone and Titterington, 2009) or fMRI data modeling (Pereira et al., 2009), to name a few important applications. A common approach to learning in these settings is to assume an underlying linear model, possibly in an expanded feature space. More robust predictors can often be obtained if we assume that only a subset of the original features are actually necessary for classification (Johnstone and Titterington, 2009). In this case, the linear model is assumed to be sparse. A common procedure to enforce sparsity is to include in the objective function a penalty term proportional to the L_1 norm of the vector of model coefficients. Some methods that use this type of regularization penalty are the lasso (Tibshirani, 1996), the 1-norm support vector machine (Zhu et al., 2004) and the elastic net (Zou and Hastie, 2005). Within a Bayesian framework, sparsity can be favored by considering a sparsifying prior for the model coefficients. Some examples are the Laplace distribution (Seeger, 2008) or the ”spike and slab” distribution (George and McCulloch, 1997). Unfortunately, exact Bayesian inference is generally not feasible when these priors are used. Approximate inference methods, such as MCMC sampling (George and McCulloch, 1997), variational inference (Nickisch and Seeger, 2009) or expectation propagation (Seeger, 2008) can be used in these cases. Sparse models often have an improved prediction accuracy, and can also be used to identify the features that are more relevant for solving the classification problem.

In most sparse classification models analyzed in the literature, the features that describe the instances to be classified are assumed to be independent. However, in some problems, there is information about the dependencies that exist between specific features. For example, daily rainfall measurements collected at nearby meteorological stations are frequently correlated. In real-world images, adjacent pixels tend to have similar colors and intensities. In the spectral decomposition of phonemes, contiguous frequencies are often dependent. Even when domain-specific information is not available, feature dependencies can also be inferred from unlabeled data. In this chapter, we propose to improve a sparse Bayesian classifier by directly incorporating this prior information into the model. For this purpose, knowledge about the dependencies among features is encoded with the help of a network. Each node in the network corresponds to a different feature. A link between two nodes in the network indicates that the corresponding features should be either both included or both excluded from the model used for prediction. Similar ideas have been investigated in the context of non-sparse (Sandler et al., 2008) and sparse models (Li and Li, 2008; Slawski et al., 2009). However, in these articles, the assumptions on the nature of the dependencies among features are different from ours. Specifically, (Li and Li, 2008; Sandler et al., 2008; Slawski et al., 2009) assume that two coefficients take similar values whenever the corresponding features are linked in the network. By contrast, in the current investigation, we assume that these coefficients are either both zero or both different from zero, but not necessarily of the same magnitude or even of the same sign. Hence, we consider the possibility that the corresponding features can be negatively as well as positively correlated.

More recently, Zhu et al (Zhu et al., 2009) have introduced a sparse network-based support vector machine (NBSVM) that uses a network to incorporate the effect of feature dependencies into the learning algorithm, in the same spirit as the approach described in this chapter. The improvements in performance that can be achieved as a result of the enhancement provided by the network are illustrated in synthetic and microarray data. A drawback of the NBSVM method is that its training cost is rather high. Another sparse method that also incorporates information on feature dependencies by means of a network is the graph lasso (GL) (Jacob et al., 2009). This technique builds a sparse logistic regression model. The model objective function includes a regularization term that favors the selection of features that are linked in the network. The GL is very efficient in terms of computational cost. Finally, another very recent sparse Bayesian linear method based on the Laplace prior that includes feature dependencies, but which does not make explicit use of a network of features, is described by van Gerven et al. (2010).

In this chapter, we propose a Bayesian alternative to the NBSVM and GL methods (Hernández-Lobato et al., 2010b). Specifically, we introduce a network-based sparse Bayesian classifier (NBSBC) based on an extension of the Bayes point machine (Herbrich et al., 2001; Minka, 2001) that is capable of learning the level of noise in the class labels directly from the data (Hernández-Lobato and Hernández-Lobato, 2008). This model considers a spike and slab sparsifying prior (George and McCulloch, 1997) combined with a Markov random field prior (Bishop, 2006; Wei and Li, 2007) that accounts for the network of feature dependencies. Approximate Bayesian inference in NBSBC is implemented using the expectation propagation (EP) algorithm (Bishop, 2006; Minka, 2001). The performance of NBSBC is evaluated in four classification problems including phoneme (Hastie et al., 1995, 2001), handwritten digit (Lecun et al., 1998), precipitation (Razuvaev et al., 2008) and gene expression (Bos et al., 2009) data. In these tasks, NBSBC is compared with the sparse Bayesian classifier (SBC) that results when NBSBC ignores the network of feature dependencies, the standard support vector machine (SVM), the network-based support vector machine (NBSVM), and the graph lasso (GL). These experiments show that NBSBC has an excellent overall predictive performance. It is the most accurate predictor in three of the four classification problems and ranks second in the modeling of the precipitation data. Additionally, the ranking of the individual features according to their relevance for solving the classification problem given by NBSBC is more accurate and robust than the estimates produced by NBSVM, GL and SBC. Finally, when the network of feature dependencies is sparse, that is, when most features are only connected to at most k neighbors and $k \ll d$, the computational complexity of the novel Bayesian classifier is $O(nd)$. This means that NBSBC can be trained much faster than NBSVM, and has a cost similar to GL.

The chapter is organized as follows: The methods NBSVM and GL are described in Section 5.2. These will be used as benchmarks for comparison with the novel network-based sparse Bayesian classifier, which is described in Section 5.3. The application of EP to the NBSBC model is described in Section 5.4. Section 5.5 reports the results of an extensive empirical assessment of NBSBC. The performance of NBSBC is compared to SVM, NBSVM, SBC and GL on four classification problems from different domains of application: *phonemes*, *handwritten digits*, *precipitation records* and *data from microarray measurements*. In this section, we also analyze the stability and robustness of the different sparse linear models under small perturbations of the training data generated via sub-sampling. Finally, the chapter closes with a summary and a discussion in Section 5.6.

5.2 Previous Work

We review two methods for the construction of sparse linear classifiers that employ a network of feature dependencies and will be used as benchmarks for comparison: the network-based support vector machine (NBSVM) (Zhu et al., 2009) and the graph lasso method (GL) (Jacob et al., 2009). Both methods build linear models whose coefficients are determined by minimizing a penalized loss function. The penalty term is included to enforce sparsity in the vector of model parameters. This penalty also takes into account the network of feature dependencies. Features that are linked by edges in the network tend to be either both excluded or both included in the learned model.

5.2.1 Network-based Support Vector Machine

Zhu et al. (2009) present an extension of the standard support vector machine (SVM) (Vapnik, 1995) that takes into account a network encoding feature dependencies. In this model, a sparsity enforcing penalty is added to the *hinge loss* function of the standard SVM so that features that are linked in the network tend to be either both excluded or both included in the model. Given a training dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with features $\mathbf{x}_i \in \mathbb{R}^{d+1}$ and corresponding class labels $y_i \in \{-1, 1\}$, the network-based SVM (NBSVM) searches for the parameter vector $\mathbf{w} = (w_0, \ldots, w_d)^\mathrm{T}$ that minimizes

$$\sum_{i=1}^{n} \left[1 - y_i \mathbf{w}^\mathrm{T} \mathbf{x}_i\right]_+ + \lambda \sum_{\{i,j\} \in E} \max\left\{|w_i|, |w_j|\right\} \tag{5.1}$$

where E is the set of edges in the network of feature dependencies and λ is a positive regularization parameter. The zeroth component of each $\mathbf{x}_i$ is assumed to be constant and equal to 1 so that w_0 is the bias coefficient for the model. Typically, w_0 is not regularized in the NBSVM. For this reason, the zeroth feature is not linked to any other feature in the network of feature dependencies. The absolute value functions in the penalty term favor sparsity in the classification model. Additionally, if a specific feature is excluded from the model, then the penalty term in (5.1) favors the exclusion of features that share an edge with it. This is a consequence of the singular nature of the $\max\{|\cdot|, |\cdot|\}$ function at the origin (Zou and Yuan, 2008). The minimization of (5.1) is a linear programming (LP) problem (Zhu et al., 2009) and can therefore be efficiently performed using standard LP solvers.

5.2.2 Graph Lasso

The graph lasso (GL) was introduced by Jacob et al. (2009) as a regularization method that allows to obtain a sparse linear model in which the selected features tend to be connected to each other in a graph. Before describing the penalty term used by GL, it is useful to introduce some notation. Let $\mathbf{w} = (w_0, \ldots, w_d)^\mathrm{T}$ be the parameter vector of a linear model and let $G = (V, E)$ be a network whose vertices $V = \{0, \ldots, d\}$ correspond to features. E is the set of edges that connect features. The elements of E are sets $\{i, j\}$ such that $i, j \in V$. Let D be the set of vertices (features) that are not linked to any other vertices in G. For any vector $\mathbf{v} = (v_0, \ldots, v_d)^\mathrm{T}$, the quantity $\|\mathbf{v}\|$ represents the Euclidean norm of $\mathbf{v}$. Let $\mathrm{supp}(\mathbf{v}) \subset V$ denote the support of $\mathbf{v}$; namely, the set of features $i \in V$ such that $v_i \neq 0$. Given $\mathbf{v}$ and an edge $e \in E$, $\mathbf{v}^e$ is the 2-dimensional vector $(v_i, v_j)^\mathrm{T}$ where i and j are the two features linked by e, and $i \leq j$. Similarly, $\mathbf{v}^D$ is the $|D|$-dimensional vector given by the components of $\mathbf{v}$ that belong to D.

To construct the penalty function in GL, we consider a decomposition of the vector $\mathbf{w}$ as a sum of $|E| + 1$ vectors:

$$\mathbf{w} = \mathbf{u} + \sum_{i=1}^{|E|} \mathbf{v}_i, \tag{5.2}$$

where $\mathbf{u}$ is a vector whose only non-zero components correspond to the disconnected part of the graph ($\mathrm{supp}(\mathbf{u}) \subset D$), and $\mathbf{v}_i$ is a vector whose only non-zero components are the elements corresponding to the vertices linked by e_i, the i-th edge in E ($\text{supp}(\mathbf{v}_i) \subset e_i$). This decomposition is not unique. Let $\mathcal{V}_{\mathbf{w}}$ be the set of $(|E|+1)$-tuples $(\mathbf{v}_1,\ldots,\mathbf{v}_{|E|},\mathbf{u})$, which correspond to all possible decompositions of $\mathbf{w}$ of this type. The GL regularization term is

$$\Omega^E_{\text{graph}}(\mathbf{w}) = \min_{(\mathbf{v}_1,\ldots,\mathbf{v}_{|E|},\mathbf{u})\in\mathcal{V}_{\mathbf{w}}} \sum_{i=1}^{|E|} \|\mathbf{v}_i^{e_i}\|\,, \tag{5.3}$$

which is written in terms of the decomposition of $\mathbf{w}$ that minimizes the sum of the Euclidean norm of the vectors that correspond to edges. The Euclidean norm in (5.3) enforces sparsity at the edge level in $\mathbf{w}$. Specifically, if one of the components of $\mathbf{v}_i^{e_i}$ is zero, the value of Euclidean norm is the absolute value of the other component. This is akin to a lasso penalty, which favors that this second component also becomes zero (Yuan and Lin, 2006). This form of regularization privileges weight vectors $\mathbf{w}$ whose support is the union of D, the disconnected part of the graph, and a subset of the edges in E. In contrast with the sparsity patterns generated by other network-based methods, which tend to select connected components in the network, the edges included in the model by GL are not necessarily connected to each other.

This penalty function is combined with the negative log-likelihood of a logistic regression model to obtain a network-based sparse classifier. Given a training dataset $\{(\mathbf{x}_i,y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^{d+1}$ is the feature vector and $y_i \in \{-1,1\}$ is the class label of the i-th example, the GL method searches for the $\mathbf{w}$ that minimizes $\sum_{i=1}^n \ell(y_i,\mathbf{x}_i,\mathbf{w}) \ +\lambda\Omega^E_{\text{graph}}(\mathbf{w})$, where $\lambda > 0$ is a regularization parameter,

$$\ell(y_i,\mathbf{x}_i,\mathbf{w}) = \frac{y_i-1}{2}\log(1-\sigma(\mathbf{w}^{\text{T}}\mathbf{x}_i)) - \frac{y_i+1}{2}\log\sigma(\mathbf{w}^{\text{T}}\mathbf{x}_i) \tag{5.4}$$

and $\sigma(\cdot)$ is the logistic function. The zeroth component of each $\mathbf{x}_i$ is constant and equal to 1 so that w_0 is the bias coefficient for the model. To avoid regularizing w_0 the zeroth component of each $\mathbf{x}_i$ is not connected to any other feature in the network G. The optimization problem can be readily solved by duplicating the features in the dataset that are involved in edges of G. Specifically, the original feature vector for the i-th instance $\mathbf{x}_i$ is replaced by the enlarged vector $\tilde{\mathbf{x}}_i$ obtained by concatenating copies of the features, one copy per edge $\tilde{\mathbf{x}}_i = (\mathbf{x}_i^{e_1},\ldots,\mathbf{x}_i^{e_{|E|}},\mathbf{x}_i^D)^{\text{T}}$. Using these expanded feature vectors, the optimization problem becomes

$$\min_{\tilde{\mathbf{w}}} \sum_{i=1}^n \ell(y_i,\tilde{\mathbf{x}}_i,\tilde{\mathbf{w}}) \quad \text{subject to} \quad \sum_{i=1}^{|E|} \|(\tilde{w}_{2i-1},\tilde{w}_{2i})\| \leq M \tag{5.5}$$

where $\tilde{\mathbf{w}} = (\tilde{w}_1,\ldots,\tilde{w}_{2|E|+|D|})^{\text{T}}$ and M is a positive regularization parameter that is in a one-to-one relation with λ. Once a solution to this expanded problem has been found, a minimizer for the original problem can be computed by realizing that at both optima $\mathbf{u}^D = (\tilde{w}_{2|E|+1},\ldots,\tilde{w}_{2|E|+|D|})^{\text{T}}$, $\mathbf{v}_i^{e_i} = (\tilde{w}_{2i-1},\tilde{w}_{2i})^{\text{T}}$ and $\mathbf{w}$ is equal to the sum of all the $\mathbf{v}_i$ and $\mathbf{u}$. The constrained optimization in (5.5) is a Group-Lasso regularization problem (Kim et al., 2006; Yuan and Lin, 2006) which can be efficiently solved using the method described by Roth and Fischer (2008).

5.3 Network-based Sparse Bayesian Classification

In this section we present a novel network-based sparse Bayesian classifier that effectively makes use of information on feature dependencies to improve the prediction accuracy of the model and the capacity to identify features that are relevant for classification. Consider a supervised learning task in which $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is a set of training instances with features $\mathbf{x}_i \in \mathbb{R}^{d+1}$ and class labels $y_i \in \{-1, 1\}$. The zeroth component of every $\mathbf{x}_i$ is constant and equal to 1. The objective is to build the linear classifier $\mathbf{w} = (w_0, \ldots, w_d)^\mathrm{T}$ that optimally separates instances of different classes. Following Herbrich et al. (2001), we assume the existence of a ”true” parameter vector $\mathbf{w}_{true} \in \mathbb{R}^{d+1}$ that has been used to label the data according to the rule $y_i = \mathrm{sign}\left(\mathbf{w}_{true}^\mathrm{T}\mathbf{x}_i\right)$. Since, in a general case, the data need not be linearly separable, we consider the possibility that some of the class labels y_i have been flipped with probability ε. Given these assumptions, the likelihood for $\mathbf{w}$ given $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\mathrm{T}$, $\mathbf{y} = (y_1, \ldots, y_n)^\mathrm{T}$ and ε is

$$\mathcal{P}(\mathbf{y}|\mathbf{w}, \varepsilon, \mathbf{X}) = \prod_{i=1}^n \mathcal{P}(y_i|\mathbf{w}, \varepsilon, \mathbf{x}_i) = \prod_{i=1}^n \left[\varepsilon\left(1 - \Theta\left(y_i \mathbf{w}^\mathrm{T}\mathbf{x}_i\right)\right) + (1-\varepsilon)\Theta\left(y_i \mathbf{w}^\mathrm{T}\mathbf{x}_i\right)\right], \tag{5.6}$$

where Θ is the Heaviside step function. Note that (5.6) is robust to outliers because it only depends on the number of errors of $\mathbf{w}$ in the training set and not on the actual size of these errors. The problems that we are interested in are characterized by a high-dimensional feature space and a small amount of training instances ($d \gg n$). This is an under-determined scenario in which many different parameter vectors fit the data equally well. To break this symmetry, we introduce a prior distribution for $\mathbf{w}$ that captures our expectation that some particular values of the parameters are more likely than others. Specifically, we assume that only a small subset of the components of $\mathbf{x}_i$ are actually relevant for predicting the class label y_i. Thus, $\mathbf{w}_{true}$ is assumed to be a sparse vector with only a few non-zero components. To incorporate this expectation, we follow George and McCulloch (1997) and introduce a vector of binary latent variables $\mathbf{z} = (z_0, \ldots, z_d)^\mathrm{T} \in \{-1, 1\}^d$, where $z_i = 1$ if the i-th component of $\mathbf{w}_{true}$ is different from zero and $z_i = -1$ otherwise. Assuming that $\mathbf{z}$ is known, the spike and slab prior density for $\mathbf{w}$ is

$$\mathcal{P}(\mathbf{w}|\mathbf{z}) = \prod_{i=0}^d \left[\frac{z_i+1}{2}\mathcal{N}(w_i|0, \sigma_i^2) + \frac{1-z_i}{2}\delta(w_i)\right], \tag{5.7}$$

where $\mathcal{N}(x|\mu, \sigma^2)$ is a Gaussian density with mean μ and variance σ^2 (the slab), $\delta(w_i)$ is a Dirac’s delta function (the spike), which corresponds to a point probability mass for w_i at zero, $\sigma_1^2, \ldots, \sigma_d^2$ are equal to 1 and σ_0^2 is equal to 100. σ_0^2 is much larger than $\sigma_1^2, \ldots, \sigma_d^2$ to guarantee that the prior for the bias parameter w_0 is not informative. To complete the specification of the prior for $\mathbf{w}$ we assume that a network that encodes the dependencies between features is known. This network is an undirected graph $G = (V, E)$ whose vertices $V = \{0, \ldots, d\}$ correspond to instance features and whose edges, E, link features that are expected to be both excluded or both included in the classification model. Given G, the prior probability for $\mathbf{z}$ is represented by a Markov random field (MRF) model (Bishop, 2006; Wei and Li, 2007). This is the main reason for choosing the latent variables z_i to take values in $\{-1, 1\}$, which is the standard notation used in Markov random fields. The corresponding prior probability for **z** is

$$\mathcal{P}(\mathbf{z}|G,\alpha,\beta) = \frac{1}{Z}\exp\left\{10z_0 + \alpha\sum_{i=1}^{d} z_i\right\}\exp\left\{\beta\sum_{\{j,k\}\in E} z_j z_k\right\}, \tag{5.8}$$

where Z is a normalization constant, $\alpha \in \mathbb{R}$ controls the level of sparsity in the model, $\beta \geq 0$ determines the correlation between z_i and z_j when features i and j are linked in G and the constant 10 reflects our expectation that $z_0 = 1$ is much more likely than $z_0 = -1$ so that the prior for w_0 does not favor solutions for which this bias coefficient is zero. Following the prescription given by Hernández-Lobato and Hernández-Lobato (2008), the prior for ε is

$$\mathcal{P}(\varepsilon) = \text{Beta}(\varepsilon|a_0,b_0) = \frac{1}{B(a_0,b_0)}\varepsilon^{a_0-1}(1-\varepsilon)^{b_0-1}, \tag{5.9}$$

where B is the beta function with parameters a_0 and b_0. The results obtained are not very sensitive to the values of these hyper-parameters, provided that they are consistent with the assumption that most of the training data are correctly labeled. Specifically, the choice made in the experiments presented in Section 5.5, $a_0 = 1$ and $b_0 = 9$, is equivalent to assuming that one out of ten data instances are mislabeled. This prior expresses a moderate level of confidence in that the labels of the training data are correct.

Once the specification of the network-based sparse Bayesian classifier (NBSBC) is made, Bayes' theorem can be used to compute the posterior distribution of the model parameters **w** and ε given the training data **X** and **y**. Assuming that the network G and the model hyperparameters α and β are known, the posterior is given by

$$\mathcal{P}(\mathbf{w},\varepsilon|\mathbf{y},\mathbf{X},G,\alpha,\beta) = \frac{\sum_{\mathbf{z}}\mathcal{P}(\mathbf{y}|\mathbf{w},\varepsilon,\mathbf{X})\,\mathcal{P}(\mathbf{w}|\mathbf{z})\mathcal{P}(\mathbf{z}|G,\alpha,\beta)\mathcal{P}(\varepsilon)}{\mathcal{P}(\mathbf{y}|\mathbf{X},G,\alpha,\beta)}. \tag{5.10}$$

The denominator in (5.10) is a normalization constant that is known as the *model evidence*. This constant can be used for model selection (Bishop, 2006; MacKay, 1992). Given an unlabeled test instance $\mathbf{x}^{\text{test}}$, the predictive distribution for the corresponding class label y^{test} is

$$\mathcal{P}(y^{\text{test}}|\mathbf{x}^{\text{test}},\mathbf{y},\mathbf{X},G,\alpha,\beta) = \int\int \mathcal{P}(y^{\text{test}}|\mathbf{w},\varepsilon,\mathbf{x}^{\text{test}})\mathcal{P}(\mathbf{w},\varepsilon|\mathbf{y},\mathbf{X},G,\alpha,\beta)\,d\mathbf{w}\,d\varepsilon. \tag{5.11}$$

An advantage of this approach is that the relevance of the features can be quantified by the posterior of **z**

$$\mathcal{P}(\mathbf{z}|\mathbf{y},\mathbf{X},G,\alpha,\beta) = \frac{\int\int\mathcal{P}(\mathbf{y}|\mathbf{w},\varepsilon,\mathbf{X})\,\mathcal{P}(\mathbf{w}|\mathbf{z})\mathcal{P}(\mathbf{z}|G,\alpha,\beta)\mathcal{P}(\varepsilon)\,d\mathbf{w}\,d\varepsilon}{\mathcal{P}(\mathbf{y}|\mathbf{X},G,\alpha,\beta)}. \tag{5.12}$$

Specifically, the relevance of the i-th feature is a number between 0 and 1 given by the marginal probability of the event $z_i = 1$ using the posterior (5.12). Finally, this Bayesian framework also allows to compute an estimate of the level of noise in the class labels as the average of ε over its posterior distribution

$$\bar{\varepsilon} = \int\int \varepsilon\,\mathcal{P}(\mathbf{w},\varepsilon|\mathbf{y},\mathbf{X},G,\alpha,\beta)\,d\mathbf{w}\,d\varepsilon. \tag{5.13}$$

This quantity also provides an estimate of the generalization error of NBSBC. Unfortunately, the sums and integrals in (5.10), (5.11), (5.12) and (5.13) are in most cases too costly to be practicable. For this reason, to implement the NBSBC model, it is necessary to resort to approximate methods for Bayesian inference. In this chapter, an approximation of the joint distribution $\mathcal{P}(\mathbf{w},\varepsilon,\mathbf{z},\mathbf{y}|\mathbf{X},G,\alpha,\beta)$ is given in terms of a simpler unnormalized distribution $\mathcal{Q}(\mathbf{w},\varepsilon,\mathbf{z})$ that belongs to the exponential family. The computation of $\mathcal{Q}$ is made using the expectation propagation (EP) algorithm (Bishop, 2006; Minka, 2001). Once $\mathcal{Q}$ is known, the previous sums and integrals can be computed in a straightforward manner.

5.4 Expectation Propagation for NBSBC

A description of the EP algorithm in its general form is given in Section 4.3. Here, we only describe the specific implementation of EP for the NBSBC approach. To apply the EP algorithm to the NBSBC model, the posterior distribution for the model parameters and the latent variables is approximated by a factorized distribution that belongs to the exponential family

$$\mathcal{P}(\mathbf{w},\varepsilon,\mathbf{z}|\mathbf{y},\mathbf{X},G,\alpha,\beta) \approx \text{Beta}(\varepsilon|a,b)\prod_{i=0}^{d}\mathcal{N}(w_i|m_i,v_i)\text{Bern}(z_i|p_i) = \mathcal{Q}(\mathbf{w},\varepsilon,\mathbf{z})\,, \tag{5.14}$$

where $\text{Bern}(z|p)$ is a Bernoulli distribution on $z\in\{-1,1\}$ such that p is the probability of the event $z=1$ and $\mathbf{m}=(m_0,\ldots,m_d)^{\text{T}}$, $\mathbf{v}=(v_0,\ldots,v_d)^{\text{T}}$, $\mathbf{p}=(p_0,\ldots,p_d)^{\text{T}}$, a and b are free distributional parameters. In this case we are not using the logistic function to parameterize the Bernoulli terms in the posterior approximation, as we did in Chapter 4. The reason for this is that in classification problems, the posterior probability of $z_i=1$ is seldom very high or very low for $i=1,\ldots,d$ and consequently, numerical stability of the EP method is better in this case than in the regression setting. Additionally, we do not use any damping scheme here because EP also has better convergence properties in the classification case.

The proposed approximation assumes that the individual components of $\mathbf{w}$ and $\mathbf{z}$ are independent. This simplification results in an EP algorithm with a reduced computational complexity. $\mathcal{P}(\mathbf{w},\varepsilon,\mathbf{z},\mathbf{y}|\mathbf{X},G,\alpha,\beta)$ can be calculated as the product of $n+|E|+3$ terms. These terms include n terms for the likelihood (5.6), one term for the sparsifying prior (5.7), one term for the first part of the MRF prior (5.8), $|E|$ terms for the second part of the MRF prior (5.8) and finally, one term for the prior of the noise in the class label (5.9). The EP approximation $\mathcal{Q}(\mathbf{w},\varepsilon,\mathbf{z})$ is computed as the product of $n+|E|+3$ approximate terms that are of the form

$$\tilde{t}_i(\mathbf{w},\varepsilon,\mathbf{z}) = \tilde{s}_i\varepsilon^{\tilde{a}_i}(1-\varepsilon)^{\tilde{b}_i}\prod_{j=0}^{d}\exp\left\{-\frac{1}{2\tilde{v}_{ij}}(w_j-\tilde{m}_{ij})^2\right\}\left\{\frac{z_i+1}{2}\tilde{c}_{ij}+\frac{1-z_i}{2}\tilde{d}_{ij}\right\}\,, \tag{5.15}$$

where $\tilde{\mathbf{m}}_i=(\tilde{m}_{i0},\ldots,\tilde{m}_{id})^{\text{T}}$, $\tilde{\mathbf{v}}_i=(\tilde{v}_{i0},\ldots,\tilde{v}_{id})^{\text{T}}$, $\tilde{\mathbf{c}}_i=(\tilde{c}_{i0},\ldots,\tilde{c}_{id})^{\text{T}}$, $\tilde{\mathbf{d}}_i=(\tilde{d}_{i0},\ldots,\tilde{d}_{id})^{\text{T}}$, $\tilde{a}_i$ and $\tilde{b}_i$ are free parameters and $\tilde{s}_i$ is a constant that guarantees that $\tilde{t}_i\mathcal{Q}^{\setminus i}$ and $t_i\mathcal{Q}^{\setminus i}$ integrate to the same value. Note that we do not constrain here the variances of the approximate terms to be positive, as we did in Chapter 4. Convergence of EP is generally very good in this case and we do not need to improve it by forcing these variances to be positive. This means that on rare occasions we will not be able to compute the EP approximation of the model evidence. Nevertheless, this is not a serious problem because this approximation is not very accurate. The reason for this is the difficulty in the estimation of the normalization constant Z in (5.8).

The first step of EP is to initialize $\mathscr{Q}$ and all the $\tilde{t}_i$ to be uniform by setting $a = b = 1$ and $m_i = 0$, $v_i = +\infty$, $p_i = 0.5$, $\tilde{a}_i = \tilde{b}_i = 0$, $\tilde{m}_{ij} = 0$, $\tilde{v}_{ij} = +\infty$, and $\tilde{c}_{ij} = \tilde{d}_{ij} = 1$ for $i = 1,\ldots,n+3+|E|$ and $j = 0,\ldots,d$. Once this has been done, the method iteratively updates all the approximate terms. These operations are described in detail in Appendix D.1. The EP algorithm is run for a maximum of 500 cycles. One cycle consists of an update of every term in the factorized approximation. The algorithm stops when the absolute value of the change in the parameters of $\mathscr{Q}$ in two consecutive cycles is smaller than 10^{-6}. If the network of feature dependencies is sparse, that is, if most features are only linked to at most k neighbors and $k \ll d$, the computational cost of EP is $O(dn)$.

Once EP has converged, the predictive distribution for the label y^{test} of a new feature vector $\mathbf{x}^{\text{test}}$ is approximated as

$$\mathcal{P}(y^{\text{test}}|\mathbf{x}^{\text{test}},\mathbf{y},\mathbf{X},G,\alpha,\beta) \approx \frac{a}{a+b} + \frac{b}{a+b}\Phi\left[\frac{y^{\text{test}}\mathbf{m}^{\text{T}}\mathbf{x}^{\text{test}}}{\sqrt{(\mathbf{v}\circ\mathbf{x}^{\text{test}})^{\text{T}}\mathbf{x}^{\text{test}}}}\right], \tag{5.16}$$

where the operator ”$\circ$” denotes the Hadamard element-wise product between vectors of the same dimension and Φ is the cumulative probability function of the standard Gaussian distribution. The relevance for the i-th feature is approximated by the value of p_i obtained after convergence of the approximation (5.14) because $\mathcal{P}(z_i|\mathbf{y},\mathbf{X},G,\alpha,\beta) \approx \text{Bern}(z_i|p_i)$ and finally, the average noise in the class labels can be approximated as $\bar{\varepsilon} \approx a/(a+b)$.

5.5 Experiments

In this section, a series of experiments is carried out to evaluate the performance of NBSBC and to compare this approach with four benchmark methods for linear binary classification: the standard support vector machine (SVM) (Hastie et al., 2001; Vapnik, 1995), the network-based support vector machine (NBSVM) (Zhu et al., 2009), the graph lasso (GL) method (Jacob et al., 2009) and the sparse Bayesian classifier (SBC) that is obtained by setting $\beta = 0$ in NBSBC. The comparison between NBSBC and SBC is useful to determine whether taking into account the network of feature dependencies leads to improvements in the performance of the novel sparse Bayesian model. The hyperparameters C for SVM, λ for NBSVM and M in GL are determined in a grid search using 10-fold cross-validation. The values considered for $\log C$, $\log\lambda$ and M are 10 points uniformly distributed in the intervals $[-9,5]$, $[1,4.5]$ and $[1,10]$, respectively.

For the methods NBSBC and SBC, the selection of an appropriate value for α is a difficult task. In this study, we take a conservative approach and set $\alpha = 0$ for both NBSBC and SBC so that the first part of the MRF prior (5.8) is not informative on $z_1,\ldots,z_d$. To determine a value for β in NBSBC, we perform a grid search so that the estimate of (5.13) given by EP is as small as possible. The values considered for $\log\beta$ are 10 points uniformly distributed in $[-2,1]$. This approach performs better than a direct maximization of the EP estimate of the model evidence, probably because of the poor quality of the EP approximation for Z in (5.8), see Appendix D.1. The proposed method for selecting β also gives better results than a 10-fold cross validation grid search. On very few occasions the EP algorithm does not converge for a particular value of β. Whenever this happens, that specific value of β is discarded.

SVM is implemented using the R package *e1071* with a linear kernel. NBSVM is implemented using the R package *lpSolve*. GL is coded in C using the efficient implementation described by Roth and Fischer (2008) and Kim et al. (2006). The EP method for SBC and NBSBC is also coded in C. Finally, note that NBSVM and GL do not regularize a coefficient when the corresponding feature is not connected in the network. This can lead to significant overfitting when most nodes are disconnected. To avoid this problem, all isolated features in the network of feature dependencies of NBSVM and GL, except for the bias component, are connected to themselves. In this manner, we guarantee that every component of $\mathbf{w}$ (except w_0) is regularized in NBSVM and GL.

In all the classification problems considered, the performance of each method is evaluated by computing an estimate of the generalization error as follows: For each classification task, 100 independent random partitions of the data into training and test sets are made. In all cases, the size of the training set n is smaller than the number of features ($n < d$). We are interested in this particular type of problem because the advantages of using a network of feature dependencies should be more apparent when the amount of data available for induction is smaller than the dimensionality of the feature space. The data instances are normalized so that each feature has zero mean and unit standard deviation in the training set. Additionally, all the feature vectors are expanded with a constant component equal to 1, corresponding to the bias term. For each training set, we build different classifiers using NBSBC, SBC, SVM, GL and NBSVM. The error of each classifier is then computed on the corresponding test set. Finally, we report the average test error of each method over the 100 independent train-test partitions.

The capacity of NBSBC, NBSVM, GL and SBC for selecting relevant features is assessed on each train-test episode. For this purpose, features are ranked according to their relevance, as estimated by each method. These rankings are then compared to a target ranking in which features are ordered according to their actual relevance. The actual relevance of each feature is estimated by the absolute value of the correlation of the feature with the class label. This correlation is computed using the whole data available for each problem, not only the corresponding training set. For NBSBC and SBC, the relevance of the i-th feature is measured in terms of the approximate posterior probability of $z_i = 1$. For GL and NBSVM, this relevance is quantified by $|w_i|$ where w_i is the i-th component of the coefficient vector of the model (Guyon et al., 2002). To evaluate the accuracy of the feature rankings given by each method, we use an index of feature selection quality (Kuncheva, 2007)

$$I_{\mathrm{FSQ}}(k) = \frac{o_k d - k^2}{k(d-k)}, \quad \text{for} \quad k = 1, \ldots, d, \tag{5.17}$$

where o_k is the number of common elements between B_k, the set of k highest-ranked features, as estimated by the classification method under consideration and A_k, the set of the k most relevant features according to an empirical estimate of the actual feature relevance (absolute value of the correlation with the class label using all the available data). The index given by (5.17) satisfies four properties. First, $I_{\mathrm{FSQ}}(k)$ increases when the overlap between B_k and A_k increases. Second, the maximum value of the index ($I_{\mathrm{FSQ}}(k) = 1$) is attained when sets B_k and A_k are equal. Third, the minimum value of the index is bound from below by -1 and finally, $I_{\mathrm{FSQ}}(k)$ takes values close to zero when B_k and A_k are independently drawn. The better a method is for feature selection, the higher the value $I_{\mathrm{FSQ}}(k)$ should be. To summarize the feature selection quality of each classification method by a single number, we compute the area under the curve generated by $I_{\mathrm{FSQ}}(k)$ when k ranges from 1 to 100. Finally, in all the experiments carried out, we compare the time (in seconds) that is needed on average to train the different models (SVM, NBSVM, GL, SBC and NBSBC) on each classification problem.

The classification methods are evaluated on four real-world datasets from different domains of application, including phoneme (Hastie et al., 1995, 2001), handwritten digit (Lecun et al., 1998), precipitation (Razuvaev et al., 2008) and gene expression (Bos et al., 2009) data. The first two datasets are standard classification problems (phonemes and handwritten digits) that have been extensively analyzed in the machine learning and statistics literature. The precipitation data have been used in meteorological studies and finally, gene expression data are a standard testbed for robust prediction methods built from limited empirical evidence. A reason for considering these particular datasets is that the number of features available for prediction is fairly large (for example, $d > 150$). Furthermore, most of the individual features are either irrelevant or are not very effective for prediction, when considered in isolation. Another reason is that in all these problems the dependencies among the features are well represented by a network that is known or that can be readily determined from unlabeled data. Additionally, these dependencies have different origins. In the first three problems, they reflect relations of proximity among the features used to describe the instances: continuity in the intensity of the log-periodogram for neighboring frequencies (phonemes), similarities in the values of nearby pixels in images (handwritten digits), or coincidence of rainfall measurements collected at meteorological stations that are close to each other. The dependencies in the microarray measurements have their origin in common transcriptional regulators that control the expression level of different genes. This variety in the origins of the dependencies allows to explore whether the performance of the different prediction systems considered is affected by the nature of the relations among features.

5.5.1 Experiments with Phoneme Data

In this section, we analyze the performance of several methods for binary classification in the task of differentiating English phonemes. In particular, discriminative models are learned for the phonemes ”aa” as in ”dark” and ”ao” as the first vowel in ”water” (Hastie et al., 1995, 2001). The data correspond to utterances of these phonemes by different speakers sampled under similar conditions. Each data instance is characterized by 256 features and an associated class label. The features correspond to the logarithm of the spectral power density of the speech signal at different frequencies. They constitute a log-periodogram, which is one of several methods used to cast speech data in a form suitable for pattern recognition. Figure 5.1 displays a sample of five log-periodograms within each phoneme class. The dataset contains 1022 instances of class ”ao” and 695 instances of class ”aa”. For the construction of the network of feature dependencies, we assume that two contiguous frequencies have similar levels of relevance for classification. This relation is apparent in the top-middle plot in Figure 5.3. This plot displays the estimate of the actual feature relevance for this problem. The estimate is the absolute value of the empirical correlation between the features and the class label computed in the complete sample. The left-hand-side of Figure 5.2 shows the network of feature dependencies used by NBSBC, NBSVM and GL in this problem. The network corresponds to a chain in which feature i is connected to feature $i+1$.

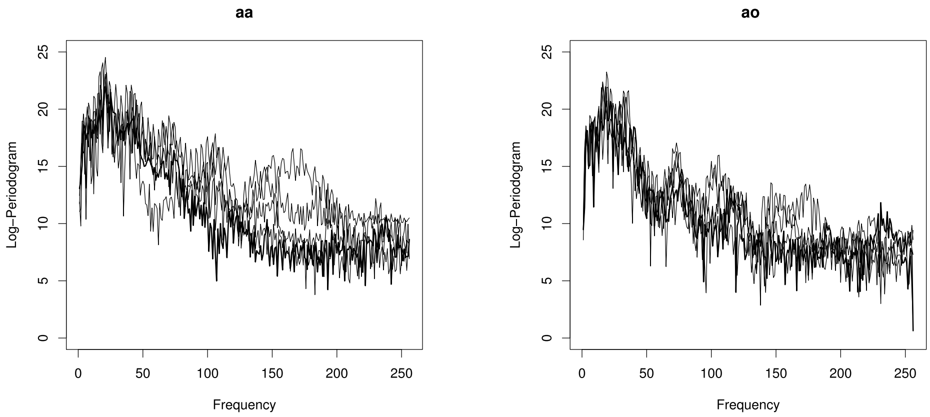

Figure 5.1: Each plot shows a sample of five log-periodograms for each phoneme class.

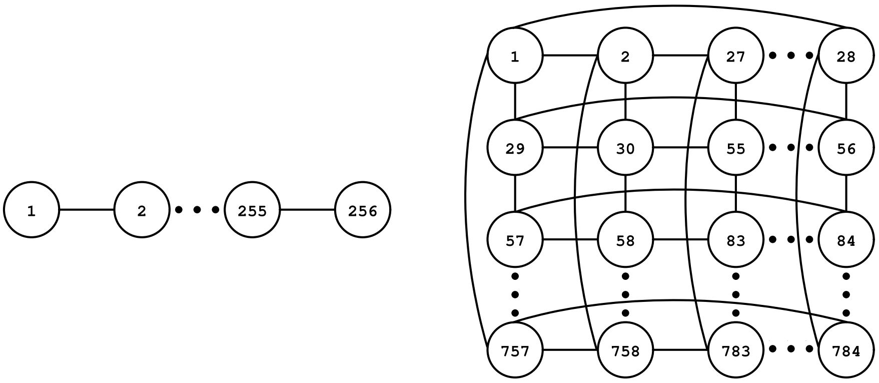

Figure 5.2: Left, Network of feature dependencies in the phoneme dataset. Right, network of feature dependencies in the handwritten digit dataset. Each node corresponds to a different feature and each edge indicates a dependence relationship between two nodes.

Table 5.1: Results for each method in the phoneme classification problem.

	SVM	**NBSVM**	**GL**	**SBC**	**NBSBC**
Avg. Test Error in %	20.66±0.01	20.24±0.01	20.55±0.01	20.19±0.01	**19.48**±0.01
Avg. Area under I_{FSQ}		29.68±11.52	22.39±5.75	40.47±4.92	**48.87**±7.87
Avg. Training Time	10.47±0.24	145.60±10.23	6.49±0.52	**0.58**±0.14	8.37±1.77

The experiments are repeated in 100 different random partitions of the original data into training (150 instances) and test sets (1567 instances). This specific value for the training set size has been selected because we are interested in evaluating the performance of the different classification methods when the dimensionality of the feature vectors (256) is larger that the number of instances available for training (150). Table 5.1 summarizes the results obtained by each method on the phoneme dataset. The first, second and third rows display the test error, the area under the index of feature selection quality and the training time in seconds for each method, averaged over the 100 train-test partitions, respectively. The corresponding standard deviations are also included in the table. The best results are highlighted in boldface. The second best results are underlined.

NBSBC is the method with lowest test error, followed by SBC. The differences between

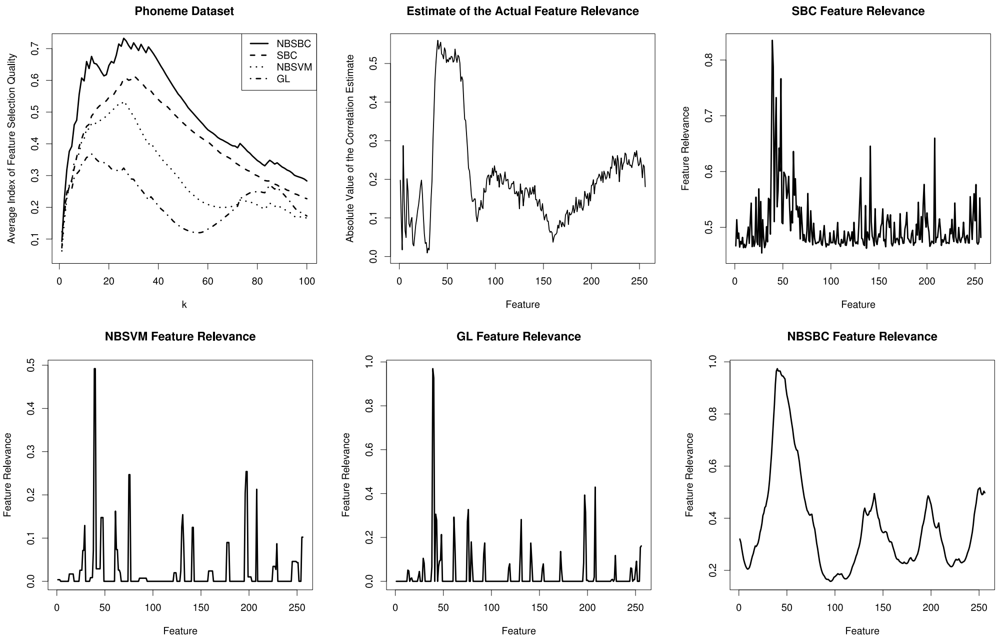

Figure 5.3: Top left, plot of the average of the index of feature selection quality for NBSBC, SBC, NBSVM and GL in the phoneme dataset. Top middle, estimate of the actual feature relevance in the phoneme dataset. Top right, bottom left, bottom middle and bottom right. Respectively, relevance for each feature given by SBC, NBSVM, GL and NBSBC. when these methods are executed on the first training set of the phoneme dataset.

these two methods are statistically significant. The p-value returned by a paired Wilcoxon test is $5.5 \cdot 10^{-13}$. NBSBC significantly improves the results of NBSVM and GL, while SVM has the poorest predictive performance. Regarding the ability to select relevant features, NBSBC is the best method. The performance of the second-best method, SBC, is very close to that of NBSBC. The poorest feature selection performance corresponds to NBSVM and GL. These results are further illustrated by the top-left of Figure 5.3, which displays the average of $I_{\text{FSQ}}(k)$ for each feature selection method. The curve for NBSBC is above the curves of SBC, NBSVM and GL. In terms of the computational cost of constructing the different classifiers, SBC has the lowest training time, followed by GL and NBSBC. Note that SBC is faster than NBSBC because in this method it is not necessary to determine β. The training costs of NBSBC, GL and SVM are similar. NBSVM is the costliest method, about 15 times slower than NBSBC.

Figure 5.3 displays the relevance given by SBC, NBSVM, GL and NBSBC to each feature in a particular realization of the phoneme problem. Two conclusions can be drawn from these graphs. First, the curve for NBSBC is fairly smooth because this method assigns similar levels of relevance to features that are connected in the underlying network. Second, NBSVM and GL favor the selection of small clusters of features that are isolated from each other.

The enhancements in performance derived from using the network of feature dependencies are expected to become less important as more training data are available. To determine whether this expectation is confirmed by the experiments, we study the performance of the different models for increasing training set sizes $n = 150, 300, 600$ and 900. Table 5.2 displays the average test error and corresponding standard deviation for each method. In all cases the best method is

Table 5.2: Average test error and standard deviation for each method in the phoneme classification problem when $n > d$.

n	SVM	NBSVM	GL	SBC	NBSBC
150	20.66±0.01	20.24±0.01	20.55±0.01	20.19±0.01	**19.48**±0.01
300	19.33±0.011	19.38±0.011	19.32±0.008	19.13±0.009	**18.92**±0.009
600	18.69±0.01	18.41±0.009	18.51±0.008	18.36±0.009	**18.33**±0.008
900	17.83±0.011	17.92±0.011	17.77±0.01	17.78±0.011	**17.77**±0.01

NBSBC. However, the differences between NBSBC and SBC become less significant as n increases. In particular, when n is large with respect to d, there seems to be enough information in the training set to perform accurate feature selection without the help of the network of features.

5.5.2 Experiments with Handwritten Digit Data

In this section we evaluate the performance of SVM, NBSVM, SBC, GL and NBSBC in the problem of automatic classification of handwritten digits. In particular, we focus on discriminating between the digits 7 and 9 in the MNIST dataset (Lecun et al., 1998). This is a challenging problem because the digits 7 and 9 present similarities. In MNIST, each digit is centered and normalized in size in a 28×28 black and white image. The pixel intensities range from 0 to 255. Additionally, the background pixel intensities are constant and equal to 0. This means that most pixels could be directly ignored by a feature selection method because their value is always 0 for all training instances. To consider a more difficult problem, in which all pixels are potential predictors for the class label, noise is added to the digitized images so that each pixel with intensity equal to 0 is replaced by a pixel whose intensity is a random number uniformly distributed between 0 and 128.

Figure 5.4 shows two sample digits from each of the two classes. The MNIST dataset contains 7293 instances of class ”7” and 6958 instances of class ”9”. To incorporate dependencies among features, we consider classifiers in which contiguous pixels in the images tend to be either both excluded or both included in the prediction model. Empirical support for this assumed dependence structure is given in the top-middle plot in Figure 5.5. This figure displays the absolute value of the linear correlation coefficient between each of the features and the class label estimated using the complete dataset. The network of feature dependencies is generated by connecting each pixel to its four nearest neighbors in the image. To avoid spurious boundary effects, the network forms a torus, in which pixels close to a given boundary are connected to pixels on the opposite boundary (see Figure 5.2). The experiments are repeated for 100 independent random partitions into a training set with 150 instances and a test set of size 14,101.

Table 5.3 summarizes the results obtained by each method in the handwritten digit dataset. The best technique in terms of test error is NBSBC, followed by SBC. The differences between these two methods are statistically significant according to a paired Wilcoxon test with a p-value lower than $2 \cdot 10^{-16}$. NBSVM is the third best method, closely followed by SVM. Finally, GL obtains the worst performance. In terms of the ability to select relevant features, the best method is NBSBC and GL is the worst one. The top-left of Figure 5.5 displays plots of the average of $I_{\text{FSQ}}(k)$ for each method. Note that the curve for NBSBC is always above those for SBC, NBSVM and GL. Regarding the time needed to build the different classifiers, NBSBC

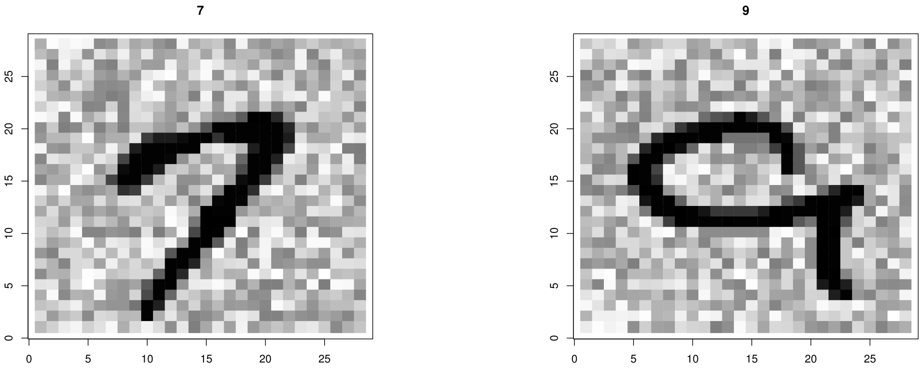

Figure 5.4: Each plot shows a sample digit from each class, that is , ”7” and ”9”.

Table 5.3: Results for each method in the handwritten digit dataset.

	SVM	**NBSVM**	**GL**	**SBC**	**NBSBC**
Avg. Test Error in %	10.32±0.015	10.23±0.013	11.18±0.012	9.18±0.009	**8.35**±0.009
Avg. Area under I_{FSQ}		41.78±9.28	31.75±3.57	54.87±5.38	**61.61**±5.59
Avg. Training Time	35.80±0.61	1992.51±211.95	29.93±2.47	**0.56**±0.04	21.32±8.49

is similar to GL and faster than SVM. In contrast, training NBSVM is about 100 times slower than NBSBC. Figure 5.5 displays the relevance assigned by SBC, NBSVM, GL and NBSBC to each feature (image pixel) in a particular realization of the handwritten digit classification problem. The relevance map given by NBSBC is composed of a few uniform patches. By contrast, NBSVM and GL tend to select individual features or small clusters of features, which are in most cases disconnected from each other.

5.5.3 Experiments with Precipitation Data

We now evaluate the accuracy of SVM, NBSVM, SBC, NBSBC and GL in the task of modeling precipitation data. In particular, we attempt to build a classifier that predicts days with zero and days with positive rainfall at a target meteorological station, given the rainfall measurements collected at other stations on the same day. The data correspond to daily precipitation measurements gathered at 223 meteorological stations in the former-USSR from 1881 until 2001 (Razuvaev et al., 2008). This is the same dataset used for the experiments of Section 3.4.3. The 223 stations are displayed in the right part of Figure 5.6.

The first task considered consists in predicting whether it rained or not in Moscow. Further experiments show that the results for this particular station are similar to those obtained when other target stations are considered. The identification number assigned to the Moscow station by the world meteorological organization (WMO) is 27,612. The instance features for the problem are the precipitation measurements collected at the other 222 meteorological stations. In the original dataset, rainfall measurements are available at all the stations for 4543 days. From these, 2217 days were dry in Moscow, leaving a total of 2326 days with positive precipitation at that station.

To construct the network of feature dependencies, we assume that two nearby stations should be either both excluded or both included in the classification model. The network used (Figure

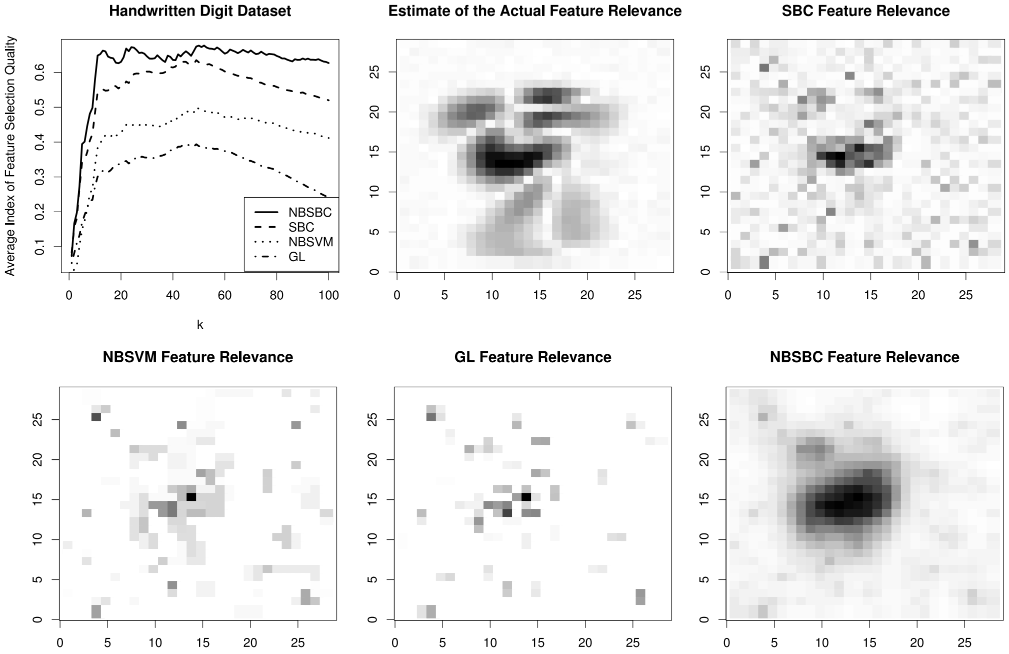

Figure 5.5: Top left, plots of the average of $I_{\mathrm{FSQ}}(k)$ for NBSBC, SBC, NBSVM and GL in the handwritten digit dataset. Top middle, estimate of the actual feature relevance in the handwritten digit dataset. Top right, bottom left, bottom middle and bottom right. Respectively, relevance for each feature given by SBC, NBSVM, GL and NBSBC. when these methods are executed on the first training set of the handwritten digit dataset. The most relevant feature is colored in black and the most irrelevant feature is colored in white.

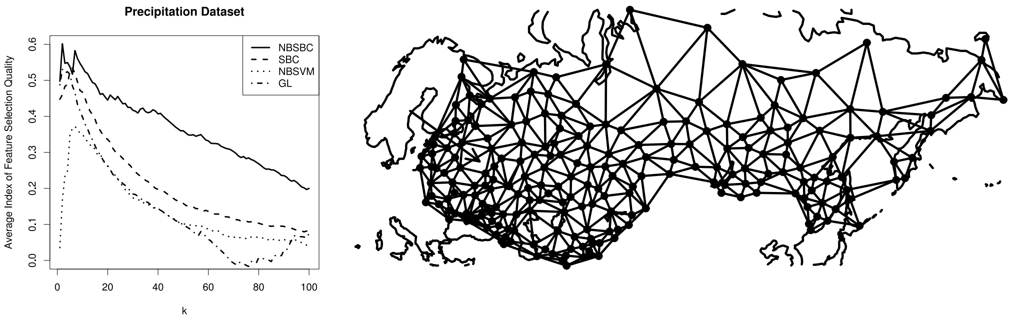

Figure 5.6: Left, average of $I_{\mathrm{FSQ}}(k)$ for each feature selection method in the precipitation dataset. Right, meteorological stations in the former-USSR. Each node corresponds to a different rainfall station. The arrow points to Moscow and represents the location of the target precipitation station with WMO number 27,612. The edges correspond to a Delaunay triangulation of all the precipitation stations except the target station. Links between stations that are more than 1000 km away from each other have been removed.

Table 5.4: Results for each method in the precipitation dataset.

	SVM	**NBSVM**	**GL**	**SBC**	**NBSBC**
Avg. Test Error in %	38.12±0.02	36.69±0.03	**32.31**±0.03	35.16±0.03	33.17±0.03
Avg. Area under I_{FSQ}		14.14±5.97	14.52±4.51	21.15±4.94	**36.71**±9.38
Avg. Training Time	9.82±0.16	254.37±19.18	14.74±0.97	**0.31**±0.12	8.36±2.92

5.6) results from a Delaunay triangulation (Renka, 1997) of the different meteorological stations, removing links between stations that are more than 1000 km apart. This type of triangulation is the dual graph of a Voronoi diagram. Voronoi diagrams are commonly used for the interpolation of scattered data in earth sciences (Sen, 2009). The experiments involve 100 independent realizations of a training set with 150 instances and a test set of size 4393.

Table 5.4 summarizes the results obtained by each method. The lowest test error is obtained by GL, followed by NBSBC. The differences between these two techniques are statistically significant according to a paired Wilcoxon test (p-value = 0.003). SBC is the third best method, followed by NBSVM and SVM. Regarding the ability to select relevant features, NBSBC is the best technique. The left of Figure 5.6 shows plots of the average of $I_{\text{FSQ}}(k)$ for each method. The curve for NBSBC is generally above the curves for the other methods. Building the classifier using NBSBC is faster than GL by a factor of ≈ 1.6 and faster than NBSVM by a factor of ≈ 30.

These experiments are also repeated for 50 different randomly selected target stations. The results show that the ranking GL (best), NBSBC, SBC, NBSVM, SVM (worst) if fairly robust: the average ranks for these methods are 1.16, 2.54, 3.02, 3.48, and 4.80, respectively.

The good results obtained by GL in this dataset are probably due to the characteristics of the feature selection process implemented by this method. Specifically, the features selected by GL correspond to edges that need not be connected. By contrast, NBSBC and NBSVM favor the selection of connected components from the original network. In this particular domain, it may be necessary to reflect other geographical information in the network, beyond the distances between the stations, (for example, the existence of geographical barriers) to provide a sufficiently accurate description of the feature dependencies. Since the sparsity pattern imposed by GL is looser, in the sense that the selected edges need not be close to each other, it is possible that the limitations of the network based exclusively on distances affect GL less severely than the other methods. Nevertheless, the differences in performance between GL and NBSBC are fairly small.

5.5.4 Experiments with Gene Expression Data from Microarray Chips

The last problem considered consists in discriminating between two classes of breast cancer patients affected by metastasis. The first class includes patients with a metastasis-free survival time (MFST), defined as the time interval between surgery and the diagnosis of metastasis, shorter than or equal to 21 months. For the second class of patients, MFST is longer than 21 months. The data consist of 204 microarray samples obtained from breast cancer patients that underwent metastatic disease (Bos et al., 2009). The data are accessible at the NCBI GEO database (Edgar et al., 2002), accession GSE12276. From the 204 samples, 104 belong to the first class of patients and 100 belong to the second class of patients. The platform used for the

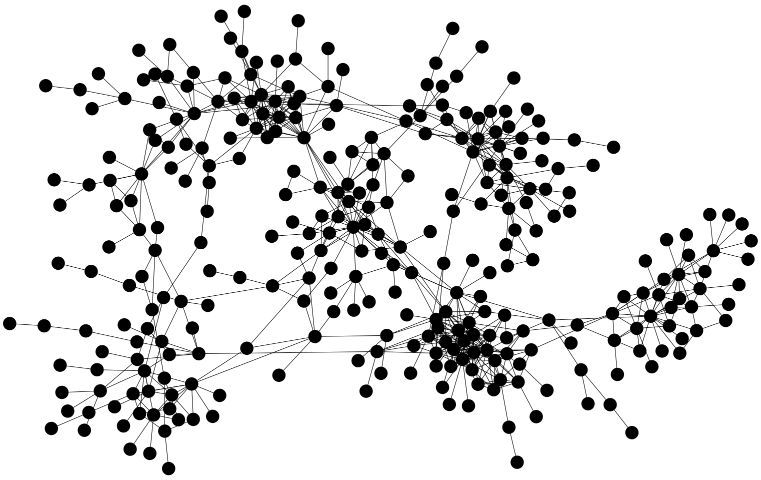

Figure 5.7: Main connected component of the network of feature dependencies for the breast cancer dataset. Each node corresponds to a different human gene. Each link indicates that the two connected genes are highly correlated.

hybridization of the tumor samples is the Affymetrix HG-U133 Plus 2.0 chip. From the set of genes whose expression is measured in this chip we consider only genes that are either cancer genes (that is, they are causally implicated in oncogenesis) (Futreal et al., 2004) or are known to interact with cancer genes (Pathway Commons, 2009). This restricts the analysis to a total of 3082 human genes. The expression level of each gene is computed as the average expression level of the probes associated to that gene in the microarray chip.

For the construction of the network of feature dependencies we use a separate set of unlabeled data corresponding to 251 tissue samples from human breast tumors (Miller et al., 2005) (data accessible at NCBI GEO database (Edgar et al., 2002), accession GSE3494). The level of gene expression in these samples is measured using two different microarray chips: the Affymetrix HG-U133A and HG-U133B platforms. This provides a total of 251 samples for each chip. From the previous 3082 genes, only 1214 are present in the HG-U133A and HG-U133B arrays. These 1214 genes are the features used in our experiment.

To construct the network of feature dependencies, we compute the correlation coefficient between expression levels for each pair of genes in the unlabeled data. Two different groups of correlation coefficient estimates are obtained, one group from the data sampled using the HG-U133A chip and another group from the data sampled using the HG-U133B chip. For each pair of genes, we obtain a final correlation estimate by averaging the corresponding coefficients in these two groups. Finally, in the network of features, two genes are linked whenever the absolute value of their correlation estimate is larger than 0.5. The resulting network of feature dependencies has 402 nodes and 766 edges. The largest connected component of this network is shown in Figure 5.7. The microarray data are normalized using the RMA method implemented in the *justRMA* routine of the R package *affy* (Gautier et al., 2004). The results reported are averages over 100 independent random partitions of the original data into a training set (2/3 of the data, 136 instances) and test set (1/3 of the data, 68 instances).

Table 5.5: Results for each method in the breast cancer dataset.

	SVM	**NBSVM**	**GL**	**SBC**	**NBSBC**
Avg. Test Error in %	33.20±0.05	34.67±0.06	36.31±0.04	32.95±0.05	**32.23**±0.05
Avg. Training Time	59.31±0.96	2083.21±186.22	38.52±2.01	**1.44**±0.35	23.83±3.28

Table 5.5 summarizes the results obtained by the different classification methods. NBSBC obtains the lowest average test error, followed by SBC. The differences between these two techniques are statistically significant according to a paired Wilcoxon test (p-value $= 10^{-4}$). The ranking of the remaining methods is SVM, NBSVM and GL, which obtains the worst results. In this case, comparisons in terms of the relevance of the selected features cannot be made. In particular, the estimates of the actual feature relevance based on the available data are not reliable because of the large sample noise and the reduced size of the dataset. In fact, the number of features is 6 times larger than the number of available data instances. Finally, in terms of average training times, NBSBC is faster than SVM by a factor of ≈ 2 and about 90 times faster than NBSVM.

5.5.5 Stability and Robustness

The classifiers built with methods that favor sparsity can be unreliable and present instabilities under slight perturbations of the training set (Haury et al., 2010; Yu et al., 2008). The lack of robustness and stability has its origin in the fact that these methods aim to generate classifiers that are accurate, but use only a small subset of the available features. The result is that one often discards features that are relevant for prediction, but are highly correlated to features already selected. The reason for this is that the improvements in performance on the training set is not sufficient to compensate the cost of including the additional features in the model. As a consequence of this behavior, the features that are selected can change significantly even when the training set is only slightly perturbed. These instabilities are specially severe when the amount of available data is limited and the dimensionality of the feature space is very high (Kalousis et al., 2007; Loscalzo et al., 2009).

NBSVM and GL can be significantly affected by these instabilities because they optimize a loss function on the training set with a penalty that favors the selection of a reduced number of features (Haury et al., 2010). This is illustrated by the graphs for NBSVM and GL in the bottom-left and bottom-middle of Figure 5.3 which display the relevance given by these methods to each feature on a particular training instance of the phoneme dataset. The most informative frequencies for prediction in this problem are a group of features clustered around the 50th frequency (see the top-middle plot in Figure 5.3). However, the bottom-left and bottom-middle plots in Figure 5.3 indicate that NBSVM and GL select only a reduced fraction of these features. Similar results can be observed for the handwritten digit dataset in Figure 5.5.

An advantage of a fully Bayesian method, such as NBSBC, over approaches that provide point-estimates of the model parameters, such as NBSVM and GL, is that they are generally more stable, specially when the data available for induction are scarce. In a fully Bayesian approach all possible values of the model parameters are considered by computing averages over the posterior distribution. Thus, the problem of selecting only a reduced number of elements among a group of highly correlated features does not appear. This is illustrated by the plot in the top-right of Figure 5.3 which displays the relevance given by the posterior distribution of SBC to each feature in a specific training instance of the phoneme dataset. In this case, the estimates of the relevance of the features around the 50th frequency are all high. Furthermore, the results are more robust when prior information is incorporated into the Bayesian model using a network of features. In particular, the values of the relevance are spatially smoothed when the MRF prior is used (see the bottom-right plots in figures 5.3 and 5.5). This smoothing reduces the magnitude of the fluctuations in the relevance values caused by small variations in the training set. The effect is similar to the reduction of noise in images by Markov random field models (Bishop, 2006; Geman et al., 1993).

To further investigate the stability of the different sparse linear classifiers when the training data are slightly perturbed, we employ an index of feature selection stability (Kuncheva, 2007). This index measures the level of agreement between the feature rankings generated by a feature selection method under different training conditions. For a given classification problem, let us assume that the prediction performance of the method under consideration is evaluated in T train/test episodes and let B_{ki} be the set with the k most relevant features as estimated by the method in the i-th train/test episode. The expression for the index of feature selection stability is

$$I_{\mathrm{FSS}}(k) = \frac{2}{T(T-1)} \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \frac{o_{ijk}d - k^2}{k(d-k)}\,, \quad \text{for} \quad k = 1,\ldots,d\,, \tag{5.18}$$

where d is the data dimensionality and o_{ijk} is the number of common elements between the sets B_{ki} and B_{kj}. This index satisfies $-1 < I_{\mathrm{FSS}}(k) \leq 1$, approaches its maximum value when the number of common features in the sets $B_{k1},\ldots,B_{kT}$ increases and takes values close to zero when the sets $B_{k1},\ldots,B_{kT}$ are independently drawn. The more stable a feature selection method is, the higher the value of $I_{FSS}(k)$.

Figure 5.8 displays plots of $I_{FSS}(k)$ for each sparse linear classifier (NBSBC, SBC, NBSVM and GL) on each of the four datasets previously analyzed (phoneme, handwritten digit, precipitation and breast cancer). The graphs are computed using the feature rankings generated by NBSBC, SBC, NBSVM and GL on each of the 100 train/test episodes of the experiments described previously. The most stable method is clearly NBSBC, followed at a significant distance by SBC. The least stable methods are NBSVM and GL.

5.6 Summary and Discussion

Some classification problems are characterized by a feature space whose dimension d is very large when compared to the number n of data instances available for induction. Under these conditions, a common approach to obtain robust and reliable classifiers is to consider sparse linear models. Sparse models often have an improved prediction accuracy and can also be used to identify those features that are more relevant for classifying new data instances. Most of the sparse classification techniques analyzed in the literature assume that the features that characterize the data instances are independent of each other. While enforcing sparsity assuming independent features is often advantageous, better results can be achieved if prior information about feature dependencies is available. This information can be encoded in the form of a network whose nodes correspond to features and whose edges represent dependence relationships

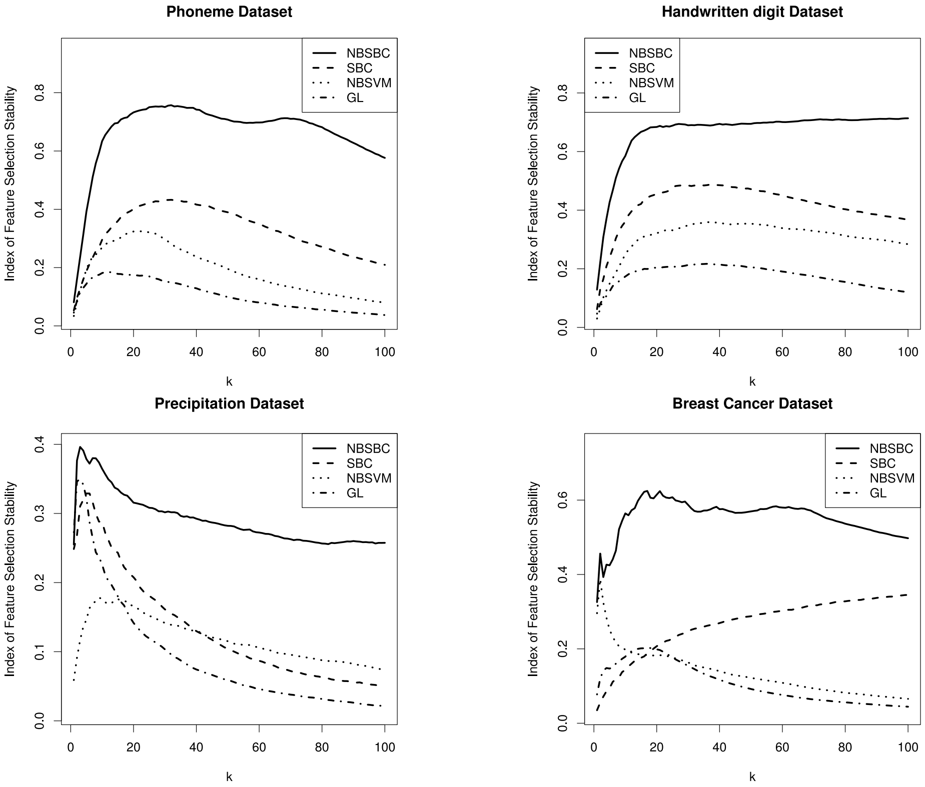

Figure 5.8: Stability of the feature rankings given by each sparse classification model (NBSBC, SBC, NBSVM and GL) on the four analyzed datasets (phoneme, handwritten digit, precipitation and breast cancer).

between features. Whenever two features are connected in the network, they are assumed to be both relevant or both irrelevant for the solution of the classification problem.

In this chapter, we have presented a new network-based sparse Bayesian classifier (NBSBC) that makes use of the information encoded in such a network to improve its prediction performance and its ability to select relevant features in problems with a reduced amount of training data and a very high-dimensional feature space. NBSBC is based on an extension of the Bayes point machine (Herbrich et al., 2001; Minka, 2001) that is capable of learning the intrinsic noise in the class labels (Hernández-Lobato and Hernández-Lobato, 2008). Sparsity in the model is favored by a spike and slab prior distribution (George and McCulloch, 1997) which is combined with a Markov random field prior (Bishop, 2006; Wei and Li, 2007) that accounts for the network of feature dependencies. Approximate Bayesian inference is implemented using the expectation propagation algorithm (EP) (Bishop, 2006; Minka, 2001). NBSBC has a fairly low computational cost. When the network of feature dependencies is sparse and the training set includes n instances and d features, the computational complexity of NBSBC is $O(nd)$.

The performance of NBSBC has been evaluated in a series of experiments with phoneme data (Hastie et al., 1995, 2001), handwritten digits (Lecun et al., 1998), precipitation data (Razuvaev et al., 2008) and gene-expression data from microarray experiments (Bos et al., 2009). For each of these datasets, we have constructed a network of features using either information specific to the problem domain or additional unlabeled data. The experiments include an exhaustive comparison with four benchmark binary classification methods: the standard support vector machine (SVM) (Hastie et al., 2001; Vapnik, 1995), the sparse Bayesian classifier (SBC) which is obtained when NBSBC ignores the network of features, the network-based support vector machine (NBSVM) (Zhu et al., 2009) and the graph lasso method (GL) (Jacob et al., 2009). Like NBSBC, NBSVM and GL assume also sparse models that consider a network to encode prior knowledge about pairwise feature dependencies. The results of these experiments show that NBSBC outperforms SVM, NBSVM, SBC and GL in all the problems analyzed except for the modeling of the precipitation data, where it ranks second. NBSBC is also very effective in the selection of features that are relevant to the solution of the classification problem.

An important property of NBSBC is the robustness of the estimates of the relative relevance of the individual features generated by this method. This stability derives from considering all the possible parameter values in the posterior distribution computed by this method. By contrast, GL and NBSVM, which employ point estimates for the model parameters, tend to discard features that, while being relevant for prediction, are highly correlated with previously selected features. Because of this, when the training data are slightly perturbed these methods generally select different groups of features. NBSBC is less affected by this instability because the Markov random field prior dampens the effect of the fluctuations in the feature relevance values arising from small variations in the training data. This effect is similar to the reduction of noise in images by Markov random field models (Bishop, 2006; Geman et al., 1993).

Chapter 6

Discovering Regulators from Gene Expression Data

This chapter describes a hierarchical sparse Bayesian model for the discovery of transcriptional regulators from gene expression data. The hierarchy incorporates the prior knowledge that only a few genes act as regulators, controlling the expression pattern of many other genes. This prior knowledge is incorporated via a spike and slab prior, in which the mixing weights are assumed to follow a hierarchical Bernoulli model. Expectation propagation is used to carry out approximate inference efficiently. The model is applied to gene expression data from the malaria parasite. Among the top ten genes identified as the most likely to be regulators, we found four genes with significant homology to transcription factors in an amoeba, another one is a known RNA regulator, three have an unknown function, and two are known not to be regulators. These results are promising, given that only gene expression data are used to identify the transcriptional regulators.

6.1 Introduction

Bioinformatics is a rich source for the application of automatic learning methods. In particular, the discovery of transcription regulatory networks has been addressed by a variety of machine learning algorithms (Gardner and Faith, 2005), including the sparse linear models with spike and slab priors described in subsection 4.4.1 of this thesis. In this chapter, we specifically focus on the identification of the genetic regulatory elements of the causative agent of severe malaria, *Plasmodium falciparum* (Hernández-Lobato et al., 2008). Several properties of this parasite necessitate a tailored method for the identification of regulators:

1. In most species gene regulation takes place at the first stage of gene expression when the DNA template is transcribed into an mRNA molecule. This transcriptional control is mediated by specific regulatory molecules called *transcription factors* (Alon, 2006). However, few specific transcription factors have been identified in Plasmodium based on sequence homology with other species (Balaji et al., 2005; Coulson et al., 2004). This

could be due to Plasmodium possessing a unique set of transcription factors or due to the presence of other mechanisms of gene regulation, for example at the level of mRNA stability or post-transcriptional regulation.

2. Compared with yeast, gene expression in Plasmodium is hardly changed by perturbations, for example by adding chemicals or changing temperature (Sakata and Winzeler, 2007). The biological interpretation of this finding is that the parasite is so narrowly adapted to its environment inside a red blood cell that it follows a stereotyped program of gene expression. From a machine learning point of view, this means that network elucidation methods relying on perturbations of the gene expression process cannot be used.

3. Similar to yeast (Spellman et al., 1998), data for three different strains of the parasite with time series of gene expression are publicly available (Llinás et al., 2006). These assay all of Plasmodium's 5600 genes for about 50 time points. In contrast to yeast, there are no ChIP-chip data available and fewer than ten transcription factor binding motifs are known (Aparicio et al., 2001).

These properties point to a vector autoregressive model, using the available gene expression time series data (point 3 above), for the identification of regulators in Plasmodium. The model should not rely on sequence homology information but it should be flexible enough to integrate sequence information in the future. This points to a Bayesian model as a favored approach since Bayesian methods can incorporate additional prior knowledge very easily (Buchan et al., 2009).

6.2 A Hierarchical Model for Gene Regulation

In this section, we introduce a hierarchical sparse Bayesian model for gene regulation which incorporates the prior knowledge that a few regulatory genes (hub genes) control by themselves the expression pattern of many other genes. Let $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ be two d-dimensional vectors whose components contain the log-concentration of mRNA for the d different genes at times t and $t+1$. As in Subsection 4.4.1, we assume a linear model for the dynamics

$$\mathbf{x}_{t+1} = \mathbf{W}\mathbf{x}_t + \boldsymbol{\sigma} \circ \mathbf{e}\,, \tag{6.1}$$

where $\mathbf{W}$ is a $d \times d$ matrix of regression coefficients that connects each gene with its parents in the underlying transcription network, $\mathbf{e}$ is a d-dimensional random vector whose elements are independent and follow a standard Gaussian distribution, $\boldsymbol{\sigma}$ is a d-dimensional vector with positive components that determine the level of noise in $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$, the operator "$\circ$" denotes the Hadamard element-wise product between vectors of the same dimension and the diagonal of $\mathbf{W}$ is assumed to be zero. Note that in this case there is a different level of noise for each gene, which should improve the accuracy of the model. Similar linear models have been used in previous investigations to describe time series of gene expression data (Beal, 2003; Jensen et al., 2007; Sabatti and James, 2006).

Let $\mathbf{X}$ denote a $d \times n$ matrix whose rows correspond to different genes and whose columns represent samples of mRNA log-concentration obtained at n consecutive time steps. Assuming

(6.1), the likelihood of $\mathbf{W}$ and $\boldsymbol{\sigma}^2$ given $\mathbf{X}$ is

$$\mathcal{P}(\mathbf{X}|\mathbf{W},\boldsymbol{\sigma}^2) = \prod_{i=1}^{d}\prod_{t=2}^{n}\mathcal{N}(x_{it}|\mathbf{w}_i\mathbf{x}_{t-1},\sigma_i^2)\,, \tag{6.2}$$

where $\boldsymbol{\sigma}^2$ is the vector with the squared components of $\boldsymbol{\sigma}$, $\mathbf{w}_i$ is the i-th row of $\mathbf{W}$, $\mathbf{x}_i$ is the i-th column of $\mathbf{X}$, x_{it} is the i-th element in $\mathbf{x}_t$ and σ_i^2 corresponds to the i-th component of $\boldsymbol{\sigma}^2$. Similarly as in Subsection 4.4.1, sparsity in $\mathbf{W}$ is enforced by assuming spike and slab prior for the elements of this matrix

$$\mathcal{P}(\mathbf{W}|\mathbf{Z}) = \prod_{i=1}^{d}\prod_{j=1}^{d}\left[z_{ij}\mathcal{N}(w_{ij}|0,v_s) + (1-z_{ij})\delta(w_{ij})\right]\,, \tag{6.3}$$

where $\mathbf{Z}$ is a $d \times d$ matrix of binary latent variables $z_{ij} = \{0,1\}$. In this case, we introduce an additional hierarchical level in the prior for $\mathbf{Z}$. Specifically, this prior is assumed to be the product of mixtures of Bernoulli terms:

$$\mathcal{P}(\mathbf{Z}|\mathbf{r}) = \prod_{i=1}^{d}\prod_{j=1,\,j\neq i}^{d}\left[r_j\mathrm{Bern}(z_{ij}|p_1) + (1-r_j)\mathrm{Bern}(z_{ij}|p_0)\right]\prod_{k=1}^{d}(1-z_{kk})\,, \tag{6.4}$$

where we have introduced another vector of binary latent variables $\mathbf{r} = (r_1,\dots,r_d)^{\mathrm{T}}$ such that $r_i = 1$ when the i-th gene is a regulator in the network and $r_i = 0$ otherwise. The relationship between regulators and their target genes suggests that $\mathcal{P}(z_{ij} = 1|r_j = 1)$ should be larger than $\mathcal{P}(z_{ij} = 1|r_j = 0)$. This is reflected in (6.4) by the positive constants p_1 and p_0 which represent precisely these two probabilities, respectively, and satisfy $p_1 > p_0$. The right-most product in (6.4) constraints the diagonal of $\mathbf{W}$ to be zero. To complete the specification of the priors for $\mathbf{W}$ and $\mathbf{Z}$, the prior of $\mathbf{r}$ is assumed to be a product of Bernoulli terms:

$$\mathcal{P}(\mathbf{r}) = \prod_{i=1}^{d}\mathrm{Bern}(r_i|p_2)\,, \tag{6.5}$$

where p_2 represents the prior probability that a randomly selected gene is a regulator in the network. Finally, the prior for $\boldsymbol{\sigma}^2$ is a product of inverse gamma distributions

$$\mathcal{P}(\boldsymbol{\sigma}^2) = \prod_{i=1}^{d}\mathrm{IG}(\sigma_i^2|\nu_i/2,\nu_i\lambda_i/2)\,, \tag{6.6}$$

where $\mathrm{IG}(x|\alpha,\beta) = \beta^{\alpha}\Gamma(\alpha)^{-1}x^{-\alpha-1}\exp(-\beta/x)$, Γ is the gamma function, λ_i is a prior estimate of the variance of the noise for the i-th gene and ν_i repents the sample size associated with that estimate. The resulting posterior distribution of the model parameters and the latent variables is given by Bayes' theorem

$$\mathcal{P}(\mathbf{W},\mathbf{Z},\mathbf{r},\boldsymbol{\sigma}^2|\mathbf{X}) = \frac{\mathcal{P}(\mathbf{X}|\mathbf{W},\boldsymbol{\sigma}^2)\mathcal{P}(\mathbf{W}|\mathbf{Z})\mathcal{P}(\mathbf{Z}|\mathbf{r})\mathcal{P}(\mathbf{r})\mathcal{P}(\boldsymbol{\sigma}^2)}{\mathcal{P}(\mathbf{X})}\,. \tag{6.7}$$

The plate notation for the proposed hierarchical model is shown in Figure 6.1 (Buntine, 1994). Given $\mathbf{X}$, we can identify those genes which are more likely to be regulators in the transcription network by computing the posterior distribution of each component of $\mathbf{r}$. This posterior $\mathcal{P}(r_i|\mathbf{X})$

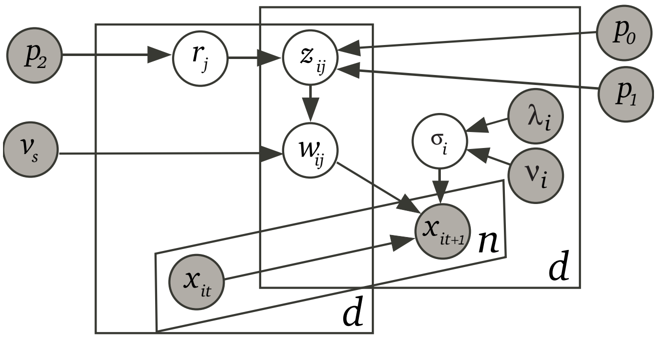

Figure 6.1: The plate notation for the hierarchical model of gene regulation. Each circle represents a different variable in the model. Shaded circles indicate that the values of the corresponding variables are known. Empty circles correspond to latent variables whose value is unknown. Rectangles or plates are used to group variables into subgraphs that repeat together. The links in the graphs represent conditional independencies between variables.

for $i = 1,\dots,d$ is obtained by marginalizing (6.7) with respect to $\mathbf{W}$, $\mathbf{Z}$, $\boldsymbol{\sigma}^2$ and all the $r_1,\dots,r_d$ except for the i-th one. These operations are too costly to be feasible in practice and some form of approximation is required. As in chapters 4 and 5, expectation propagation (EP) is used to approximate these quantities (Minka, 2001).

6.3 EP for Gene Regulation

The EP algorithm is described in detail in Section 4.3. In this section, we described how this algorithm is used to perform approximate inference in the hierarchical model described above. The posterior distribution (6.7) is approximated by the factorized exponential distribution

$$\mathcal{Q}(\mathbf{W},\mathbf{Z},\mathbf{r},\boldsymbol{\sigma}^2) = \left[\prod_{i=1}^{d}\prod_{j=1}^{d}\mathcal{N}(w_{ij}|m_{ij},v_{ij})\right]\left[\prod_{i=1}^{d}\prod_{j=1}^{d}\text{Bern}(z_{ij}|p_{ij})\right]$$
$$\left[\prod_{i=1}^{d}\text{Bern}(r_i|q_i)\right]\left[\prod_{i=1}^{d}\text{IG}(\sigma_i^2|a_i,b_i)\right], \qquad (6.8)$$

where m_{ij}, v_{ij}, p_{ij}, q_i, a_i and b_i are free distributional parameters for $i = 1,\dots,d$ and $j = 1,\dots,d$. The joint distribution of $\mathbf{W}$, $\mathbf{Z}$, $\mathbf{r}$ and $\mathbf{X}$ can be obtained as the product of $nd+3$ terms, which correspond to $(n-1)d$ terms for the likelihood (6.2), one term for the spike and slab prior on $\mathbf{W}$ (6.3), d terms for the Bernoulli mixture prior on $\mathbf{Z}$ (6.4), one term for the Bernoulli prior on $\mathbf{r}$ (6.5) and finally, one term for the inverse gamma prior on $\boldsymbol{\sigma}^2$ (6.6). In this factorization, each of the d terms for (6.4) corresponds to a different column of $\mathbf{Z}$. The EP approximation for the joint distribution is given by the product of $(n+1)d+2$ approximate terms $\tilde{t}_i(\mathbf{W},\mathbf{Z},\mathbf{r},\boldsymbol{\sigma}^2)$, where $i = 1,\dots,(n+1)d+2$. These approximate terms have the same functional form as (6.8), except that they need not be normalized. The resulting EP update operations are similar to the ones described in chapters 4 and 5. However, because we are now learning the noise levels $\sigma_1^2,\dots,\sigma_d^2$ of each individual regression, some of the EP update operations require to compute integrals which do not have a closed form expression. To avoid that, we use some simplifications in the update of the approximate likelihood terms (see Appendix E.1):

1. For computing the update of the Gaussian factors, we approximate a Student's t density by a Gaussian density with the same mean and the same variance. This approximation becomes more accurate as the degrees of freedom of the t density increase.

2. When refining the parameters of the inverse gamma (IG) factors, instead of propagating the sufficient statistics of an IG distribution, we propagate the expectations of σ_i^2 and σ_i^4. To achieve this, we perform two approximations like the one described above.

Because the exact likelihood term (6.2) is the product of $(n-1)d$ factors and the approximation (6.8) also factorizes in the components of $\mathbf{W}$, the computational complexity of EP is in this case only $O(nd^2)$. This indicates that the proposed method can be used for analyzing the expression patterns of thousands of genes. The price paid is that correlations in the posterior distribution between the components of $\mathbf{W}$ are not taken into account. Despite this limitation, the posterior approximation provided is expected to be sufficiently accurate.

In the experiments performed, we did not find necessary to constrain the variances in the approximate factors to be positive or to use damping to favor convergence. Furthermore, the computations are not affected by numerical instabilities. The EP method is stopped when the change in the parameters p_{ij} in (6.8) is less than 10^{-4} between two consecutive iterations, for $i = 1,\ldots,d$ and $j = 1\ldots,d$. Once EP has converged, we approximate the posterior probability that the i-th gene is a transcriptional regulator by the parameter q_i in (6.8). The parameters $q_1,\ldots,q_d$ can be used to select the k genes ($k \ll d$) which are more likely to be regulators given the observed data $\mathbf{X}$.

6.4 Experiments

In this section, we perform a series of experiments with simulated and actual microarray gene expression data to validate the proposed procedure for identifying transcriptional regulators. In these experiments, the hyper-parameters of the hierarchical sparse linear model are fixed as follows: $p_0 = 10^{-2}(n-1)^{-1}$, $p_0 = 10^{-1}(n-1)^{-1}$, $p_2 = (n-1)^{-1}$, $v_s = 1$, $\nu_i = 3$ and λ_i is equal to the sample variance of the i-th row in $\mathbf{X}$, for $i = 1,\ldots,d$. The resulting probabilities $q_1,\ldots,q_d$ are sensitive to the particular choice of these hyper-parameters. However the ordering of these probabilities, which determines the most likely regulators, is robust to large changes in the hyper-parameter values.

6.4.1 Experiments with Simulated Data

In this experiment, we set $n = 50$ and generated a template vector $\mathbf{v}$ with $n+1$ values from a sinusoid. We then generated 49 more vectors $\mathbf{x}_2,...,\mathbf{x}_{50}$ where $x_{it} = v_t + e_{it}$ for $i = 2,\ldots,50$ and $t = 1,\ldots,n$, where $e_{it} \sim \mathcal{N}(0,\sigma^2)$ and σ is one fourth of the sample standard deviation of the components of $\mathbf{v}$. We also generated another vector $\mathbf{x}_1$ so that $x_{1t} = v_{t+1} + e_t$ where $t = 1,\ldots,n$ and $e_t \sim \mathcal{N}(0,\sigma^2)$. In this way, $\mathbf{x}_1$ acts as a regulator for $\mathbf{x}_2,...,\mathbf{x}_{50}$. A single realization of the vectors $\mathbf{x}_1,\ldots,\mathbf{x}_{50}$ is displayed on the left of Figure 6.2. We ran the EP algorithm for gene regulation over 100 different realizations of $\mathbf{x}_1,\ldots,\mathbf{x}_{50}$. The algorithm assigned q_1 the highest value on 33 of the runs and q_1 was ranked among the top five probabilities on 74 of the runs.

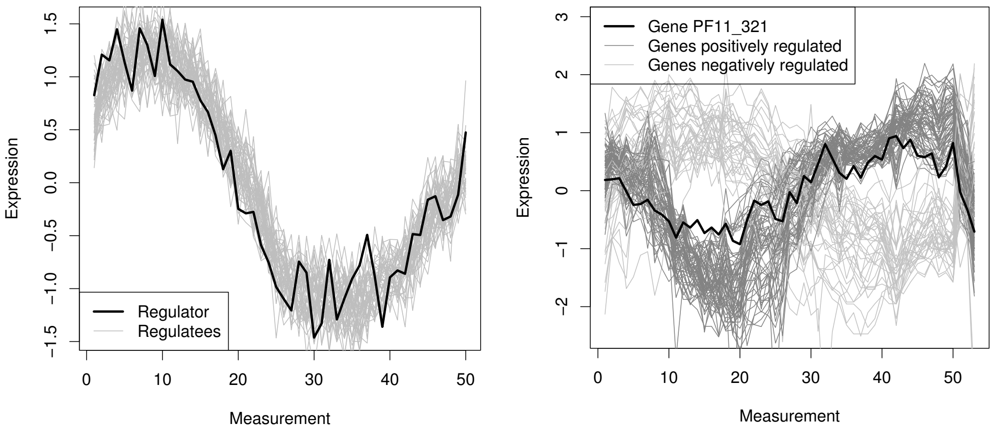

Figure 6.2: Left, plot of the vectors $\mathbf{x}_2,\ldots,\mathbf{x}_{50}$ in gray and the vector $\mathbf{x}_1$ in black. The vector $\mathbf{x}_1$ contains the expression of a regulator which would determine the expressions in $\mathbf{x}_2,\ldots,\mathbf{x}_{50}$. Right, expressions of gene PF11_321 (black) and the 100 genes which are more likely to be regulated by it (light and dark gray). Two clusters of positively and negatively regulated genes can be observed.

This indicates that the EP algorithm can successfully detect small differences in correlations and should be able to find new regulators in real microarray data.

6.4.2 Experiments with Real Microarray Data

We applied the EP method to four publicly available microarray datasets with gene expression time series. The first is a yeast cell-cycle dataset described by Spellman et al. (1998), which is commonly used as a benchmark for regulator discovery. Datasets two through four are from three different Plasmodium strains (Llinás et al., 2006). Missing values in the data were handled by the nearest neighbor method (Troyanskaya et al., 2001) using the *impute.knn* function from the R software environment (Team, 2007).

The yeast cdc15 dataset (Spellman et al., 1998) contains 23 time-series measurements for 6178 genes. We singled out 751 genes which meet a necessary criterion for cell cycle regulation (Spellman et al., 1998). The top ten genes with the highest posterior probability along with their annotation from the *Saccharomyces* Genome database (SGD project, 2007) are listed in Table 6.1. The top two genes are specific transcription factors. Additionally, YLR095w is associated with transcription regulation. As 4% of the yeast genome is associated with transcription the probability of this occurring by chance is 0.0058. Although the result is statistically significant, we were disappointed to find none of the known cell-cycle regulators (such as ACE2, FKH* or SWI*) among the top ten genes.

The three datasets for the malaria parasite (Llinás et al., 2006) contain 53 measurements (3D7), 50 measurements (Dd2) and 48 measurements (HB3). We focus on dataset for 3D7 as this is the sequenced reference strain. We singled out the 751 genes which show the highest variation as quantified by the interquartile range of the expression measurements. The top ten genes with highest probability along with their annotation from PlasmoDB (Bahl et al., 2003) are listed in Table 6.2. Recalling the motivation of our approach, that is, the paucity of known transcription

Table 6.1: Top ten genes with the highest posterior probability in the cdc15 dataset.

Rank	Gene	Annotation
1	YLR098c	DNA binding transcriptional activator
2	YOR315w	Putative transcription factor
3	YJL073w	DNAJ-like chaperone
4	YOR023c	Subunit of the ADA histone acetyl transferase complex
5	YOR105w	Dubious open reading frame
6	YLR095w	Transcription elongation
7	YOR321w	Protein O-mannosyl transferase
8	YLR231c	Kynureninase
9	YOR248w	Dubious open reading frame
10	YOR247w	Mannoprotein

Table 6.2: Top ten genes with the highest posterior probability in the 3D7 dataset.

Rank	Gene	Annotation or BLASTP hits
1	PFC0950c	25% identity to GATA TF in Dictyostelium
2	PF11_0321	25% identity to putative WRKY TF in Dictyostelium
3	PFI1210w	No BLASTP matches outside Plasmodium
4	MAL6P1.233	No BLASTP matches outside Plasmodium
5	PFD0175c	32% identity to GATA TF in Dictyostelium
6	MAL7P1.34	35% identity to GATA TF in Dictyostelium
7	MAL6P1.182	N-acetylglucosaminyl-phosphatidylinositol de-n-acetylase
8	PF13_0140	Dihydrofolate synthase/folylpolyglutamate synthase
9	PF13_0138	No BLASTP matches outside Plasmodium
10	MAL13P1.14	DEAD box helicase

factors in Plasmodium, we cannot expect to find many annotated regulators in PlasmoDB version 5.4. Thus, we list the BLASTP hits provided by PlasmoDB instead of the absent annotation. These hits were the highest scoring ones outside of the genus Plasmodium. We find four genes with a large similarity to transcription factors in Dictyostelium (a recently sequenced social amoeba) and one annotated helicase which typically functions in post-transcriptional regulation. Interestingly three genes have no known function and could be regulators. The right-most plot in Figure 6.2 displays the expression patterns of gene PF11_321 (second rank) and the 100 genes more likely to be regulated by it, as indicated by the probabilities p_{ij} given by EP. Results for the HB3 strain were similar in that five putative regulators were found, but we identified only one putative regulator (a helicase) among the top ten genes for Dd2.

6.5 Summary and Discussion

We have introduced a hierarchical Bayesian model for the discovery of genetic regulators using time series of gene expression data. The proposed hierarchal prior captures the knowledge that only a few regulators control the expression of many other genes. For efficient inference we have used the expectation propagation algorithm. Applying the proposed method on a malaria dataset, we found four genes with significant homology to transcription factors in an amoeba, one RNA regulator and three genes of unknown function.

Our approach enters a field full of alternative methods for modeling gene expression data (Beal, 2003; Jensen et al., 2007; Lucas et al., 2006; Park et al., 2007; Sabatti and James, 2006; Tienda-Luna et al., 2008). The main contributions of this chapter are: a hierarchical model to discover regulators, a tractable algorithm for fast inference in models with many interacting variables, and the application of this model to the malaria parasite. A similar hierarchical model is described by Lucas et al. (2006). The covariates in this model are a dozen external variables, which encode the experimental conditions under which the measurements were made, instead of the hundreds of expression levels of other genes, as in our model. Furthermore, the prior used by Lucas et al. (2006) enforces sparsity on the rows of $\mathbf{W}$. The prior assumption is that some genes are not influenced by any of the experimental conditions. By contrast, the hierarchical prior described here also enforces sparsity on the columns of $\mathbf{W}$ to find regulators. A possible extension of this work is to consider more sophisticated priors that incorporate information from DNA sequence data.

Chapter 7

Conclusions and Future Work

Most machine learning methods can be classified according to the complexity of the models they assume as parametric or as non-parametric. Parametric methods are generally robust to spurious patterns which are only observed by chance. However, they do not have the flexibility required to learn complex data regularities. By contrast, non-parametric methods are very flexible and can learn arbitrarily complex patterns provided that enough training data are available. However, because of their flexibility, non-parametric methods are also less robust than the parametric approaches. Flexibility and robustness are often conflicting objectives. Therefore, one cannot be improved without deteriorating the other. Selecting the optimal method to address a specific learning problem involves striking a balance between flexibility and robustness. This balance is specific to the learning problem that is being tackled. Although parametric and non-parametric methods span a broad spectrum of configurations of flexibility and robustness, there are some learning problems for which the optimal balance between flexibility and robustness cannot be attained by either of these two learning paradigms in isolation. For example, in some problems, the amount of training data is sufficiently large, so that a method that is more flexible than standard parametric approaches should be preferred. However, the performance of a fully non-parametric description of the data may be severely impaired by overfitting problems. In these cases, better results can be obtained by using semi-parametric methods, which combine the flexibility of non-parametric approaches with the robustness of parametric ones.

In this thesis, learning problems with these characteristics have been analyzed. For these problems, novel semi-parametric methods have been developed and successfully applied. In particular, semi-parametric methods can be used to construct accurate models of time series of price variations in financial markets. These time series exhibit simple trends which can be successfully captured by parametric models. However, the density functions of the innovations in these time series are not well described by standard parametric distributions. Semi-parametric methods can be used to learn these density functions in a non-parametric manner, while retaining the parametric form for the trends. Because the actual innovations are heavy-tailed, it is very difficult to construct accurate non-parametric density estimates, especially in the tails of the distribution for the innovations. To improve the quality of the approximation, the density of the innovations is estimated using an iterative algorithm based on back-transformed kernel methods. In this algorithm, the estimation of the density is performed not in the original space, but in a transformed space in which the transformed innovations are approximately normal. This method generates very accurate estimates of the parameters of the model that describe the trends, and of the density of the innovations, especially in the tails of the distribution, which is the region of interest in risk analysis.

Modeling dependencies among random variables is another learning problem in which semi-parametric methods can be useful. The dependence structure can be described using copulas. Standard parametric copulas frequently lack flexibility for capturing the complex dependencies that can be found in empirical data. If the amount of data available is scant, non-parametric copula methods are not robust and generally suffer from overfitting. The modeling difficulties are particularly severe in the tails of the distribution, where the number of samples is small. Once more, the tails are precisely the regions of interest in risk modeling. In this thesis, semi-parametric bivariate Archimedean copulas have been proposed as flexible, yet robust models for bivariate dependencies. The Archimedean copula is parametrized in terms of a generator function. In this work, a non-parametric description is used for the generator. However, this function needs to satisfy stringent constraints, which are difficult to enforce. Instead of directly modeling the generator, we propose to model a latent function that is in a one-to-one relationship with the generator. The constraints in this latent function are simpler and easier to implement. The latent function can be approximated using a basis of natural splines. Properties such as the tail dependence of the copula can be readily modeled using these splines because of their asymptotic linearity. The good overall performance of these semi-parametric copulas can be ascribed to their capacity to model complex asymmetric dependencies in the data while limiting the amount of overfitting.

Another situation in which the optimal balance between flexibility and robustness cannot be attained by either standard parametric methods or fully non-parametric approaches alone occurs when the dimensionality d of the data is very large and the size n of training set is very small ($d \gg n$). In these types of problems, even simple linear parametric models are too flexible and can be affected by overfitting. A possible way to improve the robustness of standard linear models (at the cost of reducing their flexibility) is to assume that the vector of model coefficients is sparse. In a Bayesian approach, this assumption of sparsity can be incorporated using sparsity enforcing priors, such as the spike and slab probability distribution.

The first group of learning problems with large d and small n that has been analyzed are regression tasks. The performance of linear regression models with spike and slab priors has been investigated in these problems. Approximate inference in these models is a difficult and computationally demanding problem. Gibbs sampling is a method which has often been used for this task. However, in many problems of practical interest Gibbs sampling is inefficient and long simulations of the Markov chain are required for convergence. In the analyzed problems, the algorithm expectation propagation (EP) is an efficient alternative to Gibbs sampling and has better or equivalent predictive performance at a much lower computational cost. In these datasets, the linear regression model with spike and slab priors and EP for approximate inference also outperforms other sparse linear models based on Laplace and degenerate Student's t priors. The good overall performance of the spike and slab model is explained by the superior selective shrinkage capacity of this prior. Specifically, the Laplace and degenerate Student's t priors have a single scale, which implies that the magnitudes of all the coefficients are pushed toward zero. The spike and slab prior has two scales: that of the spike and that of the slab. Therefore, only the coefficients whose posterior is dominated by the spike are shrunk. Coefficients modeled by the slab are not pushed to have zero value.

In many learning problems with large d and small n there is prior knowledge available about the dependencies that exist between specific groups of features. The performance of a learning method can be improved by taking into accounts this prior information. In particular, prior knowledge about feature dependencies can be incorporated into a sparse Bayesian linear classifier by using a spike and slab prior combined with a Markov random field. The Markov random field accounts for the dependencies between features. These dependencies are initially encoded in a network whose nodes represent features and whose edges connect two features when they are expected to be dependent. Efficient approximate inference can be implemented in this model using EP. In the set of problems analyzed, the proposed classifier is competitive with alternative state-of-the-art methods. This network-based sparse Bayesian classifier is also very robust and stable with respect to small perturbations of the training set.

The last problem with a small number of training instances and a high-dimensional space of features which has been analyzed in this thesis is the discovery of regulatory genes from time series of gene expression data. A sparse linear model can be used to address this problem. The model assumes a hierarchical spike and slab prior that reflects the fact that only a few regulatory genes control the expression pattern of many other genes. Approximate inference can be implemented in this model using EP. The resulting method identifies regulatory genes in simulated data and in real microarray data from yeast. When applied to a malaria parasite dataset, four genes with significant homology (in terms of BASTP hits) to transcription factors in an amoeba are identified among the top ten genes listed by the method.

7.1 Future Work

This section describes some directions for future research on the topics and learning methods analyzed in this thesis.

- **Semi-parametric financial time-series models:** Future research in this area includes performing a deeper analysis of back-transformed kernel methods. In this thesis, we have used transformations based on the standard Gaussian quantile function. However, if the true data density is known, better results can be obtained by using a transformation based on the Beta quantile function (Bolancé et al., 2008). For the conditional distribution of price changes, the actual density of the data is unknown. However, it would be interesting to investigate whether a combination of the Beta quantile function and normal inverse Gaussian, generalized hyperbolic or stable distributions generate a better transformation. It would also be relevant to study the influence of the transformation on the quality of the resulting density estimate in the tail of the distribution. The asymptotic convergence of the iterative method described in Section 2.3 could be analyzed using the frameworks described by Severini and Wong (1992) and Newey (1990).

- **Semi-parametric copulas:** The technique described in Chapter 3 (SPAC) is based on bivariate Archimedean copulas. These copulas are symmetric under the exchange of input variables, which can be a serious drawback in some cases. Future work could focus on semi-parametric copula models that do not have this symmetry constraint. Additionally, it would also be interesting to analyze the extension of SPAC to higher dimensions. For this, one possibility would be to combine SPAC with the methods described by Aas et al. (2009) and Kirshner (2008) for the construction of a d-dimensional copula using $d(d-1)/2$ bivariate copulas. An accurate d-dimensional copula method could be used together with the semi-parametric time-series method of Chapter 2 to obtain very accurate probabilistic models for the joint evolution of groups of asset prices in financial markets. These models could be used for the measurement of market risk or in portfolio optimization problems (Dowd, 2005; Jorion, 1997; Markowitz, 1991). Time series of precipitation data can be analyzed using more sophisticated models. For example, with Hidden Markov Models (Kirshner, 2008). SPAC could be used in this case to describe the conditional dependence structure of the temporal data.

- **Linear regression model with spike and slab prior:** The application of EP in the LRMSSP could be useful for the optimal design of experiments (Seeger, 2008) and in the related problem of *active learning* (Cohn et al., 1996). Preliminary results show that SS-EP outperforms Laplace-EP in this task when many coefficients are exactly zero in the true solution. An evaluation of the capacity of SS-EP for solving optimal design problems could be performed, in the same line as the work described by Seeger (2008). Another area of future research is the construction of recommender systems (Stern et al., 2009). The results reported by Menon and Elkan (2010) indicate that, in recommendation tasks, ridge regularization does not eliminate overfitting. It is possible that better results can be obtained in these types of problems by using spike and slab priors to enforce sparsity. Another area of future research would be to apply the LRMSSP to multi-task learning problems (Caruana, 1997), using strategies similar to the ones described by Hernández-Lobato et al. (2010). Finally, a drawback of EP is that it is not guaranteed to converge. Problems with convergence seem to be rare. However, improving the convergence of EP in specific situations is an important area of future research.

- **Network-based sparse Bayesian methods:** Li and Zhang (2010) introduce a sparse Bayesian linear regression approach with spike and slab priors and a Markov random field model for the description of transcription factor binding sites in sequences of DNA. Approximate Bayesian inference in this model is performed using Markov chain Monte Carlo methods. However, it is possible that EP provides a more efficient solution in this problem. Another direction of further research could be to investigate the differences in performance between those methods which incorporate feature dependencies by forcing some of the coefficients to take similar values (Li and Li, 2008; Sandler et al., 2008; Slawski et al., 2009) and the approach described in Chapter 5, which favors that some of the coefficients are close to zero and others are significantly different from zero.

- **Identifying genetic regulators:** The EP method described in Chapter 6 for approximate inference in the hierarchical model represents a first approach to this problem. A further step would be to incorporate in the posterior the possibility of correlations between the model coefficients, as in Chapter 4. The proposed model could also be extended to include information about DNA sequence (Jurgelenaite et al., 2009; Young et al., 2008) so that genes with similar motifs in their upstream DNA regions share common regulators.

Chapter 8

Conclusiones

La mayoría de los métodos de aprendizaje se pueden clasificar conforme a la complejidad de sus respectivos modelos en paramétricos o no paramétricos. Los métodos paramétricos son generalmente robustos ante patrones espurios cuya observación se debe solamente al azar. Sin embargo, no tienen la flexibilidad requerida para aprender regularidades complejas. Por otro lado, los métodos no paramétricos son muy flexibles y pueden aprender patrones arbitrariamente complejos siempre y cuando se disponga de suficientes datos de entrenamiento. No obstante, por este mismo motivo, los métodos no paramétricos son menos robustos que los enfoques paramétricos ante las regularidades espurias en los datos. La flexibilidad y la robustez son objectivos contradictorios. Por lo tanto, una no puede mejorarse sin deteriorarse la otra. La selección del método óptimo para un problema de aprendizaje específico implica alcanzar un equilibrio entre flexibilidad y robustez. Dicho equilibrio depende del problema que se quiera abordar. Aunque los métodos paramétricos y no paramétricos abarcan un amplio espectro de configuraciones de flexibilidad y robustez, existen ciertos problemas de aprendizaje para los que el equilibrio óptimo entre robustez y flexibilidad no se puede alcanzar utilizado solamente estos dos paradigmas de aprendizaje aisladamente. Por ejemplo, en algunos problemas, la cantidad de datos de entrenamiento puede ser suficientemente grande como para favorecer un método más flexible que los enfoques paramétricos pero, al mismo tiempo, una descripción completamente no paramétrica de los datos puede generar problemas de sobreajuste significativos. En este caso, se pueden obtener mejores resultados utilizando un método semiparamétrico, que combina la flexibilidad de los enfoques no paramétricos y la robustez de los métodos paramétricos.

En esta tesis se han analizado problemas de aprendizaje con estas características. Para dichos problemas, se han desarrollado y aplicado nuevos métodos semiparamétricos. Los métodos semiparamétricos pueden ser especialmente satisfactorios en el modelado de series temporales financieras de variaciones de precio. Estas series presentan tendencias simples que pueden ser capturadas fácilmente por modelos paramétricos. Sin embargo, las funciones de densidad de las innovaciones en dichas series no se describen de forma certera utilizando distribuciones paramétricas estándar. Los métodos semiparamétricos se pueden utilizar para aprender estas funciones de un modo no paramétrico, mientras se mantiene la formulación paramétrica para las tendencias. Debido a que la distribución de las innovaciones presenta colas pesadas, es muy difícil generar de forma certera estimaciones no paramétricas de la densidad en la cola de la distribución. Para mejorar la calidad de la aproximación, la densidad de las innovaciones se estima utilizando un algoritmo iterativo basado en métodos con núcleos de transformación y vuelta. En este algoritmo, la estimación de la densidad no se realiza en el espacio original, si no en un espacio transformado donde las innovaciones transformadas son aproximadamente normales. Finalmente, la estimación de la densidad se transforma hacia atrás sobre el espacio original. Dicho método genera estimaciones muy precisas de los parámetros de la componente paramétrica del modelo, que se utiliza para describir las tendencias, y de la densidad de las innovaciones, especialmente en la cola de la distribución, que es la región de interés para el análisis de riesgo.

Otro problema de aprendizaje donde los métodos semiparamétricos pueden ser muy útiles se corresponde con el modelado de dependencias entre variables aleatorias. La estructura de dependencia se puede describir utilizando cópulas. Las copulas paramétricas estándar suelen carecen de flexibilidad para capturar las complicadas estructuras de dependencia que se pueden encontrar en los datos empíricos. Si los datos disponibles son escasos, los métodos de cópulas no paramétricos no suelen ser robustos y tienden a sufrir problemas de sobreajuste. Los problemas de modelado son especialmente severos en las colas de la cópula, donde el número de muestras es pequeño. Una vez más, las colas son precisamente la región de interés en el modelado de riesgo. En esta tesis, las cópulas Archimedeanas semi-paramétricas se han propuesto como modelos flexibles y robustos de dependenicas entre dos variables. La cópula Archimedeana se parametriza en términos de una función generador. Los modelos introducidos se basan en una descripción no paramétrica del generador. Sin embargo, esta función tiene que satisfacer restricciones que son difíciles de modelar. En vez de describir el generador directamente, hemos propuesto modelar una función latente relacionada que se encuentra en una correspondencia uno a uno con el generador. Esta función latente se puede aproximar utilizando una base de splines cúbicos naturales y tiene que satisfacer restricciones muy sencillas. Propiedades como la dependencia en la cola de la cópula se pueden modelar fácilmente utilizando estos splines debido a su linealidad asintótica. Los buenos resultados de estas cópulas semiparamétricas se pueden explicar por la capacidad de dichos modelos para capturar dependencias asimétricas complejas mientras que se limita la cantidad de sobreajuste.

Otra situación en la que la tensión óptima entre flexibilidad y robustez no se puede alcanzar utilizando métodos paramétricos estándar o enfoques completamente no paramétricos ocurre cuando la dimensionalidad d de los datos es muy grande, pero el tamaño n del conjunto de entrenamiento es muy pequeño ($d \gg n$). En este caso, incluso los simples modelos paramétricos lineales son demasiado flexibles y pueden sufrir problemas de sobreajuste. Un modo de mejorar la robustez de los modelos lineales estándar (a costa de reducir su flexibilidad) es asumir que el vector de coeficientes del modelo es disperso. Bajo un punto de vista Bayesiano del aprendizaje automático, el supuesto de dispersidad se incorpora utilizando priors como la distribución de punta y losa.

El primer grupo de problemas con d grande y n pequeño que ha sido analizado en esta tesis incluye problemas de regresión. El desempeño del modelo de regresión lineal con prior de punta y losa se ha investigado en estos problemas. Sin embargo, la inferencia Bayesiana aproximada en dicho modelo es un problema difícil y computacionalmente exigente. El sampleo de Gibbs es un método aproximado que se ha utilizado a menudo para dicha tarea. No obstante, el coste computacional del sampleo de Gibbs es a menudo muy grande. En los problemas analizados, el método de propagación de expectaciones (PE) es una alternativa mucho más eficiente que el sampleo de Gibbs y obtiene resultados mejores o equivalentes con un coste computacional mucho menor. En estos problemas, el modelo de regresión lineal con prior de punta y losa y PE también supera los resultados de otros modelos lineales dispersos basados en priors de Laplace y de Estudiante t degenerado. Los buenos resultados en general del modelo con prior de punta y losa se explican por la mejor capacidad de encogimiento selectivo de este prior. En particular, los priors de Laplace y de Estudiante t degenerado tienen una única escala, lo que implica que las magnitudes de todos los coeficientes se comprimen hacia cero. El prior de punta y losa tiene dos escalas: una de la punta y otra de la losa. Por lo tanto, sólo se comprimen los coeficientes cuya distribución posterior se encuentra dominada por la punta. Los coeficientes modelados por la losa no se comprimen hacia el valor cero.

El rendimiento de los métodos de aprendizaje en tareas con d grande y n pequeño se puede mejorar si se conocen las características específicas del problema. En particular, la información *a priori* sobre dependencias entre atributos se puede incorporar en un clasificador Bayesiano lineal y disperso utilizando un prior de punta y losa que se combina con un campo de Markov aleatorio. El campo de Markov explica las dependencias entre atributos. Dichas dependencias se encuentran inicialmente codificadas por una red cuyos nodos se corresponden con los atributos y cuyas aristas conectan dos atributos cuando dichos atributos se espera que sean dependientes. PE permite realizar inferencia aproximada y eficiente en este modelo. En los conjuntos de datos analizados, el clasificador propuesto es competitivo con otros métodos del estado del arte. El nuevo enfoque es muy robusto ante pequeñas perturbaciones del conjunto de entrenamiento.

El último problema con un número pequeño de ejemplos de entrenamiento y un espacio de atributos de alta dimensionalidad que ha sido analizado en esta tesis se corresponde con la identificación de genes reguladores a partir de series temporales con datos de expresión genética. Un modelo lineal y disperso puede utilizarse para resolver este problema. El modelo incluye un prior de punta y losa jerárquico que incorpora el conocimiento *a priori* de que sólo unos pocos genes reguladores controlan los patrones de expresión de otros muchos genes. PE permite realizar inferencia aproximada en este modelo. El método resultante es capaz de identificar genes reguladores en datos simulados y en datos de micorarray reales de la levadura. Cuando se aplica a un conjunto de datos del parásito de la malaria, se identifican cuatro genes con una homología significativa (en términos de hallazgos BLASTP) a factores de transcripción en una ameba (de los 10 genes más relevantes considerados).

Appendix A

Appendix for Chapter 2

A.1 Adaptive Kernel Density Estimators

We describe the adaptive kernel estimator (Silverman, 1986). This method constructs a density estimate by placing kernels on each data point. However, the width of each kernel can vary from one point to another. In regions with low probability density, such as the tails, the kernels are broader, while in regions of high probability density, the kernels are more narrow. Given a sample $X_1, \dots, X_n$, this method implements the following steps:

1. Find a pilot density estimate $\hat{f}$ for the sample $X_1, \dots, X_n$. For example, the one generated by the standard kernel density estimator.

2. Define the local bandwidth factors $\lambda_i = (\hat{f}(X_i)/g)^{-\alpha}$, where g is the geometric mean of $\hat{f}(X_1), \dots, \hat{f}(X_n)$, that is,

   $$\log g = n^{-1} \sum_{i=1}^{n} \log \hat{f}(X_i) \tag{A.1}$$

   and α is a sensitivity parameter which satisfies $0 \leq \alpha \leq 1$.

3. The adaptive kernel density estimate is then defined as:

   $$\hat{f}_{\text{adap}}(x) = \frac{1}{nh} \sum_{i=1}^{n} \lambda_i^{-1} K\left(\frac{X_i - x}{h\lambda_i}\right), \tag{A.2}$$

   where, similarly to the standard kernel method, K is the kernel function and h is the bandwidth parameter.

The bandwidth parameters for the pilot density estimate and for the adaptive kernel estimator can be selected using the plug-in method (Sheather and Jones, 1991). Silverman (1986) recommends to select $\alpha = 0.5$ according to the experiments performed by Abramson (1982). Finally, in our experiments, we have used Gaussian kernels for both the pilot and the adaptive kernel estimators.

A.2 Asymptotics of Back-transformed Kernel Density Estimators

We study the asymptotic form of the back-transformed kernel density estimator. Without loss of generality, we assume that the parametric distribution used in the transformation belongs to the Pareto family, that is,

$$\bar{F}_\pi(x) = 1 - \frac{1}{x^\pi}, \tag{A.3}$$

where $\pi > 0$ and $x > 1$. Given a sample $X_1, \ldots, X_n$, the back-transformed kernel density estimator can be written as $\hat{f}(x) = h(x)\sum_{i=1}^n g_i(x)$, where the functions $h(x)$ and $g_i(x)$ are given by

$$h(x) = (2\pi)^{-1/2} \exp\left\{\frac{1}{2}\Phi^{-1}(\bar{F}_\pi(x))^2\right\} \pi x^{-\pi-1}, \tag{A.4}$$

$$g_i(x) = (2\pi h)^{-1/2} \exp\left\{-\frac{1}{2h}\left[\Phi^{-1}(\bar{F}_\pi(x)) - \Phi^{-1}(\bar{F}_\pi(X_i))\right]^2\right\} \tag{A.5}$$

and Φ^{-1} is the function of quantiles for the standard Gaussian distribution. Dominici (2003) indicates that the asymptotic behavior of $\Phi^{-1}(x)$ as $x \to 0$ is

$$\Phi^{-1}(x) \sim -\sqrt{-\log(2\pi x^2) - \log(-\log(2\pi x^2))}, \qquad x \to 0. \tag{A.6}$$

Therefore, $h(x) \sim x^{-1}\ell(x)$ when $x \to \infty$, where $\ell(x)$ is an unspecified slowly varying function (Bingham et al., 1987) and we have used the property $\Phi^{-1}(x) = -\Phi^{-1}(1-x)$. A similar analysis indicates that $g_i(x) \sim x^{-\pi/h}\ell(x)$ when $x \to \infty$ and we conclude that $\hat{f}(x) \sim x^{-\pi/h-1}\ell(x)$ when $x \to \infty$.

Since the rule used to fix the bandwidth in the back-transformed kernel density estimator is $h = 1.06n^{-1/5}$, h will be very small and $-\pi/h$ will be a large negative number as the sample size increases. Consequently, for moderate sample sizes, the tail index of $\hat{f}$ will be low enough to guarantee that the back-transformed kernel density estimate has finite second moment. For example, when $\pi = 1.5$ the Pareto distribution does not have finite variance. However, if $n = 1,000$, then it is approximately satisfied that $\hat{f}(x) \sim x^{-6.63}\ell(x)$ as $x \to \infty$, which guarantees that the second moment of the semi-parametric estimator is bounded by proposition 1.5.10 in (Bingham et al., 1987).

A.3 Tests Based on The Functional Delta Method

The tests described by Kerkhof and Melenberg (2004) are useful for detecting inaccuracies of the predicted form for the density of the innovations in the tail corresponding to losses or negative returns. Additionally, these tests allow us to validate a time series model for measuring market risk using risk measures such as Value at Risk or Expected Shortfall (Dowd, 2005; Jorion, 1997).

Given a prediction $\mathcal{P}$ for the cumulative distribution of the next day return, the Value at Risk at the α level is defined as the best result within the α fraction of best possible results under $\mathcal{P}$. In particular,

$$\rho_{\text{VaR}}(\mathcal{P}) = -\mathcal{P}^{-1}(1-\alpha), \tag{A.7}$$

where $\mathcal{P}^{-1}$ is the quantile function of $\mathcal{P}$ and the negative sign is included because the Value at Risk is usually represented as a loss. The expected shortfall at the α level is is defined as the average result obtained when the result is worse than the Value at Risk for the α fraction of best results. In particular,

$$\rho_{\text{ES}}(\mathcal{P}) = -\frac{1}{1-\alpha}\int_{-\infty}^{\mathcal{P}^{-1}(1-\alpha)} x\, d\mathcal{P}(x)\,, \tag{A.8}$$

which, likewise Value at Risk, is a positive number representing a loss.

Let $\mathcal{Q}_n$ be the empirical cumulative distribution of the test measurements generated by a time series model in the experiments of Section 2.3.2, that is,

$$\mathcal{Q}_n(x) = \sum_{i=1}^{n} \delta_{z_i}(x)\,, \tag{A.9}$$

where z_i is the i-th test measurement and δ_x is the Heaviside step function centered at x. Let $\mathcal{Q}$ be the true cumulative distribution of the test measurements $z_1,\ldots,z_n$. Then, if ρ is a functional which is Hadamard differentiable, the functional delta method (Vaart, 2000) states that

$$\sqrt{n}(\rho(\mathcal{Q}_n) - \rho(\mathcal{Q})) \approx \rho'_{\mathcal{Q}}(\sqrt{n}(\mathcal{Q}_n - \mathcal{Q})) \approx \sqrt{n}\frac{1}{n}\sum_{i=1}^{n}\rho'_{\mathcal{Q}}(\delta_{z_i} - \mathcal{Q})\,, \tag{A.10}$$

where the function $x \mapsto \rho'_{\mathcal{Q}}(\delta_x - \mathcal{Q})$ is the influence function of the functional ρ. This influence function can be computed as

$$\rho'_{\mathcal{Q}}(\delta_x - \mathcal{Q}) = \lim_{t\to 0}\frac{d}{dt}\rho((1-t)\mathcal{Q} + t\delta_x) \tag{A.11}$$

and measures the change in $\rho(\mathcal{Q})$ if an infinitesimally small part of $\mathcal{Q}$ is replaced by a point mass at x. In the last step of expression (A.10), we have made use of the linearity property of the influence function. The quantity $\rho(\mathcal{Q}_n) - \rho(\mathcal{Q})$ behaves as an average of independent random variables $\rho'_{\mathcal{Q}}(\delta_{z_i} - \mathcal{Q})$ which are known to have zero mean and finite second moments. Thus, the central limit theorem states that $\sqrt{n}(\rho(\mathcal{Q}_n) - \rho(\mathcal{Q}))$ has a normal limit distribution with mean 0 and variance $\mathbb{E}_x[\rho'_{\mathcal{Q}}(\delta_x - \mathcal{Q})^2]$ where

$$\mathbb{E}_x[\rho'_{\mathcal{Q}}(\delta_x - \mathcal{Q})^2] = \int \rho'_{\mathcal{Q}}(\delta_x - \mathcal{Q})^2\, d\mathcal{Q}(x)\,. \tag{A.12}$$

We can then use the statistic

$$S_n = \frac{\sqrt{n}(\rho(\mathcal{Q}_n) - \rho(\mathcal{Q}))}{\sqrt{\mathbb{E}_x[\rho'_{\mathcal{Q}}(\delta_x - \mathcal{Q})^2]}} \xrightarrow{d} \mathcal{N}(0,1) \tag{A.13}$$

to determine whether the difference $\rho(\mathcal{Q}_n) - \rho(\mathcal{Q})$ is statistically significant or not. Kerkhof and Melenberg (2004) show that ρ_{VaR} and ρ_{ES} are Hadamard differentiable functionals. When $\mathcal{Q}$ is standard Gaussian we obtain

$$\mathbb{E}_x[\rho'_{\text{VaR},\mathcal{Q}}(\delta_x - \mathcal{Q})^2] = \frac{\alpha(1-\alpha)}{\phi(\Phi(1-\alpha))^2}\,, \tag{A.14}$$

$$\mathbb{E}_x[\rho'_{\text{ES},\mathcal{Q}}(\delta_x - \mathcal{Q})^2] = \frac{1-\alpha-\Phi(1-\alpha)\phi(\Phi(1-\alpha))}{(1-\alpha)^2} -$$

$$\begin{aligned}&\frac{\phi(\Phi(1-\alpha))^{2}}{(1-\alpha)^{2}}+\frac{\Phi(1-\alpha)^{2}\alpha}{1-\alpha}+\\&2\Phi(1-\alpha)\frac{\phi(\Phi(1-\alpha))\alpha}{(1-\alpha)^{2}}\,,\end{aligned}\tag{A.15}$$

where Φ and ϕ are the standard Gaussian cumulative and density functions, respectively, and α represents the fraction of best results used to compute the Value at Risk and the Expected Shortfall. Finally, it is also possible to implement the test of exceedances over the Value at Risk described by Kupiec (1995) using the functional delta method. The corresponding functional for the expected number of exceedances over the Value at Risk at the level α is given by

$$\rho_{\mathrm{Exc}}(Q)=\sum_{i=1}^{n}\int(1-\delta_{Q^{-1}(1-\alpha)}(x))\,dQ(x)\,,\tag{A.16}$$

which represents the average number of elements smaller than the Value at Risk at the level α in a sample of size n from distribution Q. This functional allows us to implement the binomial test for exceedances over the Value at Risk for the α fraction of best results. We only have to calculate $\mathbb{E}_x[\rho'_{\mathrm{Exc},Q}(\delta_x-Q)^2]$ which turns out to be $(1-\alpha)\alpha n^2$. Finally, Kerkhof and Melenberg (2004) perform a complete study which compares the power of the tests for Value at Risk, Expected Shortfall and exceedances. Their results indicate that the most powerful test is the one for Expected Shortfall.

A.4 The DMPLE Density Estimation Technique

The discrete maximum penalized likelihood estimation (DMPLE) method described by Scott et al. (1980) generates a density estimate by optimizing the heights $p_1,\ldots,p_{m-1}$ of a generalized histogram at some given knots $n_1,\ldots,n_{m-1}$, which divide an interval (a,b) into m subintervals of length q. The density estimate is then a piece-wise linear function given by

$$g(x)=p_k+\frac{p_{k-1}-p_k}{q}(x-n_k)\quad\text{for}\quad x\in[n_k,n_{k+1})\tag{A.17}$$

and $g(x)=0$ when $x\notin(n_0,n_m)$, where $n_0=a$ and $n_m=b$. Given a sample $X_1,\ldots,X_n$, the estimation of the heights $p_1,\ldots,p_{m-1}$ is performed by maximizing the penalized log-likelihood

$$\sum_{i=1}^{n}\log g(X_i)-\frac{\lambda}{q}\sum_{k=1}^{m-1}(p_{k+1}-2p_k+p_{k-1})^2\tag{A.18}$$

subject to the constraints $q\sum_{k=1}^{m-1}p_k=1$ and $p_k\geq 0$ for $k=1,\ldots,m-1$, where λ is a positive regularization parameter which enforces the resulting density estimate to be smooth (Engle and González-Rivera, 1991). The previous constraints can be avoided by normalizing $p_1,\ldots,p_{m-1}$ before evaluating (A.18) so that they add up to q and by performing the optimization over $\log(p_1),\ldots,\log(p_{m-1})$. The log-likelihood can then be maximized using standard optimization methods such as the BFGS quasi-Newton algorithm (Press et al., 1992).

A.5 The SNP Density Estimator

The semi-non-parametric (SNP) method (Fenton and Gallant, 1996; Gallant and Nychka, 1987) allows to estimate simultaneously the parameters of a time-series model and the density of its innovations by maximum likelihood. The unknown density is described using an expansion in terms of Hermite functions. The class of densities considered are

$$\mathcal{F}_n = \left\{ f_n : f_n(x,\boldsymbol{\xi}) = \left[\sum_{i=0}^{p_n} \xi_i x^i\right]^2 \exp(-x^2/2)\,, \boldsymbol{\xi} \in \Xi_n \right\}, \tag{A.19}$$

$$\Xi_n = \left\{ \boldsymbol{\xi} : \boldsymbol{\xi} = (\xi_0, \xi_1, \ldots, \xi_{p_n}), \int f_n(x,\boldsymbol{\xi})\, dx = 1 \right\}, \tag{A.20}$$

where p_n is an integer that grows with the sample size n. For computational convenience, the SNP density can be expressed in terms of normalized Hermite polynomials (Fenton and Gallant, 1996), that is,

$$f_n(x,\boldsymbol{\theta}) = Z_{\boldsymbol{\theta}}^{-1} \left[\sum_{i=0}^{p_n} \theta_i w_i(x)\right]^2 \exp(-x^2/2)\,, \qquad \boldsymbol{\theta} \in \mathbb{R}^{p_n}\,, \tag{A.21}$$

where $Z_{\boldsymbol{\theta}} = \sum_{i=0}^{p_n} \theta_i^2$ is a normalization factor and the w_i are computed as follows:

$$w_0(x) = (\sqrt{2\pi})^{-1/2} \exp(-x^2/4)\,, \tag{A.22}$$

$$w_1(x) = (\sqrt{2\pi})^{-1/2} x \exp(-x^2/4)\,, \tag{A.23}$$

$$w_i(x) = \left[x w_{i-1}(x) - \sqrt{i-1} w_{i-2}(x)\right] / \sqrt{i}\,, \quad \text{for} \quad i \geq 2\,. \tag{A.24}$$

The SNP method is constrained to generate density estimates with zero mean and unit standard deviation. Let $f_n(x,\boldsymbol{\theta})$ be the unconstrained density function given by SNP. The constrained SNP density estimate is then given by $g_n(x,\boldsymbol{\theta}) = \sigma_{\boldsymbol{\theta}} f_n(x\sigma_{\boldsymbol{\theta}} + \mu_{\boldsymbol{\theta}}, \boldsymbol{\theta})$ where $\mu_{\boldsymbol{\theta}} = \int x f_n(x,\boldsymbol{\theta})\, dx$ and $\sigma_{\boldsymbol{\theta}}^2 = \int x^2 f_n(x,\boldsymbol{\theta})\, dx - \mu_{\boldsymbol{\theta}}^2$. These integrals are computed efficiently because of the orthogonality property of Hermite polynomials (Fenton and Gallant, 1996). Identification of the parameter vector $\boldsymbol{\theta}$ is performed by maximizing the log-likelihood of the model $g_n(x,\boldsymbol{\theta})$. This process can be implemented using standard optimization methods such as the BFGS quasi-Newton algorithm (Press et al., 1992). Finally, the SNP method can be used to obtain a semi-parametric model by replacing a parametric density function by $g_n(x,\boldsymbol{\theta})$ in the original description of the data.

A.6 Computational Methods

We describe the computational methods used for the implementation of the experiments of Chapter 2. All the programs have been written in the R software environment (Team, 2007). The stable density has been computed following the approach of Mittnik et al. (1997), selecting $h = 0.01$ and $q = 13$, as recommended by the authors. The density is evaluated in a grid and then interpolated with splines. The handling of splines is made with the routine *splinefun*. The stable cumulative probability is computed using the *pstable* routine from the *fBasics* package (Wuertz et al., 2004). The NIG and GHYP densities and cumulative probabilities are computed using the routines from the *fBasics* package. The plug-in method for kernel bandwidth selection (Sheather and Jones, 1991) is implemented in the routine *bw.SJ*. Samples from a NIG distribution are generated using the algorithm described by Raible (2000). Samples from a Student's t-distribution are obtained using the routine *rt*. The back-transformed kernel density estimates are evaluated on a grid with lattice constant 0.01 and then interpolated by splines. The different optimizations are performed using the routine *optim*. Constraints in the parameters of the time-series model from Section 2.2.2 are accounted for by a mapping from $\mathbb{R}$ to a space where the constraints are necessarily satisfied. Finally, quadratures, which are needed to calculate the value of the ISE, are computed using the *integrate* routine.

Appendix B

Appendix for Chapter 3

B.1 Tail Dependence and Regular Variation of f

In this appendix we analyze the relation between the indices of regular variation of f and the coefficients of upper and lower tail dependence of the corresponding Archimedean copula. The coefficients of tail dependence measure the amount of dependence in the upper-right quadrant tail or lower-left quadrant tail of a bivariate copula and they are relevant for the analysis of dependence between extreme events (Joe, 1997).

Definition B.1.1 (Lower tail dependence)**.** Let C be the copula of a random vector $(u,v)^{\mathrm{T}}$ with uniform marginals and

$$\lim_{x\downarrow 0} \mathcal{P}(v \leq x | u \leq x) = \lim_{x\downarrow 0} \frac{C(x,x)}{x} = \lambda_L \tag{B.1}$$

exists, then C has lower tail dependence if $\lambda_L > 0$ and lower tail independence if $\lambda_L = 0$. We refer to λ_L as the coefficient of lower tail dependence. Because (B.1) is a probability, λ_L satisfies $0 \leq \lambda_L \leq 1$.

Definition B.1.2 (Upper tail dependence)**.** Let C be the copula of a random vector $(u,v)^{\mathrm{T}}$ with uniform marginals and

$$\lim_{x\uparrow 1} \mathcal{P}(v > x | u > x) = \lim_{x\uparrow 1} \frac{1-2x+C(x,x)}{(1-x)} = \lambda_U \tag{B.2}$$

exists, then C has upper tail dependence if $\lambda_U > 0$ and upper tail independence if $\lambda_U = 0$. We refer to λ_U as the coefficient of upper tail dependence. Because (B.2) is a probability, λ_U satisfies $0 \leq \lambda_U \leq 1$.

Regular variation is a mathematical concept employed to study the rate of convergence of a positive function at infinity (Bingham et al., 1987). However, it is straightforward to translate the results from regular variation at infinity to regular variation at any point in the real line by performing a change of variable. Because the functions that we are considering, ϕ^{-1} and f, are defined in the unit interval, we are mainly interested in the study of regular variation in the right neighborhood of zero and in the left neighborhood of one.

Definition B.1.3 (Regular variation at zero)**.** A function f is regularly varying at zero with index $\gamma_L \in \mathbb{R}$ if for any $t > 0$

$$\lim_{x \downarrow 0} \frac{f(tx)}{f(x)} = t^{\gamma_L} . \tag{B.3}$$

$\mathcal{R}^{\gamma_L,0}$ denotes the class of functions regularly varying at zero with index γ_L.

Definition B.1.4 (Regular variation at one)**.** A function f is regularly varying at one with index $\gamma_U \in \mathbb{R}$ if for any $t > 0$

$$\lim_{x \downarrow 0} \frac{f(1-tx)}{f(1-x)} = t^{\gamma_U} . \tag{B.4}$$

$\mathcal{R}^{\gamma_U,1}$ denotes the class of functions regularly varying at one with index γ_U.

Definition B.1.5 (Slow regular variation)**.** We say that a function is slowly varying at some point in the real line when the function is regularly varying at that point with index zero.

Regular variation of the generator ϕ^{-1} at zero and at one is closely related with the coefficients of upper and lower tail dependence of the Archimedean copula. This relationship is captured by the following result due to Juri and Wüthrich (2003).

Theorem B.1.1. *Let C be an Archimedean copula with generator ϕ^{-1}.*

- *If $\phi^{-1} \in \mathcal{R}^{\gamma_L,0}$ where $-\infty \leq \gamma_L \leq 0$ then the coefficient of lower tail dependence of C is $\lambda_L = 2^{1/\gamma_L}$.*
- *If $\phi^{-1} \in \mathcal{R}^{\gamma_U,1}$ where $+\infty \geq \gamma_U \geq 1$ then the coefficient of upper tail dependence of C is $\lambda_U = 2 - 2^{1/\gamma_U}$.*

Proof. The proof of this theorem can be found in (Juri and Wüthrich, 2003). □

According to Theorem B.1.1 an Archimedean copula with upper and lower tail dependence coefficients λ_U and λ_L can be obtained by building a generator ϕ^{-1} such that

- $\phi^{-1} \in \mathcal{R}^{\gamma_L,0}$ where $\gamma_L = \log 2 / \log \lambda_L$,
- $\phi^{-1} \in \mathcal{R}^{\gamma_U,1}$ where $\gamma_U = \log 2 / \log(2 - \lambda_U)$.

We now give sufficient conditions for f so that ϕ^{-1} satisfies these two requirements.

Theorem B.1.2. *Let f be a real positive function defined in $[0,1]$ such that*

- *$f \in \mathcal{R}^{\gamma_L,0}$ where $\gamma_L = -\log 2 / \log \lambda_L$,*
- *$f \in \mathcal{R}^{\gamma_U,1}$ where $\gamma_U = -\log 2 / \log(2 - \lambda_U)$*

for $0 \leq \lambda_U, \lambda_L \leq 1$, then the associated generator ϕ^{-1} satisfies

- *$\phi^{-1} \in \mathcal{R}^{\gamma_L,0}$ where $\gamma_L = \log 2 / \log \lambda_L$,*
- *$\phi^{-1} \in \mathcal{R}^{\gamma_U,1}$ where $\gamma_U = \log 2 / \log(2 - \lambda_U)$.*

Proof. Let $f \in \mathcal{R}^{\gamma_L,0}$ where $\gamma_L \geq 0$, then by Karamata's Theorem (Bingham et al., 1987) applied to the function $x \mapsto f(1/x)$ we have that $F \in \mathcal{R}^{\gamma_L+1,0}$. A similar reasoning, applied to the function $x \mapsto F(1/x)$ leads to $\phi^{-1} \in \mathcal{R}^{-\gamma_L,0}$. A parallel derivation can be made to conclude from $f \in \mathcal{R}^{\gamma_U,1}$ where $\gamma_U \leq -1$ that $\phi^{-1} \in \mathcal{R}^{-\gamma_U,1}$. □

Corollary B.1.1. *Let f be a real positive function defined in $[0,1]$ so that*

- $f(x) = x^{\gamma_L}\ell_L(x)$ *when* $x \downarrow 0$,
- $f(x) = (1-x)^{\gamma_U}\ell_U(x)$ *when* $x \uparrow 1$,

where $\gamma_L = -\log 2/\log \lambda_L$, $\gamma_U = -\log 2/\log(2-\lambda_U)$, $0 \leq \lambda_L, \lambda_U \leq 1$ *and* $\ell_L \in \mathcal{R}^{0,0}$ *and* $\ell_U \in \mathcal{R}^{0,1}$ *are slowly varying functions at zero and at one, respectively. Then, the resulting bivariate Archimedean copula has upper and lower tail dependence coefficients equal to* λ_U *and* λ_L*, respectively.*

Proof. The results follow directly from theorems B.1.1, B.1.2 and the Characterization Theorem described by Bingham et al. (1987). □

B.2 Tail Dependence and Additive Regular Variation of g

We analyze the relation between additive regular variation of g and the coefficients of tail dependence of the corresponding Archimedean copula. The concept of additive regular variation is closely related to the concept of regular variation through a change of variables. In particular, if a function h is regularly varying at infinity with index γ_U then the function $H(x) = \log h(e^x)$ is additively regularly varying at infinity with index γ_U and vice versa. Finally, we describe the constraints that g has to satisfy in order to guarantee that $\phi^{-1}(0) = +\infty$ when the left part of this latent function is assumed to be linear.

Definition B.2.1 (Additive slow variation)**.** The real function ℓ_U is additively slowly varying as $x \to +\infty$ if

$$\lim_{x\to+\infty} (\ell_U(x+\mu) - \ell_U(x)) = 0 \tag{B.5}$$

for any μ. In a similar manner, a real function ℓ_L is additively slowly varying as $x \to -\infty$ if $x \mapsto \ell_L(-x)$ is additively slowly varying as $x \to -\infty$.

Theorem B.2.1. *Let $g : \mathbb{R} \to \mathbb{R}$ be the latent function that uniquely determines f through (3.16) and let*

- $g(x) = \gamma_L x + \ell_L(x)$ *when* $x \to -\infty$,
- $g(x) = -\gamma_U x + \ell_U(x)$ *when* $x \to +\infty$,

where $\gamma_L = -\log 2/\log \lambda_L$, $\gamma_U = -\log 2/\log(2-\lambda_U)$, $0 \leq \lambda_L, \lambda_U \leq 1$ *and* ℓ_L *and* ℓ_U *are additively slowly varying at minus infinity and at infinity respectively, then the resulting Archimedean copula has upper and lower tail dependence coefficients equal to* λ_U *and* λ_L*, respectively.*

Proof. If we replace g in (3.16) it can be verified that f satisfies (B.3) and (B.4) with indexes $\gamma_L = -\log 2/\log\lambda_L$ and $\gamma_U = -\log 2/\log(2-\lambda_U)$. Theorems B.1.1, B.1.2 complete the proof. □

Theorem B.2.2. *Let $g : \mathbb{R} \to \mathbb{R}$ be the function that uniquely determines f through (3.16) and let $g(x) = \gamma_L x + \zeta$ when $x < \delta_L$, where $\gamma_L < 0$, $-\infty < \delta_L < 0$ and ζ is a real constant, then ϕ^{-1} is not a valid generator.*

Proof. If we replace g in (3.16) it can be verified that $f \in \mathcal{R}^{\gamma_L,0}$. When $-1 < \gamma_L < 0$ we have that $F \in \mathcal{R}^{\gamma_L+1,0}$ and for $\varepsilon = \sigma(\delta_L)$, where σ is the logistic function, it is satisfied that

$$\int_0^{\varepsilon} \frac{dx}{F(x)} = \int_{1/\varepsilon}^{+\infty} z^{\gamma_L-1}\ell(z)\,dz < +\infty \tag{B.6}$$

where ℓ is a slowly varying function at infinity, $z = x^{-1}$ and the inequality is obtained by proposition 1.5.10 in (Bingham et al., 1987). Because f is a positive function, $F(x)$ is also positive for any $x \in [\varepsilon, 1]$ and $1/F(x)$ is bounded for any $x \in [\varepsilon, 1]$. Thus,

$$\int_{\varepsilon}^{1} \frac{dx}{F(x)} < +\infty, \qquad \phi^{-1}(0) = \int_0^{\varepsilon} \frac{dx}{F(x)} + \int_{\varepsilon}^{1} \frac{dx}{F(x)} < +\infty \tag{B.7}$$

and ϕ^{-1} is not a valid generator because it does not satisfy $\phi^{-1}(0) = +\infty$. When $\gamma_L \leq -1$ we have that $F(x) = +\infty$ for any $x \in (0,1]$ and $\phi^{-1}(x) = 0$ for any $x \in (0,1]$. Thus, ϕ^{-1} is not valid because it is not strictly decreasing. □

B.3 Efficient Computation of the Basis for g

We describe how to efficiently compute $\mathrm{B}^{\star} = \{N_i(x) : i = 1, \ldots, K\}$, the set of basis functions used to model g as a natural cubic spline. First, we review some properties of Archimedean copulas. Archimedean copulas are invariant to any scaling of the generator. That is, generators ϕ^{-1} and $a\phi^{-1}$, where a is a positive real constant, correspond to the same copula function. Hence, scaling f or, equivalently, adding a real constant to g does not modify the resulting Archimedean copula. In consequence, the functional space spanned by the elements of $\mathrm{B}^{\star}$ should not include any constant function.

Let $\mathrm{B} = \{B_i(x) : i = 1, \ldots, M+2\}$ be the cubic B-spline basis (de Boor, 1978) uniquely determined by the ordered sequence of knots $\xi_1, \ldots, \xi_M$. By the properties of this basis

$$\sum_{i=1}^{M+2} B_i(x) = 1 \quad \text{for all} \quad x \in [\xi_1, \xi_L]\,, \tag{B.8}$$

Therefore, if we remove any single basis function from the set, for instance $B_1(x)$, the span of the resulting basis span$\{\mathrm{B} \setminus B_1\}$ does not include any constant function. Furthermore, g can be expressed in terms of the basis $\{\mathrm{B} \setminus B_1\}$ in the interval $[\xi_1, \xi_L]$, namely

$$g(x) = \sum_{i=2}^{M+2} \alpha_i B_i(x)\,, \quad x \in [\xi_1, \xi_L]\,, \tag{B.9}$$

where $\alpha_2, \ldots, \alpha_{M+2}$ are real coefficients such that the natural boundary conditions $g''(\xi_1) = 0$ and $g''(\xi_M) = 0$ are satisfied. If $\boldsymbol{\alpha} = (\alpha_2, \ldots, \alpha_{M+2})$ and $\mathbf{C}$ is the $2 \times (M+1)$ matrix given by $C_{1,j} = B''_{j+1}(\xi_1)$ and $C_{2,j} = B''_{j+1}(\xi_L)$ then the natural boundary conditions can be compactly written as

$$\mathbf{C}\boldsymbol{\alpha} = \mathbf{0}\,. \tag{B.10}$$

Wood and Augustin (2002) specify a representation of $\boldsymbol{\alpha}$ that satisfies (B.10) without imposing and further unnecessary restrictions. Let $\mathbf{Z}$ be the $(M+1) \times (M-1)$ matrix whose columns are linearly independent and orthogonal to the rows of $\mathbf{C}$. The vector $\boldsymbol{\alpha}$ can be expressed as $\boldsymbol{\alpha} = \mathbf{Z}\boldsymbol{\theta}$ in terms of $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{M-1})$, a vector of real coefficients. This parameterization guarantees that the orthogonality condition (B.10) is fulfilled

$$\mathbf{C}\boldsymbol{\alpha} = \mathbf{C}\mathbf{Z}\boldsymbol{\theta} = \mathbf{0}\,. \tag{B.11}$$

Therefore, the set of basis functions $\mathrm{B} = \{N_i(x) : i = 1, \ldots, K\}$ where $K = M - 1$ and

$$N_i(x) = \sum_{j=1}^{M+1} \mathbf{Z}_{j,i} B_{j+1}(x)\,, \quad i = 1, 2, \ldots K \tag{B.12}$$

spans the space of natural cubic splines, with no intercept, determined by the ordered sequence of knots $\xi_1, \ldots, \xi_L$ and with domain $[\xi_1, \xi_L]$. The matrix $\boldsymbol{\Omega}$ that appears in (3.27) can be readily expressed in terms of $\mathbf{Z}$

$$\boldsymbol{\Omega} = \mathbf{Z}^{\mathrm{T}}\mathbf{H}\mathbf{Z}\,, \tag{B.13}$$

where $\mathbf{H}$ is the $(M+1) \times (M+1)$ matrix given by $\mathbf{H}_{i,j} = \int B''_{i+1}(x) B''_{j+1}(x)\, dx$. Finally, linear extrapolation of the functions $N_1, \ldots, N_{M-1}$ beyond ξ_1 and ξ_L allows to obtain a set of basis functions that covers the whole real line. Figure 3.3 displays an example of such a basis. The basis functions are uniquely determined by a total of 10 knots marked with small circles.

B.4 Computation of the Gradient

In this section, we describe the expression for the gradient of the logarithm of the Archimedean copula density (3.11) with respect to the i-th coefficient θ_i of the expansion of g in terms of natural cubic splines (3.26). For this, we use

$$\frac{dg(x)}{d\theta_i} = N_i(x)\,, \tag{B.14}$$

$$\frac{df(x)}{d\theta_i} = f(x) N_i[\sigma^{-1}(x)]\,, \tag{B.15}$$

$$\frac{dF(x)}{d\theta_i} = \int_0^x f(x) N_i[\sigma^{-1}(y)]\, dy\,, \tag{B.16}$$

$$\frac{d\phi^{-1}(x)}{d\theta_i} = -\int_x^1 \frac{1}{F(y)^2} \frac{dF(y)}{d\theta_i} dy\,, \tag{B.17}$$

$$\frac{dC(u,v)}{d\theta_i} = F[C(u,v)] \frac{d\phi^{-1}[C(u,v)]}{d\theta_i} - \frac{d\phi^{-1}(u)}{d\theta_i} - \frac{d\phi^{-1}(v)}{d\theta_i}\,, \tag{B.18}$$

where N_i is the i-th basis function in the expansion of g and σ^{-1} is the inverse of the logistic function. The logarithm of (3.11) is

$$\log c(u,v) = \log f[C(u,v)] + \log F[C(u,v)] - \log F(u) - \log F(v)\,. \tag{B.19}$$

Following (Gagliardini and Gourieroux, 2007), we obtain the derivative of (B.19) with respect to θ_i, namely

$$\begin{aligned}\frac{d\log c(u,v)}{d\theta_i} &= f[C(u,v)]^{-1}\left\{\frac{df[C(u,v)]}{d\theta_i} + f'[C(u,v)]\frac{dC(u,v)}{d\theta_i}\right\} + \\ &\quad F[C(u,v)]^{-1}\left\{\frac{dF[C(u,v)]}{d\theta_i} + f[C(u,v)]\frac{dC(u,v)}{d\theta_i}\right\} - \\ &\quad F(u)^{-1}\frac{dF(u)}{d\theta_i} - F(v)^{-1}\frac{dF(v)}{d\theta_i}\,,\end{aligned} \tag{B.20}$$

where the most costly part for its evaluation is the computation of the integral functions given by (B.16) and (B.17). These integrals are evaluated using quadrature methods as described in the next section.

B.5 Technical Details on the Estimation of g

We describe the technical details concerning the process of estimating g given a dataset $\mathcal{D} = \{U_i, V_i\}_{i=1}^N$. All the computations are performed in the software environment R (Team, 2007). The number of knots is fixed to be relatively large, Lambert (2007) employed 20 knots and we follow his recommendation since that number should be enough to capture rather complex dependence structures. With regard to knot placement, boundary knots are fixed to match quantiles 0.01 and 0.99 of (3.23). The remaining knots are placed at uniform quantiles of (3.23) in the interval spanned by the two boundary knots. The basis for g is computed using the routine *ns* from the R package *splines* with the option *intercept* deactivated. This routine computes the set of basis functions as described in Section B.3 and calculates the matrix $\mathbf{Z}$ using the QR factorization of $\mathbf{C}^\text{T}$. The matrix $\mathbf{\Omega}$ in (3.27) is obtained as $\mathbf{\Omega} = \mathbf{Z}^\text{T}\mathbf{H}\mathbf{Z}$ where $\mathbf{H}$ is computed using the routine *bsplinepen* from the R package *fda*. We maximize (3.27) with respect to $\boldsymbol{\theta}$ using the routine *constrOptim*. A 10-fold cross-validation grid search is carried out in order to find the optimal β. The logarithm of the grid values of β are the integers 1, 2, 3 and 4. These values correspond to a smoothing level that ranges from medium to high. Once the search process is completed, the value of the smoothing parameter with highest average log-likelihood on the leave-out sets is employed in the maximization of $\text{PLL}(\mathcal{D}|\boldsymbol{\theta},\beta)$ with respect to $\boldsymbol{\theta}$.

The evaluations of $\log\mathcal{L}(\mathcal{D}|\boldsymbol{\theta})$ and its gradient require the computation of quadratures of several functions and inversion of the generator of the Archimedean copula. Quadratures are numerically approximated in a fine grid using a method based on the Newton-Cotes formulas. The grid is obtained by mapping a uniform grid in the real line to the unit interval using the logistic function. Evaluation of the integrals at points outside the grid is carried out by numerical interpolation with natural splines, the *splinefun* function from the R package *splines* is employed for this task. Before doing the interpolation, the unit interval is expanded to $\mathbb{R}$ using the inverse of the logistic function and the set $\mathbb{R}^+$ is expanded to $\mathbb{R}$ using the natural logarithm. Using these operations we manage to interpolate a smooth function so that natural splines give a better fit. Inversion of the generator is done by evaluating the function in a fine grid and interpolating the points obtained.

The maximization of $\mathrm{PLL}(\mathcal{D}|\boldsymbol{\theta},\beta)$ with respect to $\boldsymbol{\theta}$ requires using good initial estimates. Thus, we follow Lambert (2007) and make use of the relation between $K(x) = \mathcal{P}\{C(U,V) \leq x\}$ and ϕ^{-1}, where C is an Archimedean copula with generator ϕ^{-1} and U and V are independent random variables uniformly distributed. Given that $\mathcal{D} = \{U_i,V_i\}_{i=1}^{N}$ is a sample from C, Genest and Rivest (1993) indicate that an estimate of K is

$$\hat{K}_1(x) = \frac{1}{N+1}\sum_{i=1}^{N}\mathbf{I}\left\{\hat{C}(U_i,V_i) \leq x\right\}, \tag{B.21}$$

where $\hat{C}(u,v)$ is given by (3.22). A smoothed version of this estimate is given by the cumulative probability of a back-transformed kernel density estimate (Wand et al., 1991), namely

$$\hat{K}_2(x) = \frac{1}{N}\sum_{i=1}^{N}\Phi_h\left\{\Phi_1^{-1}(x) - \Phi_1^{-1}\left[\hat{C}(U_i,V_i)\right]\right\}, \tag{B.22}$$

where Φ_h is the cumulative probability function of a Gaussian with zero mean and standard deviation h. The band-width h is computed using the plug-in method of Sheather and Jones (1991), implemented in the routine *bw.SJ*. The corresponding estimate of λ would be $x - \hat{K}_2(x)$, although this estimate does not generally allow to obtain a valid generator. Hence, following Lambert (2007) we obtain a valid estimate of λ through the constrained minimization of

$$\sum_{v\in\mathcal{V}}\left\{x - \hat{K}_2(x) - \hat{\lambda}(v|\boldsymbol{\alpha})\right\}^2 \tag{B.23}$$

with respect to $\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is a vector of 20 weights, $\hat{\lambda}$ corresponds to the parameterization of λ used by Lambert (2007) and $\mathcal{V}$ is a set of 100 points that lay on a uniform grid in the unit interval. The minimizer $\hat{\boldsymbol{\alpha}}$ of (B.23) can be found solving a quadratic programming problem. For such task we employ the routine *solve.QP* from the R package *quadprog*. The corresponding estimate of the generator is

$$\hat{\phi}^{-1}(x) = \exp\left\{\int_0^x \frac{1}{\hat{\lambda}(s|\hat{\boldsymbol{\alpha}})}\,ds\right\}. \tag{B.24}$$

The integral function that appears in (B.24) is approximated as described above. The estimate of f is

$$\hat{f}(x) = \frac{1 - \hat{\lambda}'(x|\hat{\boldsymbol{\alpha}})}{\hat{\phi}^{-1}(x)} \tag{B.25}$$

and the corresponding estimate of g is $\hat{g}(x) = \log\hat{f}\,[\sigma(x)]$, where σ is the logistic function. A good initialization for the process of maximizing $\mathrm{PLL}(\mathcal{D}|\boldsymbol{\theta},\beta)$ with respect to $\boldsymbol{\theta}$ is therefore the minimizer of the penalized residual sum of squares

$$\mathrm{PRSS}(\boldsymbol{\theta},\beta) = (\mathbf{y} - \mathbf{N}\boldsymbol{\theta})^{\mathrm{T}}(\mathbf{y} - \mathbf{N}\boldsymbol{\theta}) + \beta\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\Omega}\boldsymbol{\theta} \tag{B.26}$$

where $\mathbf{N}$ is the $(M-1)\times(M-1)$ matrix given by $\mathbf{N}_{i,j} = N_j(\xi_{i+1})$, the $N_1,\ldots,N_{M-1}$ basis functions are given by (B.12), $\xi_1,\ldots,\xi_L$ are the knots that uniquely determine the functions $N_1,\ldots,N_{M-1}$ and $\mathbf{y}$ is the $(M-1)$-dimensional vector given by $\mathbf{y} = \{\hat{g}(\xi_2)-\hat{g}(\xi_1),\ldots,\hat{g}(\xi_L)-\hat{g}(\xi_1)\}$. Note that only $M-1$ square terms appear in (B.26), the reason for this is that all the functions g that are a linear combination of $N_1,\ldots,N_{M-1}$ satisfy $g(\xi_1)=0$. This is also the reason for the i-th component of $\mathbf{y}$ to be $\hat{g}(\xi_{i+1})-\hat{g}(\xi_1)$ instead of just $\hat{g}(\xi_{i+1})$. Finally, to guarantee that the resulting generator is valid, we must find the minimizer of (B.26) that satisfies the linear constraint given by (3.29). This is again a quadratic programming problem.

B.6 Comparison of SPAC with Parametric Archimedean Copulas

The semi-parametric Archimedean copula estimator proposed in Chapter 3 (SPAC) is compared in this appendix with standard parametric Archimedean copulas. The parametric Archimedean models considered correspond to the Gumbel (GUM), Clayton (CLA) and Frank (FRA) copulas (Nelsen, 2006). These are the Archimedean copulas most cited in the literature (Malevergne and Sornette, 2006). The parameters of the parametric models are estimated by the method of maximum likelihood. For the comparison of the different methods we have used the financial and the precipitation data described in Chapter 3. In particular, each copula model is tested in the modeling of the dependence structure between the 32 pairs of financial assets and the 32 pairs of precipitation stations. The experimental protocol is the same as in Chapter 3.

Table B.1 displays the average test log-likelihood obtained by each Archimedean copula model (GUM, CLA, FRA and SPAC) in the experiments with financial data. For each pair of financial assets, the best and second best methods are highlighted in bold face and underlined, respectively. The rightmost column in this table contains the p-value generated by a paired t test between the two best techniques on each estimation problem. Overall, the best performing method is the semi-parametric estimator, SPAC, which obtains the highest average test log-likelihood in 21 out of the 32 estimation problems. The second best method is the Frank parametric copula (FRA) with the highest rank in 10 problems.

The different methods are statistically compared against each other using the approach of Demšar (2006). Each pair of financial assets forms a different task for the testing framework. A Friedman test rejects the hypothesis that all 4 methods have an identical performance in the 32 tasks analyzed (p-value = $1.6\cdot10^{-14}$). Pairwise comparisons between the average ranks of the different copula estimation methods using a Nemenyi test at $\alpha = 0.05$ are summarized in Figure B.1. This test indicates that SPAC and FRA are statistically superior to GUM and CLA. However, it is not able to discriminate between SPAC and FRA. The reason is that the Nemenyi test performs multiple hypothesis testing and consequently, it has little power for discriminating between two methods only. As more powerful approach, we perform a paired Wilcoxon test between SPAC and FRA. In this case, the null hypothesis that both methods have a similar performance is rejected at $\alpha = 0.05$ with a resulting p-value equal to $2.2\cdot10^{-3}$.

Table B.2 displays the average test log-likelihood obtained by each Archimedean copula model (GUM, CLA, FRA and SPAC) in the experiments with precipitation data. The best performing method is SPAC, with the best results in 28 out of the 32 estimation problems. The

Table B.1: Average log-likelihood of each Archimedean model on the financial test data.

Assets		τ	GUM	CLA	FRA	SPAC	*t*-Test
WMB	WMT	0.09	3.19	**6.09**	5.61	5.97	$2.7 \cdot 10^{-01}$
KO	LSTR	0.14	8.83	9.83	**14.70**	13.90	$4.3 \cdot 10^{-13}$
FDX	FE	0.14	12.34	9.75	**14.69**	14.35	$4.9 \cdot 10^{-03}$
CHRW	CNP	0.14	8.88	12.23	15.23	**15.63**	$7.2 \cdot 10^{-03}$
EXC	EXPD	0.15	11.35	13.51	15.31	**15.41**	$6.0 \cdot 10^{-01}$
OSG	PCG	0.15	10.69	17.71	16.53	**17.90**	$2.7 \cdot 10^{-01}$
PEG	PFE	0.15	12.02	16.05	17.09	**17.80**	$7.1 \cdot 10^{-07}$
LUV	MCD	0.16	13.44	16.06	17.43	**18.21**	$6.6 \cdot 10^{-05}$
DIS	DUK	0.15	16.66	13.87	18.32	**18.84**	$1.4 \cdot 10^{-03}$
NI	NSC	0.17	17.30	14.81	**21.03**	20.66	$1.1 \cdot 10^{-02}$
AES	AIG	0.16	17.03	19.24	19.12	**21.71**	$3.7 \cdot 10^{-19}$
PG	R	0.18	19.60	17.36	**23.03**	22.89	$3.1 \cdot 10^{-01}$
FPL	GE	0.18	16.48	20.28	23.02	**23.33**	$1.0 \cdot 10^{-01}$
AA	AEP	0.17	17.55	21.36	21.75	**23.66**	$1.5 \cdot 10^{-15}$
SO	T	0.18	17.86	21.10	23.26	**23.88**	$7.7 \cdot 10^{-03}$
XOM	YRCW	0.18	18.10	19.78	**25.14**	24.83	$4.2 \cdot 10^{-02}$
MRK	MSFT	0.19	21.83	17.07	**25.76**	25.65	$3.7 \cdot 10^{-01}$
MMM	MO	0.18	19.84	19.94	**25.68**	24.93	$2.9 \cdot 10^{-04}$
D	DD	0.19	20.33	21.84	**26.47**	26.37	$6.6 \cdot 10^{-01}$
JNJ	JPM	0.18	21.14	23.64	25.11	**27.19**	$3.6 \cdot 10^{-14}$
ALEX	AMR	0.20	21.96	23.74	**30.53**	29.87	$1.6 \cdot 10^{-05}$
UTX	VZ	0.22	28.31	25.18	33.85	**33.88**	$9.0 \cdot 10^{-01}$
CAL	CAT	0.22	27.17	28.43	**35.74**	35.23	$1.3 \cdot 10^{-02}$
INTC	JBHT	0.24	42.08	27.08	41.19	**44.22**	$4.1 \cdot 10^{-11}$
GM	GMT	0.24	36.85	36.88	45.06	**45.21**	$5.3 \cdot 10^{-01}$
AXP	BA	0.25	37.40	47.90	47.70	**52.06**	$8.0 \cdot 10^{-25}$
HD	HON	0.27	48.99	43.24	55.88	**56.84**	$3.2 \cdot 10^{-05}$
BNI	C	0.27	50.02	51.38	58.18	**61.36**	$5.1 \cdot 10^{-19}$
CNW	CSX	0.31	68.76	64.68	76.95	**80.36**	$5.4 \cdot 10^{-18}$
UNP	UPS	0.32	72.89	60.46	78.15	**80.86**	$5.9 \cdot 10^{-19}$
HPQ	IBM	0.33	78.78	70.84	84.70	**89.44**	$2.4 \cdot 10^{-23}$
ED	EIX	0.33	71.42	82.56	87.02	**93.15**	$9.4 \cdot 10^{-27}$

second best performing method is the Frank parametric copula (FRA) which obtains the highest rank in 3 problems.

Similarly as in the experiments with financial data, the different methods are statistically compared against each other using the approach of Demšar (2006). A Friedman test rejects the hypothesis that all 4 methods have an identical performance in the 32 tasks analyzed (p-value = $2.7 \cdot 10^{-17}$). Pairwise comparisons between the average ranks of the different copula estimation methods using a Nemenyi test at $\alpha = 0.05$ are summarized in Figure B.2. This test indicates that SPAC is statistically superior to all the parametric Archimedean copulas in the modeling of the precipitation data.

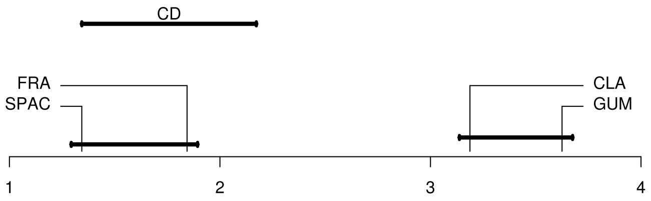

Figure B.1: All to all comparison of the Archimedean copula estimation techniques by the Nemenyi test on the financial data.

Table B.2: Average log-likelihood of each Archimedean model on the precipitation test data.

PIN[a]	WMON[b]		τ	GUM	CLA	FRA	SPAC	t-Test
1	36974	38353	0.08	**6.64**	2.10	4.64	5.80	$8.4 \cdot 10^{-13}$
2	30949	30965	0.14	11.57	8.68	**13.80**	13.57	$6.8 \cdot 10^{-03}$
3	32061	32098	0.16	20.46	8.35	18.64	**20.84**	$8.0 \cdot 10^{-02}$
4	31735	31829	0.17	26.52	10.68	23.82	**27.14**	$1.3 \cdot 10^{-04}$
5	38696	38836	0.19	28.51	16.48	26.06	**28.60**	$6.6 \cdot 10^{-01}$
6	32540	32564	0.21	34.76	11.86	33.71	**38.85**	$1.9 \cdot 10^{-31}$
7	37235	37472	0.22	40.15	18.54	36.91	**40.30**	$5.1 \cdot 10^{-01}$
8	38457	38599	0.23	40.22	17.25	38.45	**42.27**	$1.2 \cdot 10^{-13}$
9	33393	33631	0.23	42.25	21.94	39.38	**42.67**	$4.5 \cdot 10^{-02}$
10	26406	26422	0.25	41.18	29.75	46.93	**47.27**	$1.4 \cdot 10^{-02}$
11	29231	29430	0.26	47.06	34.59	52.80	**53.08**	$1.9 \cdot 10^{-01}$
12	35188	35394	0.29	54.44	40.15	**62.99**	62.88	$6.0 \cdot 10^{-01}$
13	34731	34747	0.29	56.27	42.81	64.64	**65.79**	$3.0 \cdot 10^{-07}$
14	33815	33837	0.30	61.46	35.50	67.55	**69.05**	$1.0 \cdot 10^{-12}$
15	35358	35542	0.29	64.27	42.97	64.43	**67.79**	$6.2 \cdot 10^{-24}$
16	36034	36177	0.30	62.31	45.16	68.53	**69.60**	$1.3 \cdot 10^{-05}$
17	28434	28440	0.28	67.69	31.43	61.44	**69.28**	$4.1 \cdot 10^{-10}$
18	33345	33377	0.31	65.69	47.47	74.42	**74.57**	$4.2 \cdot 10^{-01}$
19	31594	31707	0.30	68.35	45.20	71.31	**73.84**	$6.4 \cdot 10^{-16}$
20	34122	34139	0.32	64.45	45.41	**76.44**	75.88	$3.1 \cdot 10^{-03}$
21	24944	24951	0.30	71.88	33.47	68.04	**74.14**	$7.6 \cdot 10^{-17}$
22	30054	30253	0.30	72.67	31.79	67.25	**75.83**	$1.3 \cdot 10^{-17}$
23	31388	31329	0.31	70.89	45.10	72.98	**74.66**	$4.3 \cdot 10^{-09}$
24	30777	30673	0.31	71.42	41.80	74.56	**77.17**	$2.2 \cdot 10^{-16}$
25	22820	22837	0.32	77.03	45.40	78.40	**81.55**	$1.1 \cdot 10^{-22}$
26	26730	26850	0.32	77.83	47.24	81.86	**84.29**	$1.5 \cdot 10^{-19}$
27	27553	27648	0.32	77.36	50.14	77.79	**81.50**	$4.1 \cdot 10^{-26}$
28	30823	30925	0.32	77.02	45.27	79.37	**82.41**	$1.6 \cdot 10^{-16}$
29	23724	23921	0.32	78.10	41.67	80.06	**84.43**	$6.7 \cdot 10^{-33}$
30	31915	31960	0.30	86.51	28.54	69.80	**89.62**	$4.4 \cdot 10^{-26}$
31	27037	27333	0.34	84.25	58.57	90.95	**92.01**	$7.2 \cdot 10^{-05}$
32	30393	31004	0.34	103.62	46.59	92.51	**105.18**	$1.1 \cdot 10^{-09}$

[a] Pair Identification Number.

[b] World Meteorological Organization Station Numbers.

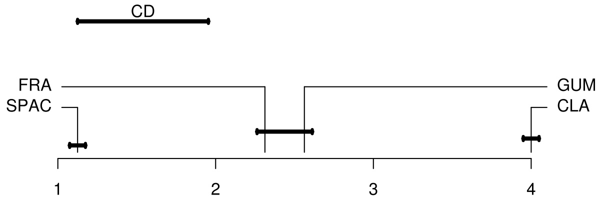

Figure B.2: All to all comparison of the Archimedean copula estimation techniques by the Nemenyi test on the precipitation data.

Appendix C

Appendix for Chapter 4

C.1 Gibbs Sampling in the LRMSSP

Approximate Bayesian inference in the LRMSSP has been traditionally implemented using Gibbs sampling. This method works by randomly sampling $\mathbf{w}$ and $\mathbf{z}$ from (4.5). Expectations over the actual posterior are then approximated by expectations over the generated samples. For the implementation of the Gibbs sampling method, we follow Lee et al. (2003) and sample $\mathbf{z}$ from its marginal distribution after integrating $\mathbf{w}$ out, which speeds up the computations. The central operation in Gibbs sampling is the evaluation of the conditional probability of $z_i = 1$ when all the other components of $\mathbf{z}$ stay fixed. This probability can be efficiently computed using the framework described by Tipping and Faul (2003). First of all, we introduce some notation. Let $\mathbf{\Phi}$ be the $n \times d$ design matrix, let $\mathbf{t}$ be the n-dimensional target vector and let $\mathbf{C_z} = \sigma_0^2\mathbf{I} + \mathbf{\Phi}\mathbf{A_z}^{-1}\mathbf{\Phi}^\text{T}$ be an $n \times n$ matrix. $\mathbf{A_z}$ is a $d \times d$ diagonal matrix whose i-th diagonal element α_i satisfies $\alpha_i = v_s^{-1}$ when $z_i = 1$ and $\alpha_i = \infty$, otherwise. The log marginal probability of $\mathbf{z}$ is then

$$\log \mathcal{P}(\mathbf{z}) = -\frac{1}{2}\log|\mathbf{C_z}| - \frac{1}{2}\mathbf{y}^\text{T}\mathbf{C_z}^{-1}\mathbf{y} + s_\mathbf{z}\log p_0 + (d - s_\mathbf{z})\log(1-p_0) + K\,, \tag{C.1}$$

where $s_\mathbf{z}$ is the number of components of $\mathbf{z}$ that are equal to one and K is a constant that does not depend on $\mathbf{z}$. Let φ_i denote the i-th column of $\mathbf{\Phi}$ and let $\mathbf{\Sigma_z}^{-1} = \mathbf{A_z} + \sigma_0^{-2}\mathbf{\Phi}^\text{T}\mathbf{\Phi}$. Following Tipping and Faul (2003), when $\mathbf{z}$ is updated by switching z_i from one to zero, the corresponding decrement in (C.1) is

$$\log\sqrt{\frac{1}{1+v_s s_i}} + \frac{q_i^2}{2(v_s^{-1}+s_i)} + \log\frac{p_0}{1-p_0}\,, \tag{C.2}$$

where q_i and s_i are given by

$$q_i = \frac{Q_i}{1-v_s S_i}\,, \qquad\qquad s_i = \frac{S_i}{1-v_s S_i} \tag{C.3}$$

and Q_i and S_i are computed using

$$Q_i = \sigma_0^{-2}\varphi_i^{\mathrm{T}}\mathbf{t} - \sigma_0^{-4}\varphi_i^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Sigma}_{\mathbf{z}}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}\,, \tag{C.4}$$

$$S_i = \sigma_0^{-2}\varphi_i^{\mathrm{T}}\varphi_i - \sigma_0^{-4}\varphi_i^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Sigma}_{\mathbf{z}}\mathbf{\Phi}^{\mathrm{T}}\varphi_i\,, \tag{C.5}$$

where $\mathbf{\Phi}$ and $\mathbf{\Sigma}_{\mathbf{z}}$ involve in (C.4) and (C.5) only those features whose corresponding components of $\mathbf{z}$ are equal to one before the update. In a similar manner, when $\mathbf{z}$ is updated by switching z_i from zero to one, the resulting increment in (C.1) is also given by (C.2). However, q_i and s_i are now fixed as $q_i = Q_i$ and $s_i = S_i$, where Q_i and S_i are obtained using (C.4) and (C.5). This allows us to efficiently compute the conditional probability of z_i as a function of q_i and s_i only, namely

$$\mathcal{P}(z_i = 1|\mathbf{z}_{\setminus i}) = p_0\left[p_0 + (1-p_0)\exp\left\{\frac{-q_i^2}{2(v_s^{-1}+s_i)}\right\}\sqrt{1+v_s s_i}\right]^{-1}\,, \tag{C.6}$$

where $\mathbf{z}_{\setminus i}$ represents $z_1,\ldots,z_d$ but with z_i omitted and q_i and s_i are obtained using either the rule $q_i = Q_i$, $s_i = S_i$ or (C.3), depending on whether $z_i = 1$ is satisfied or not during the computation of Q_i and S_i by (C.4) and (C.5). Gibbs sampling generates a sample of $\mathbf{z}$ by running randomly through all the components of this vector and drawing a value for each component according to the probability given by (C.6). The bottle-neck of this process is the computation of $\mathbf{\Sigma}_{\mathbf{z}}$ in (C.4) and (C.5), which requires $O(s_{\mathbf{z}}^2 n)$ operations when $s_{\mathbf{z}} < n$. Nevertheless, Gibbs sampling only modifies $\mathbf{\Sigma}_{\mathbf{z}}$ by adding or removing a single feature from this matrix each time. This allows us to save unnecessary computations by storing $\mathbf{L}_{\mathbf{z}}$, the Cholesky decomposition of $\mathbf{\Sigma}_{\mathbf{z}}^{-1}$, that is, $\mathbf{\Sigma}_{\mathbf{z}}^{-1} = \mathbf{L}_{\mathbf{z}}\mathbf{L}_{\mathbf{z}}^{\mathrm{T}}$ where $\mathbf{L}_{\mathbf{z}}$ is a lower triangular matrix. The cost of updating $\mathbf{L}_{\mathbf{z}}$ after switching on or off a single component of $\mathbf{z}$ is $O(s_{\mathbf{z}}^2)$ when efficient methods for modifying matrix factorizations are used (Gill et al., 1974). Once $\mathbf{L}_{\mathbf{z}}$ is available, we can compute $\mathbf{\Sigma}_{\mathbf{z}}$ in only $O(s_{\mathbf{z}}^2)$ operations. After having generated a Gibbs sample for $\mathbf{z}$, we draw a sample of $\mathbf{w}$ conditioning to the current value of $\mathbf{z}$. For this, we set to zero the components of $\mathbf{w}$ whose corresponding $z_1,\ldots,z_d$ are equal to zero. The other components of $\mathbf{w}$, represented by the $s_{\mathbf{z}}$-dimensional vector $\mathbf{w}_{\mathbf{z}}$, are sampled using

$$\mathbf{w}_{\mathbf{z}} = \sigma_0^{-2}\mathbf{\Sigma}_{\mathbf{z}}\mathbf{\Phi}\mathbf{t} + \mathbf{r}^{\mathrm{T}}\mathbf{L}'_{\mathbf{z}}\,, \tag{C.7}$$

where $\mathbf{\Phi}$ and $\mathbf{\Sigma}_{\mathbf{z}}$ involve in this formula only those features whose corresponding components of $\mathbf{z}$ are active, $\mathbf{r}$ is an $s_{\mathbf{z}}$-dimensional random vector whose components follow independent standard Gaussian distributions, that is, $\mathbf{r} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ and $\mathbf{L}'_{\mathbf{z}}$ is the Cholesky decomposition of the matrix $\mathbf{\Sigma}_{\mathbf{z}}$ used in this formula. The cost of generating a Gibbs sample of $\mathbf{w}$ is $O(s_{\mathbf{z}}^3)$. When $n < d$, the computational complexity of the method is determined by the operations involved in the sampling of $\mathbf{z}$. The expected value of $s_{\mathbf{z}}$ is $p_0 d$. Hence, generating a total of k samples of $\mathbf{z}$ and $\mathbf{w}$ has a cost equal to $O(kp_0^2 d^3)$ and often $k \gg d$ for accurate inference. Finally, the algorithm must be initialized to a solution with large posterior probability so that the Gibbs sampler is less likely to get stuck into a sub-optimal mode of the posterior distribution. For this, we follow a greedy process that starts off by setting $z_1,\ldots,z_d$ to zero and then activates the component of $\mathbf{z}$ that generates the largest reduction of the square error of the model on the training set. This activation step is repeated until $p_0 d$ components of $\mathbf{z}$ are equal to one, where $p_0 d$ is rounded to its closest integer.

C.2 Product and Quotient Rules

We describe the product and quotient rules for Gaussian and Bernoulli distributions, which are useful for the derivation of EP in the LRMSSP. Let $\mathcal{N}(\mathbf{x}|\mathbf{m},\mathbf{V})$ be a d-dimensional Gaussian density with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{V}$. The product of two Gaussian densities is another Gaussian density, although no longer normalized:

$$\mathcal{N}(\mathbf{x}|\mathbf{m}_1,\mathbf{V}_1)\mathcal{N}(\mathbf{x}|\mathbf{m}_2,\mathbf{V}_2) \propto \mathcal{N}(\mathbf{x}|\mathbf{m}_3,\mathbf{V}_3)\,, \tag{C.8}$$

where $\mathbf{V}_3 = (\mathbf{V}_1^{-1}+\mathbf{V}_2^{-1})^{-1}$, $\mathbf{m}_3 = \mathbf{V}_3(\mathbf{m}_1^{\mathrm{T}}\mathbf{V}_1^{-1}+\mathbf{m}_2^{\mathrm{T}}\mathbf{V}_2^{-1})$ and the normalization constant in the right part of (C.8) is

$$(2\pi)^{-d/2}\frac{|\mathbf{V}_3|^{1/2}}{|\mathbf{V}_1|^{1/2}|\mathbf{V}_2|^{1/2}}\exp\left\{-\frac{1}{2}\left(\mathbf{m}_1^{\mathrm{T}}\mathbf{V}_1^{-1}\mathbf{m}_1+\mathbf{m}_2^{\mathrm{T}}\mathbf{V}_2^{-1}\mathbf{m}_2-\mathbf{m}_3^{\mathrm{T}}\mathbf{V}_3^{-1}\mathbf{m}_3\right)\right\}. \tag{C.9}$$

Similarly, the quotient of two Gaussian densities is another Gaussian density that is no longer normalized:

$$\frac{\mathcal{N}(\mathbf{x}|\mathbf{m}_1,\mathbf{V}_1)}{\mathcal{N}(\mathbf{x}|\mathbf{m}_2,\mathbf{V}_2)} \propto \mathcal{N}(\mathbf{x}|\mathbf{m}_3,\mathbf{V}_3)\,, \tag{C.10}$$

where $\mathbf{V}_3 = (\mathbf{V}_1^{-1}-\mathbf{V}_2^{-1})^{-1}$, $\mathbf{m}_3 = \mathbf{V}^{-3}(\mathbf{m}_1^{\mathrm{T}}\mathbf{V}_1^{-1}-\mathbf{m}_2^{\mathrm{T}}\mathbf{V}_2^{-1})$ and the normalization constant in the right part of (C.10) is in this case

$$(2\pi)^{d/2}\frac{|\mathbf{V}_3|^{1/2}|\mathbf{V}_2|^{1/2}}{|\mathbf{V}_1|^{1/2}}\exp\left\{-\frac{1}{2}\left(\mathbf{m}_1^{\mathrm{T}}\mathbf{V}_1^{-1}\mathbf{m}_1-\mathbf{m}_2^{\mathrm{T}}\mathbf{V}_2^{-1}\mathbf{m}_2-\mathbf{m}_3^{\mathrm{T}}\mathbf{V}_3^{-1}\mathbf{m}_3\right)\right\}. \tag{C.11}$$

Let $\mathrm{Bern}(x|\sigma(p)) = x\sigma(p)+(1-\sigma(p))(1-x)$ be a Bernoulli distribution, where $x\in\{0,1\}$, p is a real parameter, σ is the logistic function (4.13) and $\sigma(p)$ represents the probability of $x=1$. The product of two Bernoulli distributions is another Bernoulli distribution, but no longer normalized:

$$\mathrm{Bern}(x|\sigma(p_1))\mathrm{Bern}(x|\sigma(p_2)) \propto \mathrm{Bern}(x|\sigma(p_3))\,, \tag{C.12}$$

where $p_3 = p_1+p_2$ and the normalization constant in the right part of (C.12) is $\sigma(p_1)\sigma(p_2)+(1-\sigma(p_1)(1-\sigma(p_2)$. Finally, the quotient of two Bernoulli distributions is also a Bernoulli distribution which is no longer normalized:

$$\frac{\mathrm{Bern}(x|\sigma(p_1))}{\mathrm{Bern}(x|\sigma(p_2))} \propto \mathrm{Bern}(x|\sigma(p_3))\,, \tag{C.13}$$

where $p_3 = p_1 - p_2$ and the normalization constant in the right part of (C.13) is computed as $\sigma(p_1)/\sigma(p_2)+(1-\sigma(p_1)/(1-\sigma(p_2)$.

C.3 Derivation of the EP Update Operations

In this appendix, we describe the EP update operations for minimizing $\mathrm{D}_{\mathrm{KL}}(t_i\mathcal{Q}^{\setminus i}\|\tilde{t}_i\mathcal{Q}^{\setminus i})$ with respect to $\tilde{t}_i$ for the cases $i=1$ and $i=2$. The update operation for $\tilde{t}_3$ is not discussed because it is trivial. To obtain the update rules for $\tilde{t}_1$ and $\tilde{t}_2$ we follow two steps. First, $\mathcal{Q}$ is refined so that $\mathrm{KL}(t_i\mathcal{Q}^{\setminus i}\|\mathcal{Q})$ is minimized and second, $\tilde{t}_i$ is updated as the ratio between $\mathcal{Q}$ and $\mathcal{Q}^{\setminus i}$ for $i=1$

and $i = 2$. These operations are performed using the normalized versions of Q and $Q^{\setminus i}$, that is, $\mathcal{Q}$ and $\mathcal{Q}^{\setminus i}$, respectively.

C.3.1 The First Approximate Term

To minimize $\mathrm{D}_{\mathrm{KL}}(t_1 Q^{\setminus 1} \| Q)$ we first compute $\mathcal{Q}^{\setminus 1}$, which has the same functional form as $\mathcal{Q}$ because all the $\tilde{t}_i$ belong to the same family of exponential distributions. The parameters of $\mathcal{Q}^{\setminus 1}$ are obtained from the ratio between Q and $\tilde{t}_1$ (see Appendix C.2), namely

$$\mathcal{Q}^{\setminus 1}(\mathbf{w},\mathbf{z}) = \prod_{i=1}^{d} \mathcal{N}(w_i|\tilde{m}_{2i},\tilde{v}_{2i})\mathrm{Bern}(z_i|\sigma(\tilde{p}_{2i}+\tilde{p}_{3i}))\,. \tag{C.14}$$

The KL divergence is minimized when $\mathcal{Q}$ is updated so that the fist and second marginal moments of $\mathbf{w}$ and the first marginal moment of $\mathbf{z}$ are the same under $\mathcal{Q}$ and $t_1\mathcal{Q}^{\setminus 1}Z_1^{-1}$, where Z_1 is the normalization constant of $t_1\mathcal{Q}^{\setminus 1}$. Therefore, the update rule for $\mathcal{Q}$ is

$$\mathbf{m}^{\text{new}} = \mathbb{E}[\mathbf{w}]\,, \qquad \mathbf{v}^{\text{new}} = \mathrm{diag}(\mathbb{E}[\mathbf{w}\mathbf{w}^{\mathrm{T}}] - \mathbb{E}[\mathbf{w}]\mathbb{E}[\mathbf{w}]^{\mathrm{T}})\,, \qquad \mathbf{p}^{\text{new}} = \sigma^{-1}(\mathbb{E}[\mathbf{z}])\,, \tag{C.15}$$

where $\mathrm{diag}(\cdot)$ extracts the diagonal of a square matrix, all the expectations are taken with respect to $t_1\mathcal{Q}^{\setminus 1}Z_1^{-1}$ and $\sigma^{-1}((x_1,\ldots,x_d)^{\mathrm{T}}) = (\sigma^{-1}(x_1),\ldots,\sigma^{-1}(x_d))^{\mathrm{T}}$ where σ^{-1} is the logit function (4.21). Computing the expectation of $\mathbf{z}$ under $t_1\mathcal{Q}^{\setminus 1}Z_1^{-1}$ is trivial. To compute the first and second moments of $\mathbf{w}$, we note that the likelihood factor t_1 has a Gaussian form on $\mathbf{w}$ which is characterized by an inverse precision matrix $\mathbf{\Lambda}_1$ and a mean vector $\mathbf{m}_1$ such that $\mathbf{\Lambda}_1 = \sigma_0^{-2}\mathbf{X}^{\mathrm{T}}\mathbf{X}$ and $\mathbf{\Lambda}_1\mathbf{m}_1 = \sigma_0^{-2}\mathbf{X}^{\mathrm{T}}\mathbf{y}$. Because $\mathcal{Q}^{\setminus 1}$ is also Gaussian in $\mathbf{w}$, we can use the product rule for Gaussian distributions (see Appendix C.2) to obtain the moments of $\mathbf{w}$ with respect to $t_1\mathcal{Q}^{\setminus 1}Z_1^{-1}$. The final update operation for $\mathcal{Q}$ is given by (4.27) and the logarithm of the normalization constant of $t_1\mathcal{Q}^{\setminus 1}$ is obtained as

$$\begin{aligned}\log Z_1 = &-\frac{n}{2}\log(2\pi\sigma_0^2) - \frac{1}{2}\log|\mathbf{I}+\sigma_0^{-2}\tilde{\mathbf{V}}_2\mathbf{X}^{\mathrm{T}}\mathbf{X}| + \frac{1}{2}\mathbf{m}^{\mathrm{T}}(\tilde{\mathbf{V}}_2^{-1}\tilde{\mathbf{m}}_2 + \sigma_0^{-2}\mathbf{X}^{\mathrm{T}}\mathbf{y}) \\ &-\frac{1}{2}\tilde{\mathbf{m}}_2^{\mathrm{T}}\tilde{\mathbf{V}}_2^{-1}\tilde{\mathbf{m}}_2 - \frac{1}{2}\sigma_0^{-2}\mathbf{y}^{\mathrm{T}}\mathbf{y}\,,\end{aligned} \tag{C.16}$$

where $\mathbf{m}$ is the expectation of $\mathbf{w}$ under $\mathcal{Q}$ after the update of this distribution and $\tilde{\mathbf{V}}_2$ is a $d\times d$ diagonal matrix such that $\mathrm{diag}(\tilde{\mathbf{V}}_2) = \tilde{\mathbf{v}}_2$. Once $\mathcal{Q}$ has been refined, the update rule for $\tilde{t}_1$ is computed as the ratio between $\mathcal{Q}$ and $\mathcal{Q}^{\setminus 1}$, see (4.30). Finally, the positive constant $\tilde{s}_1$ in (4.14) is fixed so that condition

$$\tilde{t}_1(\mathbf{w},\mathbf{z}) = Z_1\frac{\mathcal{Q}(\mathbf{w},\mathbf{z})}{\mathcal{Q}^{\setminus 1}(\mathbf{w},\mathbf{z})} \tag{C.17}$$

is satisfied. This equality is translated into equation (4.32) for the value of $\log\tilde{s}_1$.

C.3.2 The Second Approximate Term

To minimize $\mathrm{D}_{\mathrm{KL}}(t_2 Q^{\setminus 2} \| Q)$, we first compute $\mathscr{Q}^{\setminus 2}$, whose parameters are obtained from the ratio between Q and $\tilde{t}_2$, namely

$$\mathscr{Q}^{\setminus 2}(\mathbf{w},\mathbf{z}) = \prod_{i=1}^{d} \mathcal{N}(w_i|\tilde{m}_{1i},\tilde{v}_{1i}) \mathrm{Bern}(z_i|\sigma(\tilde{p}_{3i}))\,. \tag{C.18}$$

The divergence is minimized when $\mathscr{Q}$ is updated so that the marginal moments of $\mathbf{w}$ (first and second moment) and $\mathbf{z}$ (first moment) are the same under $\mathscr{Q}$ and $t_2\mathscr{Q}^{\setminus 2}Z_2^{-1}$, where Z_2 is the normalization constant of $t_2\mathscr{Q}^{\setminus 2}$. Hence, the update rule for the parameters of $\mathscr{Q}$ is

$$m_i^{\text{new}} = \mathbb{E}[w_i]\,, \qquad v_i^{\text{new}} = \mathbb{E}[w_i^2] - \mathbb{E}[w_i]^2\,, \qquad p_i^{\text{new}} = \sigma^{-1}(\mathbb{E}[z_i])\,, \tag{C.19}$$

where all the expectations are taken with respect to $t_2\mathscr{Q}^{\setminus 2}Z_2^{-1}$. Since $t_2\mathscr{Q}^{\setminus 2}$ can be factorized in the components of $\mathbf{w}$ and $\mathbf{z}$, Z_2 is given by the product of the normalization constants of the resulting factors, that is, $Z_2 = \prod_{i=1}^{d} n_i$, where

$$n_i = \sigma(\tilde{p}_{3i})\mathcal{N}(0|\tilde{m}_{1i},\tilde{v}_{1i}+v_s) + \sigma(-\tilde{p}_{3i})\mathcal{N}(0|\tilde{m}_{1i},\tilde{v}_{1i}) \tag{C.20}$$

and we have used the property $1-\sigma(x) = \sigma(-x)$ for any $x \in \mathbb{R}$ of the logistic function. Given n_i, we calculate the mean and variance of w_i under $t_2\mathscr{Q}^{\setminus 2}Z_2^{-1}$ very easily. For this, we need only the partial derivatives of $\log n_i$ with respect to $\tilde{m}_{1i}$ and $\tilde{v}_{1i}$ as indicated by formulas (3.18) and (3.19) in the thesis of Minka (2001). Furthermore, the expectation of z_i under $t_2\mathscr{Q}^{\setminus 2}Z_2^{-1}$ is also computed in a straightforward manner. Consequently, we obtain

$$\mathbb{E}[w_i] = \tilde{m}_{1i} + \tilde{v}_{1i}\frac{\partial \log n_i}{\partial \tilde{m}_{1i}}\,, \tag{C.21}$$

$$\mathbb{E}[w_i^2] - \mathbb{E}[w_i]^2 = \tilde{v}_{1i} - \tilde{v}_{1i}^2\left[\left(\frac{\partial \log n_i}{\partial \tilde{m}_{1i}}\right)^2 - 2\frac{\partial \log n_i}{\partial \tilde{v}_{1i}}\right]\,, \tag{C.22}$$

$$\mathbb{E}[z_i] = \sigma(\tilde{p}_{3i})\mathcal{N}(0|\tilde{m}_{1i},\tilde{v}_{1i}+v_s)n_i^{-1}\,. \tag{C.23}$$

Once $\mathscr{Q}$ has been refined, we obtain the update for $\tilde{t}_2$ by computing the ratio between $\mathscr{Q}$ and $\mathscr{Q}^{\setminus 2}$ (see Appendix C.2):

$$\tilde{v}_{2i}^{\text{new}} = \left[(v_i^{\text{new}})^{-1} - \tilde{v}_{1i}^{-1}\right]^{-1}\,, \tag{C.24}$$

$$\tilde{m}_{2i}^{\text{new}} = \tilde{v}_{2i}^{\text{new}}\left[m_i^{\text{new}}(v_i^{\text{new}})^{-1} - \tilde{m}_{1i}\tilde{v}_{1i}^{-1}\right]^{-1}\,, \tag{C.25}$$

$$\tilde{p}_{2i}^{\text{new}} = p_i^{\text{new}} - \tilde{p}_{3i}\,, \tag{C.26}$$

After some arithmetic simplifications, these formulas are translated into (4.22), (4.23) and (4.24). Finally, the positive constant $\tilde{s}_2$ in (4.15) is fixed so that condition

$$\tilde{t}_2(\mathbf{w},\mathbf{z}) = Z_2\frac{\mathscr{Q}(\mathbf{w},\mathbf{z})}{\mathscr{Q}^{\setminus 2}(\mathbf{w},\mathbf{z})} \tag{C.27}$$

is satisfied. This equality is translated into equation (4.33) for the value of $\log \tilde{s}_2$.

C.4 Constrained Minimization of the KL Divergence

When $\mathrm{D}_{\mathrm{KL}}(t_2 \mathcal{Q}^{\setminus 2} \| \tilde{t}_2 \mathcal{Q}^{\setminus 2})$ is minimized with respect to $\tilde{m}_{2i}$, $\tilde{v}_{2i}$ and $\tilde{p}_{2i}$, the optimal value for $\tilde{v}_{2i}$ can be negative. To avoid this situation, we minimize the divergence subject to constraint $\tilde{v}_{2i} \geq 0$. Two different scenarios are possible. In the first one, the optimal unconstrained value for $\tilde{v}_{2i}$ is zero or positive and condition $(a_i^2 - b_i)^{-1} \geq \tilde{v}_{1i}$ is satisfied, where a_i and b_i are given by (4.25) and (4.26). The update rules for $\tilde{m}_{2i}$, $\tilde{v}_{2i}$ and $\tilde{p}_{2i}$ are in this case the same as in the unconstrained setting, that is, (4.22), (4.23) and (4.24). In the second scenario, the optimal unconstrained value for $\tilde{v}_{2i}$ is negative and condition $(a_i^2 - b_i)^{-1} < \tilde{v}_{1i}$ is satisfied. In this case, the original update operation for $\tilde{v}_{2i}$ needs to be modified. Recall that $\mathrm{D}_{\mathrm{KL}}(t_2 \mathcal{Q}^{\setminus 2} \| \tilde{t}_2 \mathcal{Q}^{\setminus 2})$ is convex in the natural parameters $\eta_i = \tilde{m}_{2i}\tilde{v}_{2i}^{-1}$ and $\nu_i = \tilde{v}_{2i}^{-1}$ (Bishop, 2006). Under this reparameterization, constraint $\tilde{v}_{2i} \geq 0$ is translated into constraint $\nu_i \geq 0$. The optimal constrained value for ν_i must then lay on the border $\nu_i = 0$ since the optimal unconstrained value for ν_i is negative under this second scenario and the target function is convex. The resulting update rule for $\tilde{v}_{2i}$ is thus given by $\tilde{v}_{2i} = \infty$. Additionally, the update rule for $\tilde{p}_{2i}$ is still given by (4.24) because the optimal value for $\tilde{p}_{2i}$ does not depend on $\tilde{m}_{2i}$ or $\tilde{v}_{2i}$. Finally, the optimal value for $\tilde{m}_{2i}$ in the second scenario is again given by (4.23) since this formula yields the minimizer of $\mathrm{D}_{\mathrm{KL}}(t_2 \mathcal{Q}^{\setminus 2} \| \tilde{t}_2 \mathcal{Q}^{\setminus 2})$ with respect to $\tilde{m}_{2i}$ when conditioning to the value selected for $\tilde{v}_{2i}$.

Appendix D

Appendix for Chapter 5

D.1 The EP Update Operations for NBSBC

This section describes the EP update operations for refining the $\tilde{t}_i$ terms whose product approximates the joint distribution $\mathcal{P}(\mathbf{w},\varepsilon,\mathbf{z},\mathbf{y}|\mathbf{X},G,\alpha,\beta)$ in the NBSBC model. Recall that $\mathcal{Q}$ and the $\tilde{t}_i$ have the form given by (5.14) and (5.15), respectively. The update equations for a given $\tilde{t}_i$ are obtained from the minimization of the Kullback-Leibler (KL) divergence between $\tilde{t}_i\mathcal{Q}^{\setminus i}$ and $t_i\mathcal{Q}^{\setminus i}$. The solution to this optimization problem is obtained when all the following expectation constraints are satisfied

$$
\begin{aligned}
\mathbb{E}_{\tilde{t}_i\mathcal{Q}^{\setminus i}}[\mathbf{w}] &= \mathbb{E}_{t_i\mathcal{Q}^{\setminus i}}[\mathbf{w}]\,, && \text{(D.1)}\\
\mathbb{E}_{\tilde{t}_i\mathcal{Q}^{\setminus i}}[\mathbf{w}\circ\mathbf{w}] &= \mathbb{E}_{t_i\mathcal{Q}^{\setminus i}}[\mathbf{w}\circ\mathbf{w}]\,, && \text{(D.2)}\\
\mathbb{E}_{\tilde{t}_i\mathcal{Q}^{\setminus i}}[\mathbf{z}] &= \mathbb{E}_{t_i\mathcal{Q}^{\setminus i}}[\mathbf{z}]\,, && \text{(D.3)}\\
\mathbb{E}_{\tilde{t}_i\mathcal{Q}^{\setminus i}}[\log(\varepsilon)] &= \mathbb{E}_{t_i\mathcal{Q}^{\setminus i}}[\log(\varepsilon)]\,, && \text{(D.4)}\\
\mathbb{E}_{\tilde{t}_i\mathcal{Q}^{\setminus i}}[\log(1-\varepsilon)] &= \mathbb{E}_{t_i\mathcal{Q}^{\setminus i}}[\log(1-\varepsilon)]\,, && \text{(D.5)}
\end{aligned}
$$

where the operator $\circ$ denotes the Hadamard element-wise product. Note that $\mathcal{Q}$ is always equal to the product of $\mathcal{Q}^{\setminus i}$ and $\tilde{t}_i$. Therefore, because it is computationally more efficient, the update of each $\tilde{t}_i$ is performed first, by finding the $\mathcal{Q}$ that satisfies the previous constraints and second, by computing the ratio between $\mathcal{Q}$ and $\mathcal{Q}^{\setminus i}$, that is, $\tilde{t}_i = \mathcal{Q}/\mathcal{Q}^{\setminus i}$. These operations are typically implemented using the normalized versions of $\mathcal{Q}$ and $\mathcal{Q}^{\setminus i}$, that is, $\mathcal{Q}$ and $\mathcal{Q}^{\setminus i}$, respectively.

The first step before refining any $\tilde{t}_i$ is to compute $\mathcal{Q}^{\setminus i}$. Note that $\mathcal{Q}$ and all the $\tilde{t}_i$ are in the same family of exponential distributions. Therefore, $\mathcal{Q}^{\setminus i}$ has the same analytical form as $\mathcal{Q}$

$$
\mathcal{Q}^{\setminus i}(\mathbf{w},\varepsilon,\mathbf{z}) = \text{Beta}(\varepsilon|a^{\setminus i},b^{\setminus i})\prod_{j=0}^{d}\mathcal{N}(w_j|m_j^{\setminus i},v_j^{\setminus i})\text{Bern}(z_j|p_j^{\setminus i})\,, \qquad \text{(D.6)}
$$

where $\mathbf{m}^{\setminus i} = (m_0^{\setminus i}, \ldots, m_d^{\setminus i})^{\mathrm{T}}$, $\mathbf{v}^{\setminus i} = (v_0^{\setminus i}, \ldots, v_d^{\setminus i})^{\mathrm{T}}$, $\mathbf{p}^{\setminus i} = (p_0^{\setminus i}, \ldots, p_d^{\setminus i})^{\mathrm{T}}$, $a^{\setminus i}$ and $b^{\setminus i}$ are determined by computing the ratio between $\mathcal{Q}$ and $\tilde{t}_i$ and normalizing,

$$\mathbf{v}^{\setminus i} = (\mathbf{v}^{-1} - \tilde{\mathbf{v}}_i^{-1})^{-1}\,, \tag{D.7}$$

$$\mathbf{m}^{\setminus i} = \mathbf{m} + \mathbf{v}^{\setminus i} \circ \tilde{\mathbf{v}}_i^{-1} \circ (\mathbf{m} - \tilde{\mathbf{m}}_i)\,, \tag{D.8}$$

$$\mathbf{p}^{\setminus i} = \mathbf{p} \circ \tilde{\mathbf{c}}_i^{-1} \circ (\mathbf{p} \circ \tilde{\mathbf{c}}_i^{-1} + (1 - \mathbf{p}) \circ \tilde{\mathbf{d}}_i^{-1})^{-1}\,, \tag{D.9}$$

$$a^{\setminus i} = a - \tilde{a}_i\,, \tag{D.10}$$

$$b^{\setminus i} = b - \tilde{b}_i\,. \tag{D.11}$$

Here, the inverse of a vector is defined as the vector whose components are the inverse components of the original vector. Once $\mathcal{Q}^{\setminus i}$ is available, EP proceeds by updating $\mathcal{Q}$ so that the expectations of $\mathbf{w}$, $\mathbf{w} \circ \mathbf{w}$, $\mathbf{z}$, $\log(\varepsilon)$ and $1 - \log(\varepsilon)$ with respect to $\mathcal{Q}$ and $Z_i^{-1} t_i \mathcal{Q}^{\setminus i}$ are equal, where Z_i is the normalization constant of $t_i \mathcal{Q}^{\setminus i}$. These update equations for $\mathcal{Q}$ depend on the particular $\tilde{t}_i$ that is being processed. Once $\mathcal{Q}$ is updated, the new $\tilde{t}_i$ is fixed to be equal to $Z_i \mathcal{Q}/\mathcal{Q}^{\setminus i}$. We now describe for each type of approximate term the corresponding update rule for $\mathcal{Q}$.

When EP refines the approximate terms for the likelihood, that is, $\tilde{t}_i$ for $i = 1, \ldots, n$, the update equations for $\mathcal{Q}$ are given by the constraints in the expectations of $\mathbf{w}$, $\mathbf{w} \circ \mathbf{w}$, $\log(\varepsilon)$ and $1 - \log(\varepsilon)$. These expectations must be equal under $\mathcal{Q}$ and under $Z_i^{-1} t_i \mathcal{Q}^{\setminus i}$. The expectations of $\mathbf{w}$ and $\mathbf{w} \circ \mathbf{w}$ yield the following update rules for $\mathbf{m}$ and $\mathbf{v}$,

$$\mathbf{m}^{\text{new}} = \mathbf{m}^{\setminus i} + y_i \alpha_i \mathbf{v}^{\setminus i} \circ \mathbf{x}_i\,, \tag{D.12}$$

$$\mathbf{v}^{\text{new}} = \mathbf{v}^{\setminus i} - \frac{y_i \alpha_i \mathbf{x}_i^{\mathrm{T}} \mathbf{m}^{\text{new}}}{\mathbf{x}_i^{\mathrm{T}} (\mathbf{v}^{\setminus i} \circ \mathbf{x}_i)} (\mathbf{v}^{\setminus i} \circ \mathbf{x}_i) \circ (\mathbf{v}^{\setminus i} \circ \mathbf{x}_i)\,, \tag{D.13}$$

where α_i is given by

$$\alpha_i = \frac{(1 - 2\bar{\varepsilon}^{\setminus i}) \mathcal{N}(\beta_i | 0, 1)}{\bar{\varepsilon}^{\setminus i} + (1 - 2\bar{\varepsilon}^{\setminus i}) \Phi(\beta_i)} \left[\mathbf{x}_i^{\mathrm{T}} (\mathbf{v}^{\setminus i} \circ \mathbf{x}_i) \right]^{-1/2}, \tag{D.14}$$

$$\beta_i = y_i \mathbf{x}_i^{\mathrm{T}} \mathbf{m}^{\setminus i} \left[\mathbf{x}_i^{\mathrm{T}} (\mathbf{v}^{\setminus i} \circ \mathbf{x}_i) \right]^{-1/2}, \tag{D.15}$$

$$\bar{\varepsilon}^{\setminus i} = a^{\setminus i} / (a^{\setminus i} + b^{\setminus i})\,, \tag{D.16}$$

Φ is the standard Gaussian cumulative distribution function and $\mathcal{N}(\cdot|0,1)$ is the standard Gaussian density function. Note that, on very rare occasions, this update rule may fail because of a negative value in some component of $\mathbf{v}^{\setminus i}$. To deal with it, the i-th approximate term $(1 < i < n)$ is simply ignored until the next iteration of the EP algorithm whenever some component of $\mathbf{v}^{\setminus i}$ is negative (Minka, 2001). The expectations of $\log(\varepsilon)$ and $1 - \log(\varepsilon)$ yield the the following update rules for a and b,

$$\psi(a^{\text{new}}) - \psi(a^{\text{new}} + b^{\text{new}}) = \frac{\bar{\varepsilon}^{\setminus i}(1 - \Phi(\beta_i))}{a^{\setminus i} \left[\bar{\varepsilon}^{\setminus i} + (1 - 2\bar{\varepsilon}^{\setminus i}) \Phi(\beta_i) \right]} + \psi(a^{\setminus i}) - \psi(a^{\setminus i} + b^{\setminus i} + 1)\,, \tag{D.17}$$

$$\psi(b^{\text{new}}) - \psi(a^{\text{new}} + b^{\text{new}}) = \frac{(1 - \bar{\varepsilon}^{\setminus i}) \Phi(\beta_i)}{b^{\setminus i} \left[\bar{\varepsilon}^{\setminus i} + (1 - 2\bar{\varepsilon}^{\setminus i}) \Phi(\beta_i) \right]} + \psi(b^{\setminus i}) - \psi(a^{\setminus i} + b^{\setminus i} + 1)\,, \tag{D.18}$$

where ψ is the digamma function. However, because ψ is non-linear, there is not an explicit formula for a^{new} and b^{new} that can be derived from (D.17) and (D.18). Thus, instead of propagating the expectations of $\log(\varepsilon)$ and $\log(1-\varepsilon)$ we follow Cowell et al. (1996) and choose to propagate the expectations of ε and ε^2 instead. Although we are no longer minimizing the Kullback-Leibler divergence, the resulting approximation is still very accurate. This leads to the following update equations for a and b,

$$a^{\text{new}} = \frac{\mathbb{E}_{Z_i^{-1}t_i\mathcal{Q}^{\setminus i}}[\varepsilon] - \mathbb{E}_{Z_i^{-1}t_i\mathcal{Q}^{\setminus i}}[\varepsilon^2]}{\mathbb{E}_{Z_i^{-1}t_i\mathcal{Q}^{\setminus i}}[\varepsilon^2] - \mathbb{E}_{Z_i^{-1}t_i\mathcal{Q}^{\setminus i}}[\varepsilon]^2}\mathbb{E}_{Z_i^{-1}t_i\mathcal{Q}^{\setminus i}}[\varepsilon]\,, \tag{D.19}$$

$$b^{\text{new}} = \frac{\mathbb{E}_{Z_i^{-1}t_i\mathcal{Q}^{\setminus i}}[\varepsilon] - \mathbb{E}_{Z_i^{-1}t_i\mathcal{Q}^{\setminus i}}[\varepsilon^2]}{\mathbb{E}_{Z_i^{-1}t_i\mathcal{Q}^{\setminus i}}[\varepsilon^2] - \mathbb{E}_{Z_i^{-1}t_i\mathcal{Q}^{\setminus i}}[\varepsilon]^2}(1 - \mathbb{E}_{Z_i^{-1}t_i\mathcal{Q}^{\setminus i}}[\varepsilon])\,, \tag{D.20}$$

where $\mathbb{E}_{Z_i^{-1}t_i\mathcal{Q}^{\setminus i}}[\varepsilon]$ and $\mathbb{E}_{Z_i^{-1}t_i\mathcal{Q}^{\setminus i}}[\varepsilon^2]$ are given by

$$\mathbb{E}_{Z_i^{-1}t_iQ^{\setminus i}}[\varepsilon] = \frac{1}{Z_i(a^{\setminus i}+b^{\setminus i}+1)}\left[\Phi(\beta_i)(1-\bar{\varepsilon}^{\setminus i})a^{\setminus i} + (1-\Phi(\beta_i))\bar{\varepsilon}^{\setminus i}(a^{\setminus i}+1)\right]\,, \tag{D.21}$$

$$\begin{aligned}\mathbb{E}_{Z_i^{-1}t_iQ^{\setminus i}}[\varepsilon^2] = &\frac{a^{\setminus i}+1}{Z_i(a^{\setminus i}+b^{\setminus i}+1)(a^{\setminus i}+b^{\setminus i}+2)}\cdot\\ &\left[\Phi(\beta_i)(1-\bar{\varepsilon}^{\setminus i})a^{\setminus i} + (1-\Phi(\beta_i))\bar{\varepsilon}^{\setminus i}(a^{\setminus i}+2)\right]\end{aligned} \tag{D.22}$$

and $Z_i = \bar{\varepsilon}^{\setminus i} + \left(1 - 2\bar{\varepsilon}^{\setminus i}\right)\Phi(\beta_i)$.

When EP refines the approximate term for the sparse prior (5.7), that is, $\tilde{t}_i$ for $i = n+1$, the update rules for $\mathcal{Q}$ are given by the constraints in the expectations of $\mathbf{w}$, $\mathbf{w}\circ\mathbf{w}$ and $\mathbf{z}$. These expectations must be equal under $\mathcal{Q}$ and under $Z_i^{-1}t_i\mathcal{Q}^{\setminus i}$. The update equations for $\mathbf{m}$, $\mathbf{v}$ and $\mathbf{p}$ are

$$\mathbf{m}^{\text{new}} = \mathbf{m}^{\setminus i} + \mathbf{k}'\circ\mathbf{v}^{\setminus i}\,, \tag{D.23}$$

$$\mathbf{v}^{\text{new}} = \mathbf{v}^{\setminus i} - \mathbf{k}'''\circ\mathbf{v}^{\setminus i}\circ\mathbf{v}^{\setminus i}\,, \tag{D.24}$$

$$\mathbf{p}^{\text{new}} = \mathbf{p}^{\setminus i}\circ\mathbf{g}''\circ(\mathbf{p}^{\setminus i}\circ\mathbf{g}'' + (1-\mathbf{p}^{\setminus i})\circ\mathbf{g}''')^{-1}\,, \tag{D.25}$$

where $\mathbf{k}'$, $\mathbf{k}'''$, $\mathbf{g}''$ and $\mathbf{g}'''$ are given by

$$\mathbf{g}' = \mathbf{p}^{\setminus i}\circ\mathbf{g}'' + (1-\mathbf{p}^{\setminus i})\circ\mathbf{g}'''\,, \tag{D.26}$$

$$\mathbf{g}'' = \mathcal{N}(0|\mathbf{m}^{\setminus i}, \mathbf{v}^{\setminus i} + \boldsymbol{\sigma^2})\,, \tag{D.27}$$

$$\mathbf{g}''' = \mathcal{N}(0|\mathbf{m}^{\setminus i}, \mathbf{v}^{\setminus i})\,, \tag{D.28}$$

$$\mathbf{k}' = -\frac{\mathbf{p}^{\setminus i}\circ\mathbf{g}''\circ\mathbf{m}^{\setminus i}}{\mathbf{g}'\circ(\mathbf{v}^{\setminus i}+\boldsymbol{\sigma^2})} - \frac{(1-\mathbf{p}^{\setminus i})\circ\mathbf{g}'''\circ\mathbf{m}^{\setminus i}}{\mathbf{g}'\circ\mathbf{v}^{\setminus i}}\,, \tag{D.29}$$

$$\begin{aligned}\mathbf{k}'' = &\frac{\mathbf{p}^{\setminus i}\circ\mathbf{g}''\circ\mathbf{m}^{\setminus i}\circ\mathbf{m}^{\setminus i}}{\mathbf{g}'\circ(\mathbf{v}^{\setminus i}+\boldsymbol{\sigma^2})\circ(\mathbf{v}^{\setminus i}+\boldsymbol{\sigma^2})} - \frac{\mathbf{p}^{\setminus i}\circ\mathbf{g}''}{\mathbf{g}'\circ(\mathbf{v}^{\setminus i}+\boldsymbol{\sigma^2})} +\\ &\frac{(1-\mathbf{p}^{\setminus i})\circ\mathbf{g}'''\circ\mathbf{m}^{\setminus i}\circ\mathbf{m}^{\setminus i}}{\mathbf{g}'\circ\mathbf{v}^{\setminus i}\circ\mathbf{v}^{\setminus i}} - \frac{(1-\mathbf{p}^{\setminus i})\circ\mathbf{g}'''}{\mathbf{g}'\circ\mathbf{v}^{\setminus i}}\,,\end{aligned} \tag{D.30}$$

$$\mathbf{k}''' = \mathbf{k}'\circ\mathbf{k}' - \mathbf{k}'' \tag{D.31}$$

and $\boldsymbol{\sigma}^2$ is a d-dimensional vector whose zeroth component is equal to 100 and whose components $1,\ldots,d$ are equal to 1. Note that in this case $Z_i = \prod_{j=0}^{d} g'_j$.

When EP refines the approximate term for the first part of the MRF prior (5.8), that is, $\tilde{t}_i$ for $i = n+2$, the update equations for $\mathcal{Q}$ are given by the constraints in the expectation of $\mathbf{z}$. This expectation must be equal under $\mathcal{Q}$ and under $Z_i^{-1} t_i \mathcal{Q}^{\setminus i}$. The resulting update rule for $\mathbf{p}$ is

$$\mathbf{p}^{\text{new}} = \exp(\mathbf{h}) \circ \mathbf{p}^{\setminus i} \circ (\exp(\mathbf{h}) \circ \mathbf{p}^{\setminus i} + \exp(-\mathbf{h}) \circ (1 - \mathbf{p}^{\setminus i}))^{-1}, \tag{D.32}$$

where $\mathbf{h} = (h_0,\ldots,h_d)^{\text{T}}$ is a vector of dimension $d+1$ whose zeroth component is 10 and whose last d components are equal to α. Finally, $Z_i = \prod_{j=0}^{d} [p_j^{\setminus i} \exp(h_j) + (1 - p_j^{\setminus i}) \exp(-h_j)]$.

When EP refines the approximate terms for the second part of the MRF prior (5.8), that is, $\tilde{t}_i$ for $i = n+3,\ldots,n+2+|E|$, the update equations for $\mathcal{Q}$ are given by the constraints in the expectation of $\mathbf{z}$. This expectation must be equal under $\mathcal{Q}$ and under $Z_i^{-1} t_i \mathcal{Q}^{\setminus i}$. The corresponding update rules for the parameters of $\mathcal{Q}$ are

$$p_j^{\text{new}} = \frac{A_i + D_i}{A_i + B_i + C_i + D_i}, \qquad p_k^{\text{new}} = \frac{A_i + C_i}{A_i + B_i + C_i + D_i}, \tag{D.33}$$

where j and k are the features linked by the edge corresponding the t_i that is being approximated and A_i, B_i, C_i and D_i are given by

$$A_i = p_j^{\setminus i} p_k^{\setminus i} \exp(\beta), \qquad B_i = (1 - p_j^{\setminus i})(1 - p_k^{\setminus i}) \exp(\beta), \tag{D.34}$$

$$C_i = (1 - p_j^{\setminus i}) p_k^{\setminus i} \exp(-\beta), \qquad D_i = p_j^{\setminus i}(1 - p_k^{\setminus i}) \exp(-\beta). \tag{D.35}$$

Note that in this case $Z_i = A_i + B_i + C_i + D_i$.

Finally, when EP refines the approximate term for the noise prior (5.9), that is, $\tilde{t}_i$ where $i = n + |E| + 3$, the update equations for $\mathcal{Q}$ are given by the constraints in the expectations of $\log(\varepsilon)$ and $1 - \log(\varepsilon)$. These expectations must be equal under $\mathcal{Q}$ and under $Z_i^{-1} t_i \mathcal{Q}^{\setminus i}$. The corresponding update rules for a and b are

$$a^{\text{new}} = a_0 + a^{\setminus i} - 1, \qquad b^{\text{new}} = b_0 + b^{\setminus i} - 1 \tag{D.36}$$

and in this case $Z_i = \text{B}(a,b)\text{B}(a_0,b_0)^{-1}\text{B}(a^{\setminus i}, b^{\setminus i})^{-1}$.

Once $\mathcal{Q}$ is modified, EP updates the proximate term $\tilde{t}_i$ that is being refined. This is achieved by setting $\tilde{t}_i$ equal to $Z_i \mathcal{Q}/\mathcal{Q}^{\setminus i}$. Hence, the parameters of this approximate term are updated as

$$\tilde{\mathbf{v}}_i^{\text{new}} = (\mathbf{v}^{-1} - (\mathbf{v}^{\setminus i})^{-1})^{-1}, \tag{D.37}$$

$$\tilde{\mathbf{m}}_i^{\text{new}} = \tilde{\mathbf{v}}_i^{\text{new}} \circ \mathbf{v}^{-1} \circ \mathbf{m} - \tilde{\mathbf{v}}_i^{\text{new}} \circ (\mathbf{v}^{\setminus i})^{-1} \circ \mathbf{m}^{\setminus i}, \tag{D.38}$$

$$\tilde{\mathbf{c}}_i^{\text{new}} = \mathbf{p} \circ (\mathbf{p}^{\setminus i})^{-1}, \tag{D.39}$$

$$\tilde{\mathbf{d}}_i^{\text{new}} = (1 - \mathbf{p}) \circ (1 - \mathbf{p}^{\setminus i})^{-1}, \tag{D.40}$$

$$\tilde{a}_i^{\text{new}} = a - a^{\setminus i}, \tag{D.41}$$

$$\tilde{b}_i^{\text{new}} = b - b^{\setminus i} \tag{D.42}$$

and the update for $\tilde{s}_i$ depends on the particular $\tilde{t}_i$ that is being refined. In particular,

$$\tilde{s}_i^{\text{new}} = Z_i \sqrt{\prod_{j=0}^{d} \frac{\tilde{v}_{ij}^{\text{new}} + v_j^{\setminus i}}{\tilde{v}_{ij}^{\text{new}}}} \exp\left\{\frac{1}{2}\sum_{j=0}^{d} \frac{(\tilde{m}_{ij}^{\text{new}} - m_j^{\setminus i})^2}{(\tilde{v}_{ij}^{\text{new}} + v_j^{\setminus i})}\right\} \frac{\mathrm{B}(a^{\setminus i}, b^{\setminus i})}{\mathrm{B}(a,b)}\,, \tag{D.43}$$

for $i = 1,\ldots,n$ and

$$\tilde{s}_i^{\text{new}} = Z_i \prod_{j=0}^{d} \sqrt{\frac{v_j^{\setminus i} + \tilde{v}_j^{\text{new}}}{\tilde{v}_j^{\text{new}}}} \exp\left\{\frac{1}{2}\frac{\left(k'_j\right)^2}{k'''_j}\right\}, \tag{D.44}$$

when $i = n+1$. The corresponding update for $i = n+2,\ldots,n+|E|+2$ is given by $\tilde{s}_i^{\text{new}} = Z_i$ and finally, $\tilde{s}_i^{\text{new}} = \mathrm{B}(a_0,b_0)^{-1}$ when $i = n+|E|+3$.

Once EP has converged, we can approximate the model evidence, $\mathcal{P}(\mathbf{y}|\mathbf{X},G,\alpha,\beta)$, by the normalization constant of $\mathcal{Q}$,

$$\mathcal{P}(\mathbf{y}|\mathbf{X},G,\alpha,\beta) \approx \tilde{Z}^{-1} C\,(2\pi)^{d/2} \exp\left(D/2\right) \mathrm{B}(A,B) \left[\prod_{i=1}^{n+3+|E|} \tilde{s}_i\right] \left[\prod_{j=0}^{d} \sqrt{v_j}\right], \tag{D.45}$$

where A, B, C, and D are given by

$$A = \sum_{i=1}^{n+3+|E|} \tilde{a}_i + 1\,, \tag{D.46}$$

$$B = \sum_{i=1}^{n+3+|E|} \tilde{b}_i + 1\,, \tag{D.47}$$

$$C = \prod_{j=0}^{d} \left\{ \prod_{i=1}^{n+3+|E|} \tilde{c}_{ij} + \prod_{i=1}^{n+3+|E|} \tilde{d}_{ij} \right\}, \tag{D.48}$$

$$D = \mathbf{m}^{\mathrm{T}}(\mathbf{v}^{-1} \circ \mathbf{m}) - \sum_{i=1}^{n+3+|E|} \tilde{\mathbf{m}}_i^{\mathrm{T}} (\tilde{\mathbf{v}}_i^{-1} \circ \tilde{\mathbf{m}}_i) \tag{D.49}$$

and $\tilde{Z}$ is an estimate of the normalization constant Z that appears in the MRF prior (5.8). This estimate is obtained by running the EP algorithm only on the term corresponding to the Markov Random Field.

Appendix E

Appendix for Chapter 6

E.1 Approximations for Computing the EP Update Operations

Let

$$\begin{aligned}\mathcal{Q}^{\setminus it}(\mathbf{W},\mathbf{Z},\mathbf{r},\boldsymbol{\sigma}^2) = &\left[\prod_{k=1}^{d}\prod_{j=1}^{d}\mathcal{N}(w_{kj}|m_{kj}^{\setminus it},v_{kj}^{\setminus it})\right]\left[\prod_{k=1}^{d}\prod_{j=1}^{d}\mathrm{Bern}(z_{kj}|p_{kj}^{\setminus it})\right]\\ &\left[\prod_{k=1}^{d}\mathrm{Bern}(r_k|q_k^{\setminus it})\right]\left[\prod_{k=1}^{d}\mathrm{IG}(\sigma_k^2|a_k^{\setminus it},b_k^{\setminus it})\right]\end{aligned} \tag{E.1}$$

be the distribution obtained from the ratio between $\mathcal{Q}(\mathbf{W},\mathbf{Z},\mathbf{r},\boldsymbol{\sigma}^2)$, that is, the posterior approximation (6.8), and the approximate term corresponding to the factor $\mathcal{N}(x_{it}|\mathbf{w}_i\mathbf{x}_{it-1},\sigma_i^2)$ in (6.2), where $m_{kj}^{\setminus it}$, $v_{kj}^{\setminus it}$, $p_{kj}^{\setminus it}$, $q_k^{\setminus it}$, $a_k^{\setminus it}$ and $b_k^{\setminus it}$ are parameters obtained following rules similar to those described in Appendix C.2. For obtaining the EP update, we have to compute the expectations of $\mathbf{w}_i$, $\mathbf{w}_i \circ \mathbf{w}_i$, σ_i^2 and σ_i^4 with respect to $Z_{it}(a_i^{\setminus it},b_i^{\setminus it})^{-1}\mathcal{Q}^{\setminus it}(\mathbf{W},\mathbf{Z},\mathbf{r},\boldsymbol{\sigma}^2)\mathcal{N}(x_{it}|\mathbf{w}_i\mathbf{x}_{it-1},\sigma_i^2)$, where "$\circ$" denotes the Hadamard element-wise product between vectors of the same dimension and $Z_{it}(a_i^{\setminus it},b_i^{\setminus it})$ is a normalization constant given by

$$\begin{aligned}Z_{it}(a_i^{\setminus it},b_i^{\setminus it}) &= \sum_{\mathbf{Z},\mathbf{r}}\int\int\mathcal{Q}^{\setminus it}(\mathbf{W},\mathbf{Z},\mathbf{r},\boldsymbol{\sigma}^2)\mathcal{N}(x_{it}|\mathbf{w}_i\mathbf{x}_{it-1},\sigma_i^2)\,d\mathbf{W}\,d\boldsymbol{\sigma}\\ &= \int\mathcal{N}(x_{it}|\mathbf{m}_i^{\setminus it}\mathbf{x}_{it-1},\sigma_i^2+\mathbf{v}_i^{\setminus it}(\mathbf{x}_{it-1}\circ\mathbf{x}_{it-1}))\mathrm{IG}(\sigma_i^2|a_i^{\setminus it},b_i^{\setminus it})\,d\sigma_i\\ &= \int\int\mathcal{N}(y|x_{it}-\mathbf{m}_i^{\setminus it}\mathbf{x}_{it-1},\mathbf{v}_i^{\setminus it}(\mathbf{x}_{it-1}\circ\mathbf{x}_{it-1}))\mathcal{N}(y|0,\sigma_i^2)\mathrm{IG}(\sigma_i^2|a_i^{\setminus it},b_i^{\setminus it})\,d\sigma_i\,dy\\ &= \int\mathcal{N}(y|x_{it}-\mathbf{m}_i^{\setminus it}\mathbf{x}_{it-1},\mathbf{v}_i^{\setminus it}(\mathbf{x}_{it-1}\circ\mathbf{x}_{it-1}))\mathcal{T}(y|0,b_i^{\setminus it}/a_i^{\setminus it},2a_i^{\setminus it})\,dy\,,\\ &\simeq \int\mathcal{N}(y|x_{it}-\mathbf{m}_i^{\setminus it}\mathbf{x}_{it-1},\mathbf{v}_i^{\setminus it}(\mathbf{x}_{it-1}\circ\mathbf{x}_{it-1}))\mathcal{N}(y|0,2b_i^{\setminus it}/(2a_i^{\setminus it}-2))\,dy\,,\\ &= \mathcal{N}(0|x_{it}-\mathbf{m}_i^{\setminus it}\mathbf{x}_{it-1},\mathbf{v}_i^{\setminus it}(\mathbf{x}_{it-1}\circ\mathbf{x}_{it-1})+2b_i^{\setminus it}/(2a_i^{\setminus it}-2))\,,\end{aligned} \tag{E.2}$$

$\mathbf{m}_i^{\setminus it} = (m_{1i}^{\setminus it}, \ldots, m_{di}^{\setminus it})^{\mathrm{T}}$, $\mathbf{v}_i^{\setminus it} = (v_{1i}^{\setminus it}, \ldots, v_{di}^{\setminus it})^{\mathrm{T}}$,

$$\mathcal{T}(x, \mu, \lambda, \nu) = \frac{\Gamma((\nu+1)/2)}{\sqrt{\pi\nu\lambda}\Gamma(\mu/2)} \left(1 + \frac{(x-\mu)^2}{\nu\lambda}\right)^{(1-\nu)/2} \tag{E.3}$$

denotes a Student's t density with mean μ, variance parameter λ and degrees of freedom ν, and in the fifth line of (E.2) the Student's t density has been approximated by a Gaussian density with the same mean and the same variance. Since $\mathrm{IG}(x|\alpha,\beta)x = \mathrm{IG}(x|\alpha+1,\beta)\alpha\beta^{-1}$, we have that the required expectation of σ_i^2 is given by $Z_{it}(a_i^{\setminus it}, b_i^{\setminus it})^{-1} Z_{it}(a_i^{\setminus it}+1, b_i^{\setminus it}) a_i^{\setminus it}/b_i^{\setminus it}$, which can be approximated using (E.2). The expectation of σ_i^4 is calculated in a similar manner. Finally, for computing the expectations of $\mathbf{w}_i$ and $\mathbf{w}_i \circ \mathbf{w}_i$, we only need to calculate the partial derivatives of the logarithm of $Z_{it}(a_i^{\setminus it}, b_i^{\setminus it})$ with respect to $m_{ij}^{\setminus it}$ and $v_{ij}^{\setminus it}$ for $j = 1, \ldots, d$, as indicated by formulas (3.18) and (3.19) in the thesis of Minka (2001). For this, we also use the approximation described above.

Bibliography

Aas, K., Czado, C., Frigessi, A., and Bakken, H. (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44(2):182–198.

Abramson, I. S. (1982). On bandwidth variation in kernel estimates-a square root law. *The Annals of Statistics*, 10(4):1217–1223.

Alon, U. (2006). *An Introduction to Systems Biology*. CRC Press.

Alpaydin, E. (2004). *Introduction to Machine Learning*. The MIT Press.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Labys, P. (2003). Modelling and forecasting realized volatility. *Econometrica*, 71(2):579–625.

Aparicio, O., Geisberg, J. V., and Struhl, K. (2001). *Chromatin Immunoprecipitation for Determining the Association of Proteins with Specific Genomic Sequences In Vivo*. John Wiley & Sons, Inc.

Attias, H. (1999). Inferring parameters and structure of latent variable models by variational Bayes. In Laskey, K. B. and Prade, H., editors, *UAI*, pages 21–30. Morgan Kaufmann.

Bachelier, L. (1900). Théorie de la spéculation. *Annales de l'École normale supérieure*, 3(17):21–86.

Bahl, A., Brunk, B., Crabtree, J., Fraunholz, M. J., Gajria, B., Grant, G. R., Ginsburg, H., Gupta, D., Kissinger, J. C., Labo, P., et al. (2003). PlasmoDB: the plasmodium genome resource. A database integrating experimental and computational data. *Nucleic Acids Research*, 31(1):212.

Balaji, S., Babu, M. M., Iyer, L. M., and Aravind, L. (2005). Discovery of the principal specific transcription factors of Apicomplexa and their implication for the evolution of the AP2-integrase DNA binding domains. *Nucleic Acids Research*, 33(13):3994–4006.

Balcázar, J. L., Bonchi, F., Gionis, A., and Sebag, M., editors (2010). *ECML-PKDD 2010*, volume 6321 of *Lecture Notes in Artificial Intelligence*. Springer.

Barabási, A. L. and Oltvai, Z. N. (2004). Network biology: Understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113.

Barndorff-Nielsen, O. E. (1997). Normal inverse Gaussian distributions and stochastic volatility modelling. *Scandinavian Journal of Statistics*, 24(1):1–13.

Beal, M. J. (2003). *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University College London.

Berkowitz, J. (2001). Testing density forecasts, with applications to risk management. *The Journal of Business and Economic Statistics*, 19(4):465–474.

Bingham, N. H., Goldie, C. M., and Teugels, J. L. (1987). *Regular Variation*. Cambridge University Press.

Bishop, C. M. (1996). *Neural Networks for Pattern Recognition*. Oxford University Press.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Blei, D. M. and Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–144.

Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the ACL*, pages 440–447.

Bolancé, C., Guillén, M., and Nielsen, J. P. (2008). Inverse beta transformation in kernel density estimation. *Statistics & Probability Letters*, 78(13):1757 – 1764.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327.

Bollerslev, T. (1987). A conditionally heteroskedastic time series model for speculative prices and rates of return. *The Review of Economics and Statistics*, 69(3):542–547.

Bollerslev, T. and Wooldbridge, J. M. (1992). Quasi-maximum likelihood estimation and inference in dynamic models with time-varying covariances. *Econometric Reviews*, 11(2):143–172.

Bos, P. D., Zhang, X. H. F., Nadal, C., Shu, W., Gomis, R. R., Nguyen, D. X., Minn, A. J., van de Vijver, M. J., Gerald, W. L., Foekens, J. A., and Massague, J. (2009). Genes that mediate breast cancer metastasis to the brain. *Nature*, 459(7249):1005–1009.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and Regression Trees*. Chapman and Hall/CRC.

Brockwell, J. P. and Davis, R. A. (1996). *Introduction to Time Series and Forecasting*. Springer.

Buchan, I., Winn, J., and Bishop, C. (2009). A unified modeling approach to data-intensive healthcare. In Hey, A. J. G. and Hey, T., editors, *The Fourth Paradigm: Data-Intensive Scientific Discovery*, pages 91–97. Microsoft Pr, 2009.

Buntine, W. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2:159–225.

Bush, C. A. and MacEachern, S. N. (1996). A semiparametric Bayesian model for randomised block designs. *Biometrika*, 83(2):275–285.

Candès, E. (2006). Compressive sampling. In *Proceedings of the International Congress of Mathematicians*, volume 3, pages 1433–1452.

Caruana, R. (1997). Multitask learning. *Machine Learning*, 28(1):41–75.

Chen, X. and Fan, Y. (2006). Estimation and model selection of semiparametric copula-based multivariate dynamic models under copula misspecification. *Journal of Econometrics*, 135(1-2):125–154.

Cherkassky, V. S. and Mulier, F. (1998). *Learning from Data: Concepts, Theory, and Methods*. John Wiley & Sons, Inc.

Clark, P. K. (1973). A subordinated stochastic process model with finite variance for speculative prices. *Econometrica*, 41(1):135–155.

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research (JAIR)*, 4:129–145.

Cont, R. (2001). Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance*, 1(2):223–236.

Coulson, R. M. R., Hall, N., and Ouzounis, C. A. (2004). Comparative genomics of transcriptional control in the human malaria parasite *Plasmodium falciparum*. *Genome Research*, 14(8):1548–1554.

Cowell, R. G., Dawid, A. P., and Sebastiani, P. (1996). A comparison of sequential learning methods for incomplete data. In *Bayesian Statistics 5*, pages 553–541. Oxford University Press.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, New York, NY, USA. ACM.

de Boor, C. (1978). *A Practical Guide to Splines*. Number 27 in Applied Mathematical Sciences Series. Springer.

Demarta, S. and McNeil, A. J. (2005). The *t* copula and related copulas. *International Statistical Review*, 73(1):111–129.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.

Dimitrova, D. S., Kaishev, V. K., and Penev, S. I. (2008). GeD spline estimation of multivariate Archimedean copulas. *Computational Statistics & Data Analysis*, 52(7):3570–3582.

Ding, Z., Granger, C. W. J., and Engle, R. F. (1993). A long memory property of stock market returns and a new model. *Journal of Empirical Finance*, 1(1):83–106.

Domingos, P. (1999). The role of Occam's razor in knowledge discovery. *Data Mining and Knowledge Discovery*, 3(4):409–425.

Dominici, D. (2003). The inverse of the cumulative standard normal probability function. *Integral Transforms and Special Functions*, 14(4):281–292.

Donoho, D. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306.

Dowd, K. (2005). *Measuring Market Risk*. John Wiley & Sons.

Drost, F. C., Klaassen, C. A. J., and Werker, B. J. M. (1997). Adaptive estimation in time-series models. *The Annals of Statistics*, 25(2):786–817.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. John Wiley & Sons.

Dudoit, S. and Fridlyand, J. (2003). *Statistical Analysis of Gene Expression Microarray Data*, chapter Classification in Microarray Experiments, pages 93–158. Chapman & Hall / CRC.

Duong, T. (2007). ks: Kernel density estimation and kernel discriminant analysis for multivariate data in R. *Journal of Statistical Software*, 21(7):1–16.

Duong, T. and Hazelton, M. (2003). Plug-in bandwidth matrices for bivariate kernel density estimation. *Journal of Nonparametric Statistics*, 15(1):17–30.

Edgar, R., Domrachev, M., and Lash, A. E. (2002). Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 1(30):207–210.

Eilers, P. H. C. and Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89–121.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007.

Engle, R. F. and González-Rivera, G. (1991). Semiparametric ARCH models. *The Journal of Business and Economic Statistics*, 9(4):345–359.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, 25(2):383–417.

Fang, K. T., Kotz, S., and Ng, K. W. (1990). *Symmetric Multivariate and Related Distributions*. Chapman and Hall.

Fenton, V. M. and Gallant, A. R. (1996). Qualitative and asymptotic performance of SNP density estimators. *Journal of Econometrics*, 74(1):77–118.

Ferenstein, E. and Gasowski, M. (2004). Modelling stock returns with AR-GARCH processes. *SORT*, 28(1):55–68.

Fermanian, J. and Scaillet, O. (2003). Nonparametric estimation of copulas for time series. *The Journal of Risk*, 5(4):25–54.

Forsberg, L. and Bollerslev, T. (2002). Bridging the gap between the distribution of realized (ECU) volatility and ARCH modelling (of the euro): the GARCH-NIG model. *Journal of Applied Econometrics*, 17(5):535–548.

Futreal, P. A., Coin, L., Marshall, M., Down, T., Hubbard, T., Wooster, R., Rahman, N., and Stratton, M. R. (2004). A census of human cancer genes. *Nature Reviews Cancer*, 4(3):177–183.

Gagliardini, P. and Gourieroux, C. (2007). An efficient nonparametric estimator for models with nonlinear dependence. *Journal of Econometrics*, 137(1):189–229.

Galambos, J. (1975). Order statistics of samples from multivariate distributions. *Journal of the American Statistical Association*, 70(351):674–680.

Gallant, A. R., Hsieh, D., and Tauchen, G. (1997). Estimation of stochastic volatility models with diagnostics. *Journal of Econometrics*, 81(1):159–192.

Gallant, R. A. and Nychka, D. W. (1987). Semi-nonparametric maximum likelihood estimation. *Econometrica*, 55(2):363–390.

Gardner, T. S. and Faith, J. J. (2005). Reverse-engineering transcription control networks. *Physics of Life Reviews*, 2(1):65–88.

Gautier, L., Cope, L., Bolstad, B. M., and Irizarry, R. A. (2004). affy–analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics*, 20(3):307–315.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58.

Geman, S., Geman, D., Abend, K., Harley, T. J., and Kanal, L. N. (1993). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *Journal of Applied Statistics*, 20(5):25–62.

Genest, C., Ghoudi, K., and Rivest, L. P. (1995). A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, 82(3):543–552.

Genest, C. and Rivest, L. (1993). Statistical inference procedures for bivariate Archimedean copulas. *Journal of the American Statistical Association*, 88(423):1034–1043.

Genton, M. G. (2004). *Skew-elliptical Distributions and their Applications: a Journey Beyond Normality*. Chapman & Hall/CRC.

George, E. I. and McCulloch, R. E. (1997). Approaches for Bayesian variable selection. *Statistica Sinica*, 7(2):339–373.

Ghosh, P., Gill, P., Muthukumarana, S., and Swartz, T. (2010). A semiparametric Bayesian approach to network modelling using Dirichlet process prior distributions. *Australian & New Zealand Journal of Statistics*, 52(3):289–302.

Gill, P. E., Golub, G. H., Murray, W., and Saunders, M. A. (1974). Methods for modifying matrix factorizations. *Mathematics of Computation*, 28(126):505–535.

Guyon, I., Janson, B., Stephen, B., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422.

Härdle, W., Müller, M., Sperlich, S., and Werwatz, A. (2004). *Nonparametric and Semiparametric Models*. Springer.

Hastie, T., Buja, A., and Tibshirani, R. (1995). Penalized discriminant analysis. *The Annals of Statistics*, 23(1):73–102.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inferene, and Prediction*. Springer.

Haury, A. C., Jacob, L., and Vert, J. P. (2010). Increasing stability and interpretability of gene expression signatures. *ArXiv e-prints*, (arXiv:1001.3109):1–18.

Herbrich, R., Graepel, T., and Campbell, C. (2001). Bayes point machines. *Journal of Machine Learning Research*, 1:245–279.

Hernández-Lobato, D. and Hernández-Lobato, J. M. (2008). Bayes machines for binary classification. *Pattern Recognition Letters*, 29(10):1466 – 1473.

Hernández-Lobato, D., Hernández-Lobato, J. M., Helleputte, T., and Dupont, P. (2010). Expectation propagation for Bayesian multi-task feature selection. In Balcázar et al. (2010), pages 522–537.

Hernández-Lobato, D., Hernández-Lobato, J. M., and Suárez, A. (2010a). Expectation propagation for microarray data classification. *Pattern Recognition Letters*, 31(12):1618–1626. Pattern Recognition of Non-Speech Audio.

Hernández-Lobato, J. M., Dijkstra, T., and Heskes, T. (2008). Regulator discovery from gene expression time series of malaria parasites: a hierachical approach. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems 20*, pages 649–656. MIT Press, Cambridge, MA.

Hernández-Lobato, J. M. and Dijkstra, T. M. H. (2010). Hub gene selection methods for the reconstruction of transcription networks. In Balcázar et al. (2010).

Hernández-Lobato, J. M., Hernández-Lobato, D., and Suárez, A. (2007). GARCH processes with non-parametric innovations for market risk estimation. In de Sá, J. M., Alexandre, L. A., Duch, W., and Mandic, D. P., editors, *ICANN (2)*, volume 4669 of *Lecture Notes in Computer Science*, pages 718–727. Springer.

Hernández-Lobato, J. M., Hernández-Lobato, D., and Suárez, A. (2010b). Network-based sparse Bayesian classification. *Pattern Recognition*. In Press.

Hernández-Lobato, J. M. and Suárez, A. (2009). Modeling dependence in financial data with semiparametric Archimedean copulas. In *International Workshop on Advances in Machine Learning for Computational Finance*. http://web.mac.com/davidrh/AMLCF09/papers/1.pdf.

Hoti, F. and Holmström, L. (2004). A semiparametric density estimation approach to pattern classification. *Pattern Recognition*, 37(3):409–419.

Ishwaran, H. and Rao, J. S. (2005). Spike and slab variable selection: Frequentist and Bayesian strategies. *The Annals of Statistics*, 33(2):730–773.

Jacob, L., Obozinski, G., and Vert, J. (2009). Group lasso with overlap and graph lasso. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 433–440, New York, NY, USA. ACM.

Jensen, S. T., Chen, G., and Stoeckert, C. J. (2007). Bayesian variable selection and data integration for biological regulatory networks. *The Annals of Applied Statistics*, 1(2):612–633.

Ji, S., Xue, Y., and Carin, L. (2008). Bayesian compressive sensing. *IEEE Transactions on Signal Processing*, 56(6):2346–2356.

Joe, H. (1997). *Multivariate Models and Dependence Concepts*. CRC Press.

Joe, H. (2005). Asymptotic efficiency of the two-stage estimation method for copula-based models. *Journal of Multivariate Analysis*, 94(2):401–419.

Johnstone, I. M. and Titterington, D. M. (2009). Statistical challenges of high-dimensional data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906):4237–4253.

Jorion, P. (1997). *Value at Risk: The New Benchmark for Controlling Market Risk*. McGraw-Hill.

Jurgelenaite, R., Dijkstra, T. M. H., Kocken, C. H. M., and Heskes, T. (2009). Gene regulation in the intraerythrocytic cycle of Plasmodium falciparum. *Bioinformatics*, 25(12):1484–1491.

Juri, A. and Wüthrich, M. V. (2003). Tail dependence from a distributional point of view. *Extremes*, 6(3):213–246.

Kaishev, V. K., Dimitrova, D. S., Haberman, S., and Verrall, R. (2006). Geometrically designed, varible knot regression splines assymptotics and inference. Technical Report Statistical Research Paper No. 28, Cass Business School, London.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

Kalousis, A., Prados, J., and Hilario, M. (2007). Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and Information Systems*, 12(1):95–116.

Kerkhof, J. and Melenberg, B. (2004). Backtesting for risk-based regulatory campital. *Journal of Banking and Finance*, 28(8):1845–1865.

Kim, Y., Kim, J., and Kim, Y. (2006). Blockwise sparse regression. *Statistica Sinica*, 16(2):375–390.

Kindermann, R. and Snell, J. L. (1980). *Markov Random Fields and Their Applications*. American Mathematical Society.

Kirshner, S. (2008). Learning with tree-averaged densities and distributions. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems 20*, pages 761–768. MIT Press, Cambridge, MA.

Kon, S. J. (1984). Models of stock returns–a comparison. *Journal of Finance*, 39(1):147–165.

Kosorok, M. R. (2009). What's so special about semiparametric methods? *Sankhyā: The Indian Journal of Statistics*, 71-A(2):331–353.

Kuncheva, L. I. (2007). A stability index for feature selection. In *Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications*, pages 390–395. ACTA Press.

Kupiec, P. H. (1995). Techniques for verifying the accuracy of risk measurement models. *The Journal of Derivatives*, 3(2):73–84.

Lambert, P. (2007). Archimedean copula estimation using Bayesian splines smoothing techniques. *Computational Statistics & Data Analysis*, 51(12):6307–6320.

Lange, K. (1999). *Numerical Analysis for Statisticians*. Springer.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, K. E., Sha, N., Dougherty, E. R., Vannucci, M., and Mallick, B. K. (2003). Gene selection: a Bayesian variable selection approach. *Bioinformatics*, 19(1):90–97.

Lees, J. M. (2008). GEOmap: Topographic and geologic mapping. R package available at http://lib.stat.cmu.edu/R/CRAN.

Li, C. and Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24(9):1175–1182.

Li, F. and Zhang, N. R. (2010). Bayesian variable selection in structured high-dimensional covariate spaces with applications in genomics. *Journal of the American Statistical Association*, (In Press).

Llinás, M., Bozdech, Z., Wong, E. D., Adai, A., and DeRisi, J. L. (2006). Comparative whole genome transcriptome analysis of three *Plasmodium falciparum* strains. *Nucleic Acids Research*, 34(4):1166–1173.

Loscalzo, S., Yu, L., and Ding, C. (2009). Consensus group stable feature selection. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 567–576.

Lucas, J., Carvalho, C., Wang, Q., Bild, A., Nevins, J., and West, M. (2006). Sparse statistical modelling in gene expression genomics. In Müller, P., Do, K. A., and Vannucci, M., editors, *Bayesian Inference for Gene Expression and Proteomics*, pages 155–176. Cambridge University Press.

MacKay, D. J. C. (1992). Bayesian interpolation. *Neural Computation*, 4(3):415–447.

MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

Malevergne, Y. and Sornette, D. (2006). *Extreme Financial Risks: From Dependence to Risk Management*. Springer.

Mandelbrot, B. (1963). The variation of certain speculative prices. *The Journal of Business*, 36(4):394–419.

Manning, C. D. and Schütze, H. (2000). *Foundations of Statistical Natural Language Processing*. MIT Press.

Marbach, D., Schaffter, T., Mattiussi, C., and Floreano, D. (2009). Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology*, 16(2):229–239.

Markowitz, H. (1991). *Portfolio Selection*. Blackwell Publishing.

McNeil, A. J. and Nešlehová, J. (2007). Multivariate Archimedean copulas, d-monotone functions and ℓ_1-norm symmetric distributions. *The Annals of Statistics*, 37(5B):3059–3097.

Menon, A. and Elkan, C. (2010). Predicting labels for dyadic data. *Data Mining and Knowledge Discovery*, 21(2):327–343.

Mikosch, T. and Starica, C. (2000). Limit theory for the sample autocorrelations and extremes of a GARCH(1,1) process. *The Annals of Statistics*, 28(5):1427–1451.

Miller, A. J. (2002). *Subset Selection in Regression*. Chapman & Hall/CRC.

Miller, L. D., Smeds, J., George, J., Vega, V. B., Vergara, L., Ploner, A., Pawitan, Y., Hall, P., Klaar, S., Liu, E. T., and Jonas, B. (2005). An expression signature for p53 status in human breast cancer predicts mutation status, transcriptional effects, and patient survival. *Proceedings of the National Academy of Sciences USA*, 102(38):13550–13555.

Minka, T. (2001). *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, MIT.

Minka, T. and Lafferty, J. (2002). Expectation-propagation for the generative aspect model. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 352–359.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York.

Mittnik, S., Doganoglu, T., and Chenyao, D. (1997). Computing the probability density function of the stable paretian distribution. *Mathematical and Computer Modelling*, 29(10):235–240.

Nelsen, R. B. (2006). *An Introduction to Copulas*. Springer.

Newey, W. K. (1990). Semiparametric efficiency bounds. *Journal of Applied Econometrics*, 5(2):99–135.

Nickisch, H. and Seeger, M. W. (2009). Convex variational Bayesian inference for large scale generalized linear models. In *ICML*, pages 761–768. ACM.

Nolan, J. P. (2002). *Stable Distributions*. Birkhauser.

Osborne, M. F. M. (1959). Brownian motion in the stock market. *Operations Research*, 7(2):145–173.

O'Sullivan, F. (1986). A statistical perpective on ill-posed inverse problems. *Statistical Science*, 1(4):502–518.

Panorska, A. K., Mittnik, S., and Rachev, S. T. (1995). Stable GARCH models for financial time series. *Applied Mathematics Letters*, 8(5):33–37.

Park, M., Hastie, T., and Tibshirani, R. (2007). Averaged gene expressions for regression. *Biostatistics*, 8(2):212–227.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076.

Pathway Commons (2009). http://www.pathwaycommons.org.

Patton, A. J. (2006). Modelling asymmetric exchange rate dependence. *International Economic Review*, 47(2):527–556.

Pereira, F., Mitchell, T., and Botvinick, M. (2009). Machine learning classifiers and fMRI: A tutorial overview. *NeuroImage*, 45(1):199 – 209.

Praetz, P. D. (1972). The distribution of share price changes. *The Journal of Business*, 45(1):49–55.

Prause, K. (1999). The generalized hyperbolic model: Estimation, financial derivatives and risk measures. *PhD Dissertation, University of Freiburg*.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1992). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press.

Qi, Y. A., Minka, T. P., Picard, R., and Ghahramani, Z. (2004). Predictive automatic relevance determination by expectation propagation. In *Proceedings of the twenty-first international conference on Machine learning*, pages 85–92, New York, NY, USA. ACM.

Raible, S. (2000). *Lévy Processes in Finance: Theory, Numerics, and Empirical Facts*. PhD thesis, University of Freiburg, Germany.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Razuvaev, V. N., Apasova, E. B., and Martuganov, R. A. (2008). Daily temperature and precipitation data for 223 former-USSR stations. *Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, U.S. Department of Energy, Oak Ridge, Tennessee*, ORNL/CDIAC-56(NDP-040).

Reinsch, C. H. (1967). Smoothing by spline functions. *Numerische Mathematik*, 10(3):177–183.

Renka, R. J. (1997). Algorithm 772: STRIPACK: Delaunay triangulation and Voronoi diagram on the surface of a sphere. *ACM Trans. Math. Softw.*, 23(3):416–434.

Roth, V. and Fischer, B. (2008). The group-lasso for generalized linear models: Uniqueness of solutions and efficient algorithms. In *ICML*, pages 848–855. ACM.

Sabatti, C. and James, G. M. (2006). Bayesian sparse hidden components analysis for transcription regulation networks. *Bioinformatics*, 22(6):739–746.

Sakata, T. and Winzeler, E. A. (2007). Genomics, systems biology and drug development for infectious diseases. *Molecular BioSystems*, 3:841–848.

Samuelson, P. A. (1965). Proof that properly anticipated prices fluctuate randomly. *Industrial Management Review*, 6(2):41–49.

Sandler, T., Talukdar, P. P., Ungar, L. H., and Blitzer, J. (2008). Regularized learning with networks of features. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 21*, pages 1401–1408.

Scott, D. W., Tapia, R. A., and Thompson, J. R. (1980). Nonparametric probability density estimation by discrete maximum penalized-likelihood criteria. *The Annals of Statistics*, 8(4):820–832.

Seeger, M., Nickisch, H., and Schlkopf, B. (2010). Optimization of *k*-space trajectories for compressed sensing by Bayesian experimental design. *Magnetic Resonance in Medicine*, 63(1):116–126.

Seeger, M. W. (2008). Bayesian inference and optimal design for the sparse linear model. *The Journal of Machine Learning Research*, 9:759–813.

Sen, Z. (2009). *Spatial Modeling Principles in Earth Sciences*. Springer.

Severini, T. A. and Wong, W. H. (1992). Profile likelihood and conditionally parametric models. *The Annals of Statistics*, 20(4):1768–1802.

SGD project (2007). Saccharomyces genome database. http://www.yeastgenome.org.

Sheather, S. J. (2004). Density estimation. *Statistical Science*, 19(4):588–597.

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):683–690.

Shen, X., Zhu, Y., and Song, L. (2008). Linear B-spline copulas with applications to nonparametric estimation of copulas. *Computational Statistics & Data Analysis*, 52(7):3806–3819.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC.

Sklar, A. (1959). Fonctions de répartion à n dimensions et leurs marges. *Publ. Inst. Stat. Univ. Paris*, 8:229–231.

Slawski, M., zu Castell, W., and Tutz, G. (2009). Feature selection guided by structural information. Technical Report 51, Department of Statistics, University of Munich, LMU.

Slonim, D. K. (2002). From patterns to pathways: Gene expression data analysis comes of age. *Nature Genetics*, 32:502–508.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher., B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 9(12):3273–3297.

Steinke, F., Seeger, M., and Tsuda, K. (2007). Experimental design for efficient identification of gene regulatory networks using sparse Bayesian models. *BMC Systems Biology*, 1(1):51.

Stern, D. H., Herbrich, R., and Graepel, T. (2009). Matchbox: Large scale online Bayesian recommendations. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 111–120, New York, NY, USA. ACM.

Stolovitzky, G., Monroe, D., and Califano, A. (2007). Dialogue on reverse-engineering assessment and methods. *Annals of the New York Academy of Sciences*, 1115:1–22.

Taylor, S. (1986). *Modelling Financial Time Series*. John Wiley & Sons.

Team, R. D. C. (2007). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Terrell, G. R. (1990). The maximal smoothing principle in density estimation. *Journal of the American Statistical Association*, 85(410):470–477.

Thieffry, D., Huerta, A. M., Pérez-Rueda, E., and Collado-Vides, J. (1998). From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in Escherichia coli. *BioEssays*, 20(5):433–440.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58(1):267–288.

Tienda-Luna, I. M., Yin, Y., Carrion, M. C., Huang, Y., Cai, H., Sanchez, M., and Wang, Y. (2008). Inferring the skeleton cell cycle regulatory network of malaria parasite using comparative genomic and variational Bayesian approaches. *Genetica*, 132(2):131–142.

Tikhonov, A. N. and Arsenin, V. Y. (1977). *Solutions of Ill-Posed Problems*. Wiley, New York.

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244.

Tipping, M. E. and Faul, A. (2003). Fast marginal likelihood maximisation for sparse Bayesian models. In Bishop, C. M. and Frey, B. J., editors, *Proceedings of the ninth international workshop on artificial intelligence and statistics*.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525.

Vaart, V. D. (2000). *Asymptotic Statistics*. Cambridge University Press.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

van Gerven, M., Cseke, B., Oostenveld, R., and Heskes, T. (2009). Bayesian source localization with the multivariate Laplace prior. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A., editors, *Advances in Neural Information Processing Systems 22*, pages 1901–1909.

van Gerven, M. A., Cseke, B., de Lange, F. P., and Heskes, T. (2010). Efficient Bayesian multivariate fMRI analysis using a sparsifying spatio-temporal prior. *NeuroImage*, 50(1):150 – 161.

Vandenhende, F. and Lambert, P. (2005). Local dependence estimation using semiparametric Archimedean copulas. *The Canadian Journal of Statistics*, 33(3):377–388.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer.

Wand, M. P., Marron, J. S., and Ruppert, D. (1991). Transformations in density estimation. with discussion and a rejoinder by the authors. *Journal of the American Statistical Association*, 86(414):343–361.

Wasserman, L. (2003). *All of Statistics: A Concise Course in Statistical Inference*. Springer.

Wasserman, L. (2006). *All of Nonparametric Statistics*. Springer.

Wei, Z. and Li, H. (2007). A Markov random field model for network-based analysis of genomic data. *Bioinformatics*, 23(12):1537–1544.

Wipf, D., Palmer, J., and Rao, B. (2004). Perspectives on Sparse Bayesian Learning. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA.

Wood, S. N. and Augustin, N. H. (2002). GAMs with integrated model selection using penalized regression splines and applications to environmental modelling. *Ecological Modelling*, 157(2-3):157–177.

Wuertz, Diethelm, and Others (2004). *fBasics: Financial Software Collection - fBasics*.

Yahoo! Finance (2008). Yahoo Inc.

Young, J., Johnson, J., Benner, C., Yan, S. F., Chen, K., Le Roch, K., Zhou, Y., and Winzeler, E. (2008). In silico discovery of transcription regulatory elements in Plasmodium falciparum. *BMC Genomics*, 9(1):70.

Yu, L., Ding, C., and Loscalzo, S. (2008). Stable feature selection via dense feature groups. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 803–811.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–67.

Zhu, H. and Rohwer, R. (1995). Bayesian invariant measurements of generalization. *Neural Processing Letters*, 2(6):28–31.

Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R. (2004). 1-norm support vector machines. In *NIPS*, pages 49–56. MIT Press.

Zhu, Y., Shen, X., and Pan, W. (2009). Network-based support vector machine for classification of microarray samples. *BMC Bioinformatics*, 10(Suppl 1):S21.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, 67(2):301–320.

Zou, H. and Yuan, M. (2008). The F_∞-norm support vector machine. *Statistica Sinica*, 18(1):379–398.