Black-box α-divergence Minimization

José Miguel Hernández–Lobato[1]
Harvard Intelligent Probabilistic Systems Group
School of Engineering and Applied Sciences
Harvard University

http://jmhl.org, jmh@seas.harvard.edu

Joint work with
Yingzhen Li[1], Mark Rowland, Daniel Hernández-Lobato,
Thang D. Bui and Rich E. Turner.

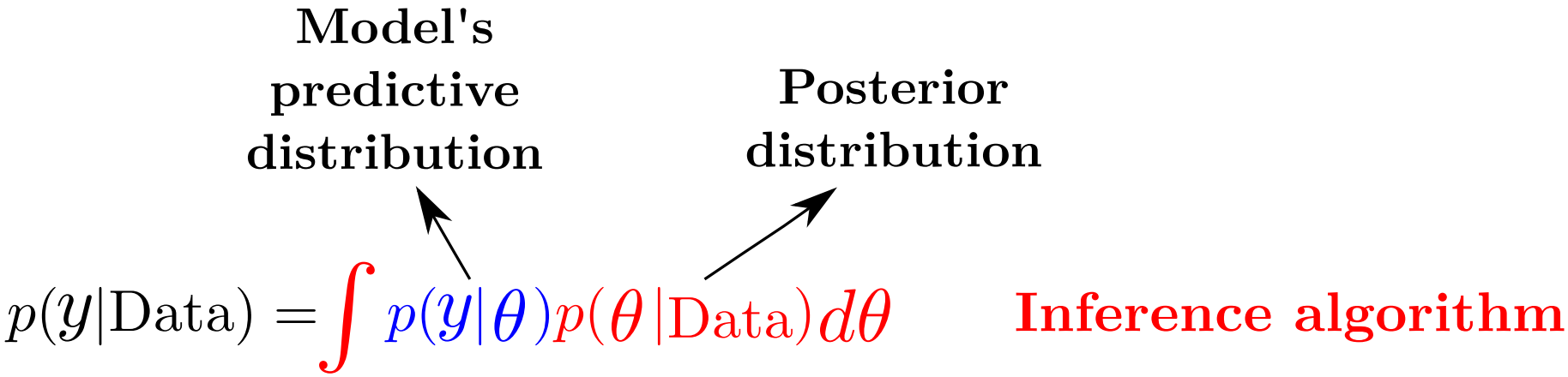
[1] Equal contributors.

$$p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data})d\theta$$ **Inference algorithm**

Model's predictive distribution

$$p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data})d\theta$$

Inference algorithm


Model's predictive distribution
Posterior distribution
$p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data})d\theta$
Inference algorithm

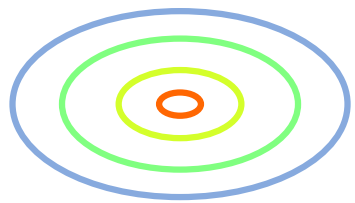
Model's predictive distribution

Posterior distribution

$$p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data})d\theta$$

Inference algorithm

Approximation $q(\theta) =$



α-divergence

$$D_\alpha(p||q) = \frac{\int_{\boldsymbol{\theta}} \left(\alpha p(\boldsymbol{\theta}) + (1-\alpha) q(\boldsymbol{\theta}) - p(\boldsymbol{\theta})^\alpha q(\boldsymbol{\theta})^{1-\alpha}\right) \, d\boldsymbol{\theta}}{\alpha(1-\alpha)} \, .$$

[Amari, 1985].

α-divergence

$$D_\alpha(p||q) = \frac{\int_{\boldsymbol{\theta}} \left(\alpha p(\boldsymbol{\theta}) + (1-\alpha) q(\boldsymbol{\theta}) - p(\boldsymbol{\theta})^\alpha q(\boldsymbol{\theta})^{1-\alpha}\right) \, d\boldsymbol{\theta}}{\alpha(1-\alpha)} \, .$$

[Amari, 1985].

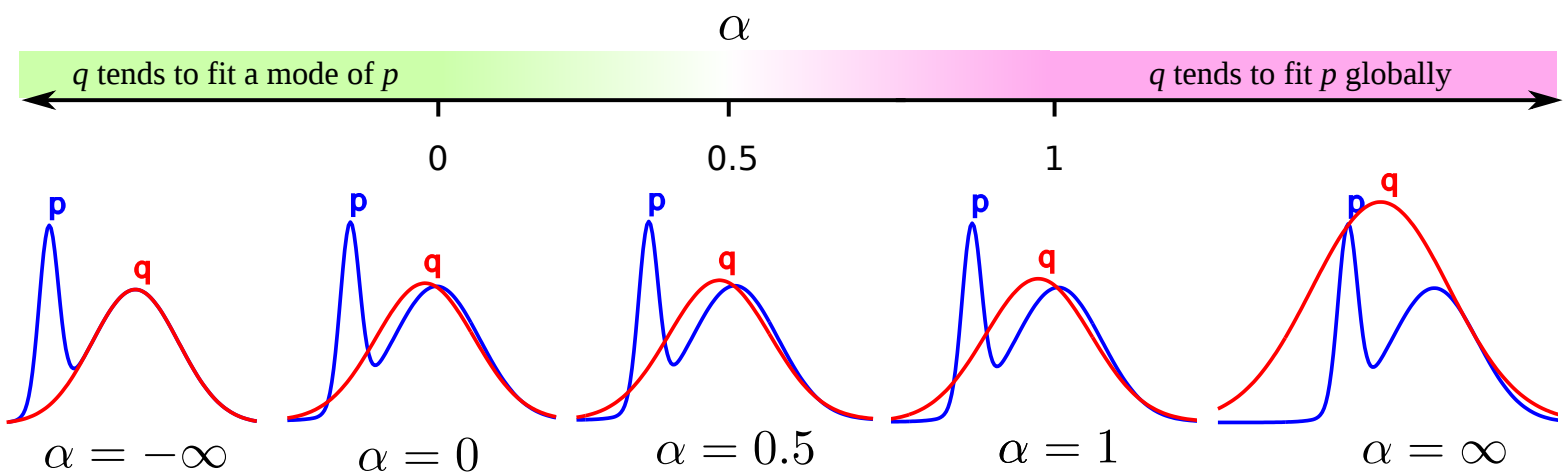


Figure source: [Minka, 2005].

α-divergence

$$D_\alpha(p||q) = \frac{\int_{\boldsymbol{\theta}} \left(\alpha p(\boldsymbol{\theta}) + (1-\alpha) q(\boldsymbol{\theta}) - p(\boldsymbol{\theta})^\alpha q(\boldsymbol{\theta})^{1-\alpha}\right) \, d\boldsymbol{\theta}}{\alpha(1-\alpha)} \, .$$

[Amari, 1985].

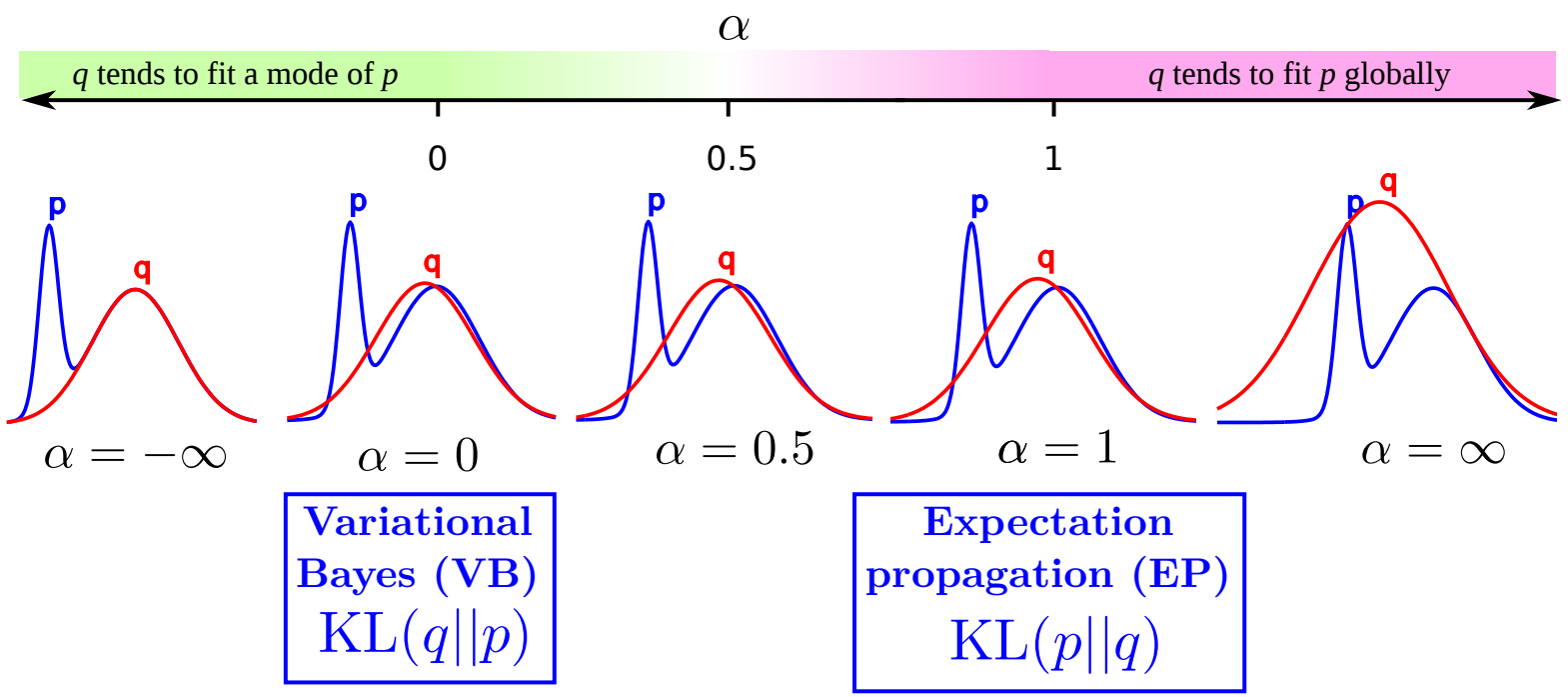


Figure source: [Minka, 2005].

α-divergence

$$D_\alpha(p||q) = \frac{\int_{\boldsymbol{\theta}} \left(\alpha p(\boldsymbol{\theta}) + (1-\alpha) q(\boldsymbol{\theta}) - p(\boldsymbol{\theta})^\alpha q(\boldsymbol{\theta})^{1-\alpha}\right) \, d\boldsymbol{\theta}}{\alpha(1-\alpha)} \, .$$

[Amari, 1985].

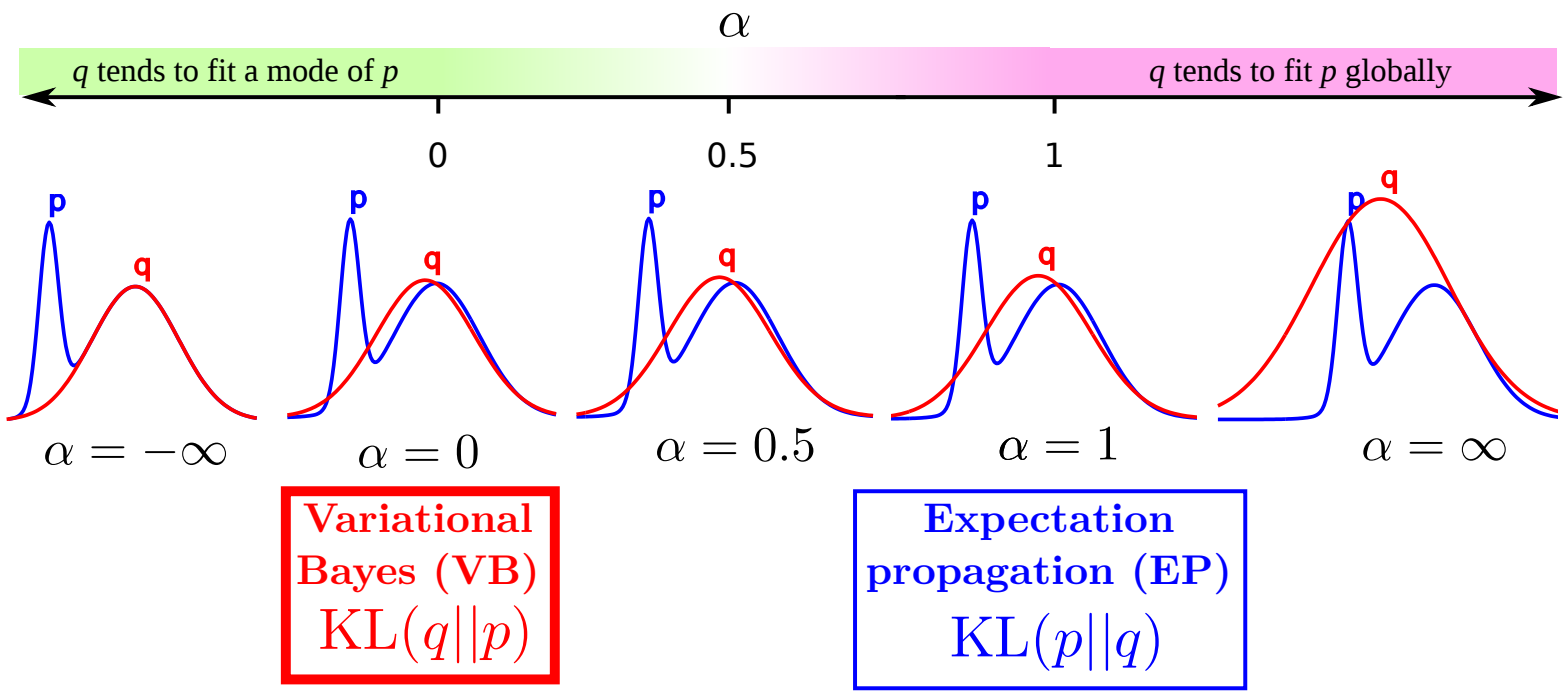


Figure source: [Minka, 2005].

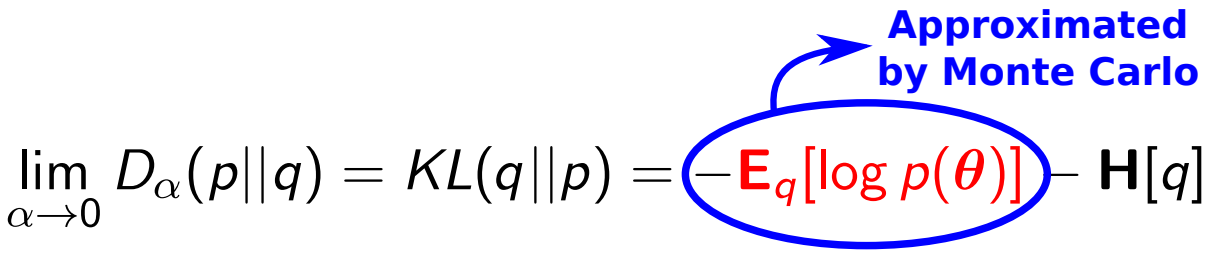
Variational Bayes

$$\lim_{\alpha \to 0} D_\alpha(p||q) = KL(q||p) = -\mathbf{E}_q[\log p(\boldsymbol{\theta})] - \mathbf{H}[q]$$

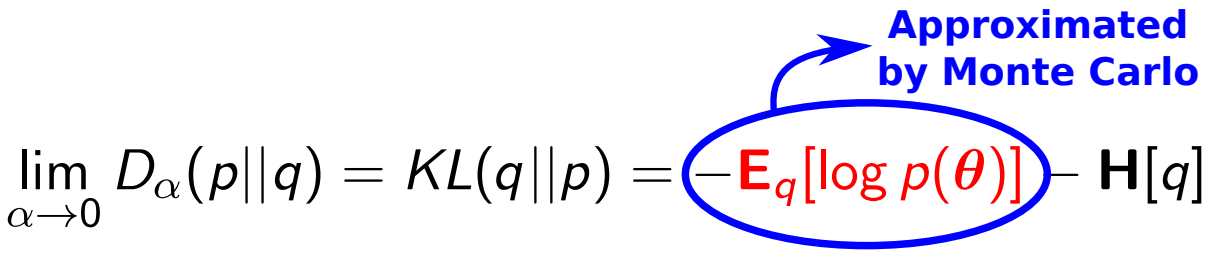
Variational Bayes

$$\lim_{\alpha \to 0} D_\alpha(p||q) = KL(q||p) = -\mathbf{E}_q[\log p(\theta)] - \mathbf{H}[q]$$

Variational Bayes



Variational Bayes

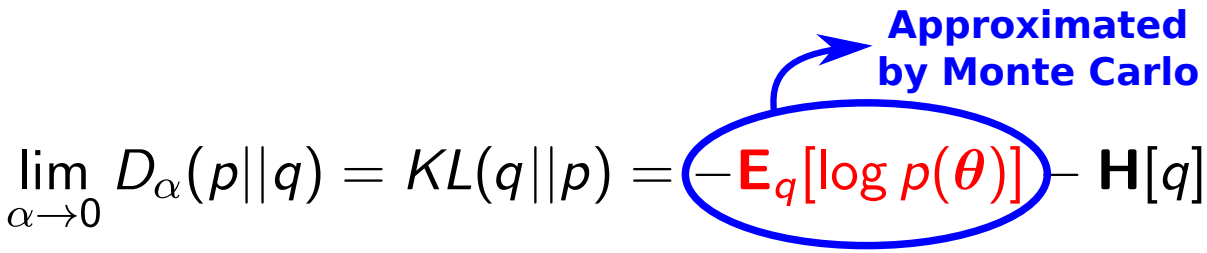


There are automatic (black-box) tools for minimizing this objective using

- Stochastic optimization.
- Automatic differentiation.

[Kucukelbir et al., 2015, Ranganath et al., 2014, Salimans et al., 2013].

Variational Bayes



There are automatic (black-box) tools for minimizing this objective using

- Stochastic optimization.
- Automatic differentiation.

[Kucukelbir et al., 2015, Ranganath et al., 2014, Salimans et al., 2013].

Can we have similar tools for other values of α?

Local α-divergence minimization (Power EP)

Approximates $\boxed{p(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} f_n(\boldsymbol{\theta})}$ with $\boxed{q(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} \tilde{f}_n(\boldsymbol{\theta})}$

[Minka, 2004]

Local α-divergence minimization (Power EP)

Approximates $p(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} f_n(\boldsymbol{\theta})$ with $q(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} \tilde{f}_n(\boldsymbol{\theta})$

[Minka, 2004]

$p(\boldsymbol{\theta}) \propto \quad p_0(\boldsymbol{\theta}) \quad f_1(\boldsymbol{\theta}) \; f_2(\boldsymbol{\theta}) \; f_3(\boldsymbol{\theta}) \quad \approx \quad q(\boldsymbol{\theta}) \propto \quad p_0(\boldsymbol{\theta}) \quad \tilde{f}_1(\boldsymbol{\theta}) \; \tilde{f}_2(\boldsymbol{\theta}) \; \tilde{f}_3(\boldsymbol{\theta})$

Local α-divergence minimization (Power EP)

Approximates $p(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} f_n(\boldsymbol{\theta})$ with $q(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} \tilde{f}_n(\boldsymbol{\theta})$

[Minka, 2004]

$$p(\boldsymbol{\theta}) \propto \quad p_0(\boldsymbol{\theta}) \quad f_1(\boldsymbol{\theta}) \; f_2(\boldsymbol{\theta}) \; f_3(\boldsymbol{\theta}) \quad \approx \quad q(\boldsymbol{\theta}) \propto \quad p_0(\boldsymbol{\theta}) \quad \tilde{f}_1(\boldsymbol{\theta}) \; \tilde{f}_2(\boldsymbol{\theta}) \; \tilde{f}_3(\boldsymbol{\theta})$$

The $\tilde{f}_n$ are tuned by minimizing the local α-divergences

$$D_\alpha[p_n||q] \quad \text{for } n = 1, \ldots, N, \quad \text{where} \quad \begin{array}{rcl} p_n(\boldsymbol{\theta}) & \propto & f_n(\boldsymbol{\theta}) \prod_{j \neq n} \tilde{f}_j(\boldsymbol{\theta}) \\ q(\boldsymbol{\theta}) & \propto & \tilde{f}_n(\boldsymbol{\theta}) \prod_{j \neq n} \tilde{f}_j(\boldsymbol{\theta}) \end{array}.$$

The Power-EP approximation to the **evidence** is given by

$$\log Z_{\mathrm{PEP}} = \log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}_n(\boldsymbol{\theta})} \right)^{\alpha} \right] ,$$

The Power-EP approximation to the **evidence** is given by

$$\log Z_{\text{PEP}} = \log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}_n(\boldsymbol{\theta})} \right)^{\alpha} \right] ,$$

The power-EP solution for q can be obtained by solving

$$\max_{q} \min_{\tilde{f}_1,\ldots,\tilde{f}_N} \log Z_{\text{PEP}} \quad \text{subject to} \quad q(\boldsymbol{\theta}) = p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} \tilde{f}_n(\boldsymbol{\theta}) .$$

The Power-EP approximation to the **evidence** is given by

$$\log Z_{\text{PEP}} = \log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}_n(\boldsymbol{\theta})} \right)^{\alpha} \right] ,$$

The power-EP solution for q can be obtained by solving

$$\max_{q} \min_{\tilde{f}_1,\ldots,\tilde{f}_N} \log Z_{\text{PEP}} \quad \text{subject to} \quad q(\boldsymbol{\theta}) = p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} \tilde{f}_n(\boldsymbol{\theta}) .$$

Solved with **double-loop** algorithm [Heskes et al., 2002].

The Power-EP approximation to the **evidence** is given by

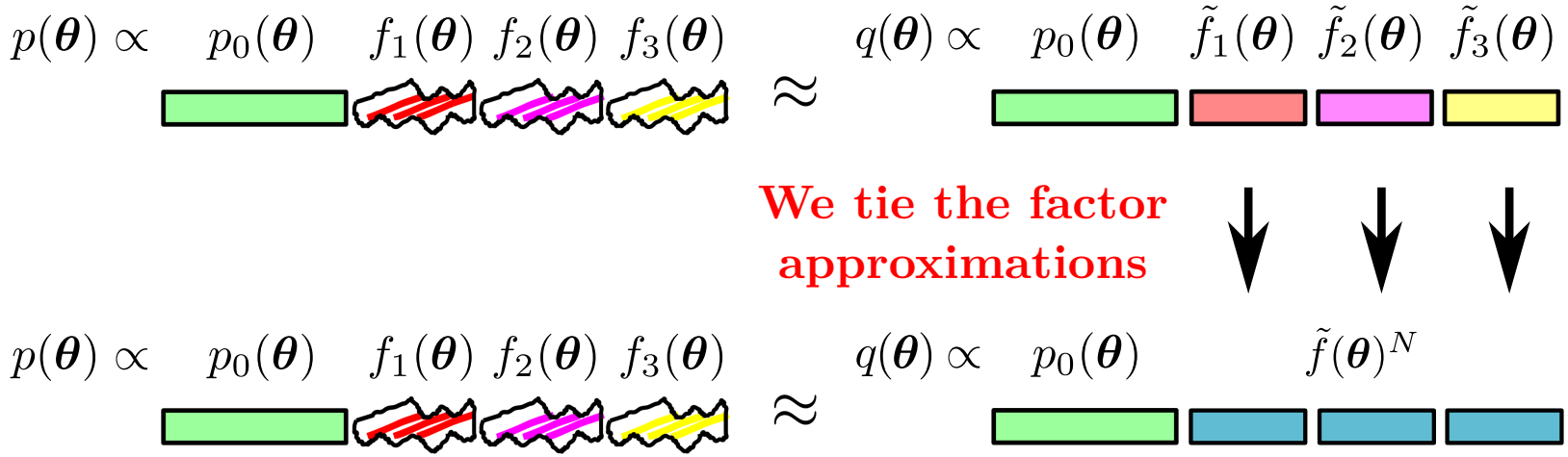
$$\log Z_{\text{PEP}} = \log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}_n(\boldsymbol{\theta})} \right)^{\alpha} \right] ,$$

The power-EP solution for q can be obtained by solving

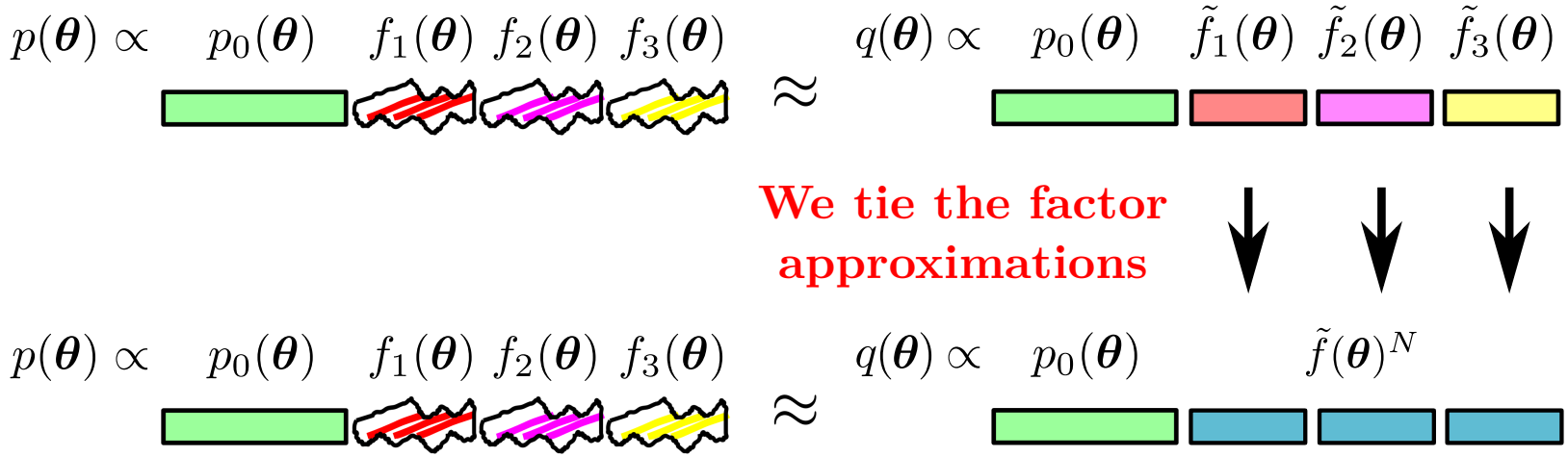
$$\max_{q} \min_{\tilde{f}_1,\ldots,\tilde{f}_N} \log Z_{\text{PEP}} \quad \text{subject to} \quad q(\boldsymbol{\theta}) = p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} \tilde{f}_n(\boldsymbol{\theta}) .$$

Solved with **double-loop** algorithm [Heskes et al., 2002]. **Too slow in practice!**

Main contribution

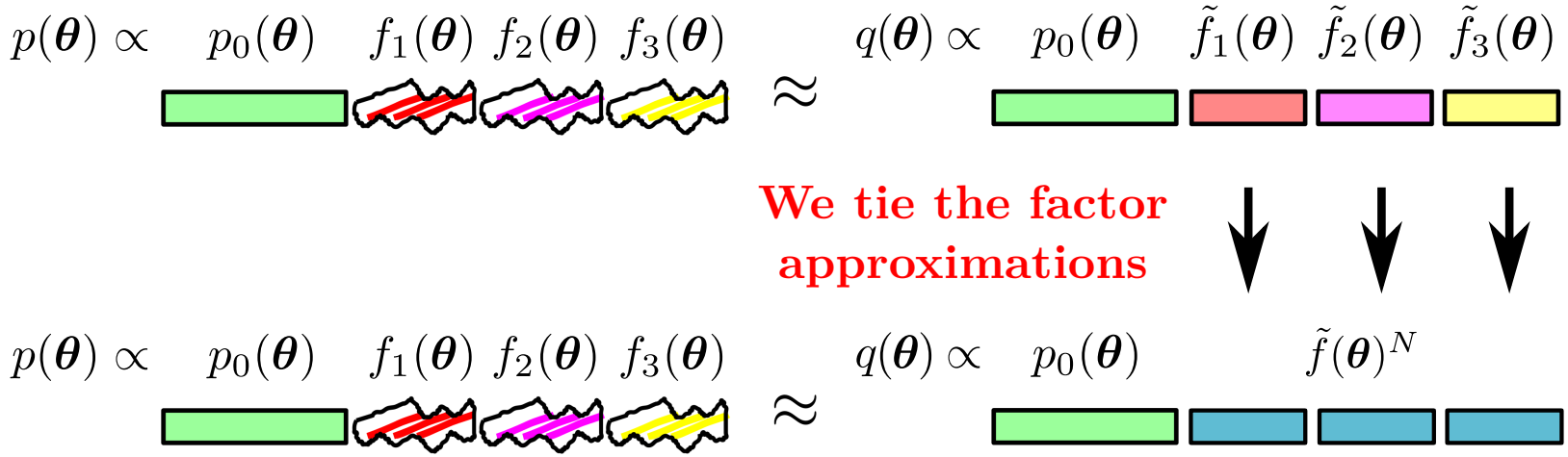


Main contribution



- $\max_{q} \min_{\tilde{f}_1,\ldots,\tilde{f}_N}$ problem $\rightarrow$ $\max_{q}$ problem, **no double-loop needed!**

Main contribution



- $\max_q \min_{\tilde{f}_1,\ldots,\tilde{f}_N}$ problem $\rightarrow$ $\max_q$ problem, **no double-loop needed!**
- **Memory saving** scales as $\mathcal{O}(N)$.

$$\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}_n(\boldsymbol{\theta})} \right)^{\alpha} \right]$$

$$\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}_n(\boldsymbol{\theta})} \right)^{\alpha} \right]$$

$$\downarrow$$

$$\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}(\boldsymbol{\theta})} \right)^{\alpha} \right]$$

$$\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}_n(\boldsymbol{\theta})} \right)^{\alpha} \right]$$

$$\downarrow$$

$$\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}(\boldsymbol{\theta})} \right)^{\alpha} \right]$$

$$\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}_n(\boldsymbol{\theta})} \right)^{\alpha} \right]$$

$$\downarrow$$

$$\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}(\boldsymbol{\theta})} \right)^{\alpha} \right]$$

$$\downarrow$$

$$\log Z_q + \frac{N}{|\mathbf{S}|} \sum_{n \in \mathbf{S}} \frac{1}{\alpha} \log \frac{1}{K} \sum_{k=1}^{K} \left(\frac{f_n(\boldsymbol{\theta}_k)}{\tilde{f}(\boldsymbol{\theta}_k)} \right)^{\alpha}$$

$\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K \sim q$

$$\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}_n(\boldsymbol{\theta})} \right)^{\alpha} \right]$$

$$\downarrow$$

$$\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}(\boldsymbol{\theta})} \right)^{\alpha} \right]$$

$$\downarrow$$

$$\log Z_q + \frac{N}{|\mathbf{S}|} \sum_{n \in \mathbf{S}} \frac{1}{\alpha} \log \frac{1}{K} \sum_{k=1}^{K} \left(\frac{f_n(\boldsymbol{\theta}_k)}{\tilde{f}(\boldsymbol{\theta}_k)} \right)^{\alpha}$$

$\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K \sim q$

Stochastic estimate of the objective for **automatic, scalable** inference.

$$\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}_n(\boldsymbol{\theta})} \right)^{\alpha} \right]$$

$$\downarrow$$

$$\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}(\boldsymbol{\theta})} \right)^{\alpha} \right]$$

$$\downarrow$$

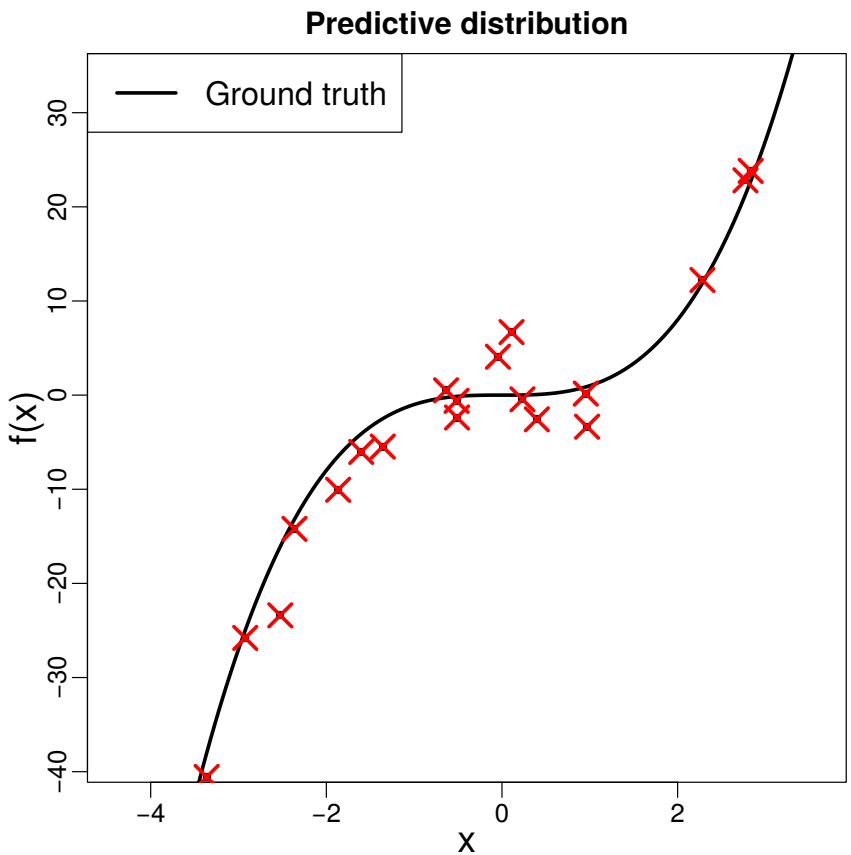
$$\log Z_q + \frac{N}{|\mathbf{S}|} \sum_{n \in \mathbf{S}} \frac{1}{\alpha} \log \frac{1}{K} \sum_{k=1}^{K} \left(\frac{f_n(\boldsymbol{\theta}_k)}{\tilde{f}(\boldsymbol{\theta}_k)} \right)^{\alpha}$$

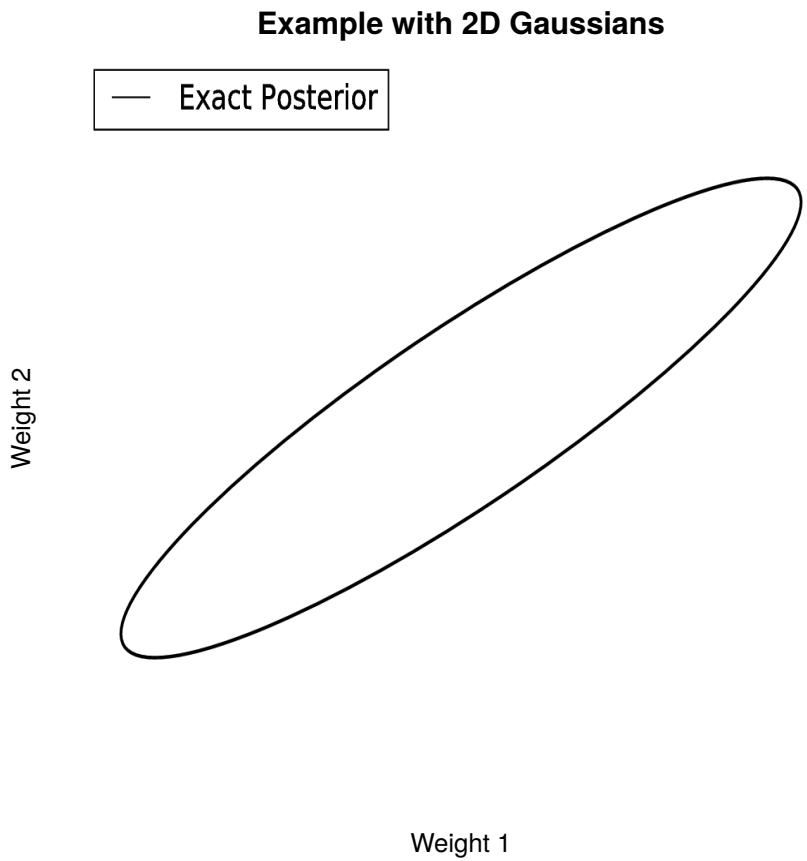
$\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K \sim q$

Stochastic estimate of the objective for **automatic, scalable** inference.

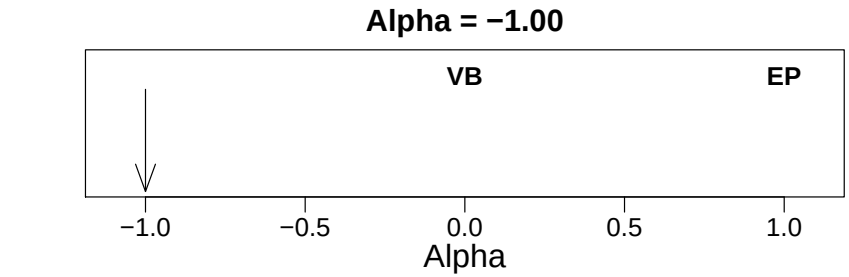
Biased estimator!

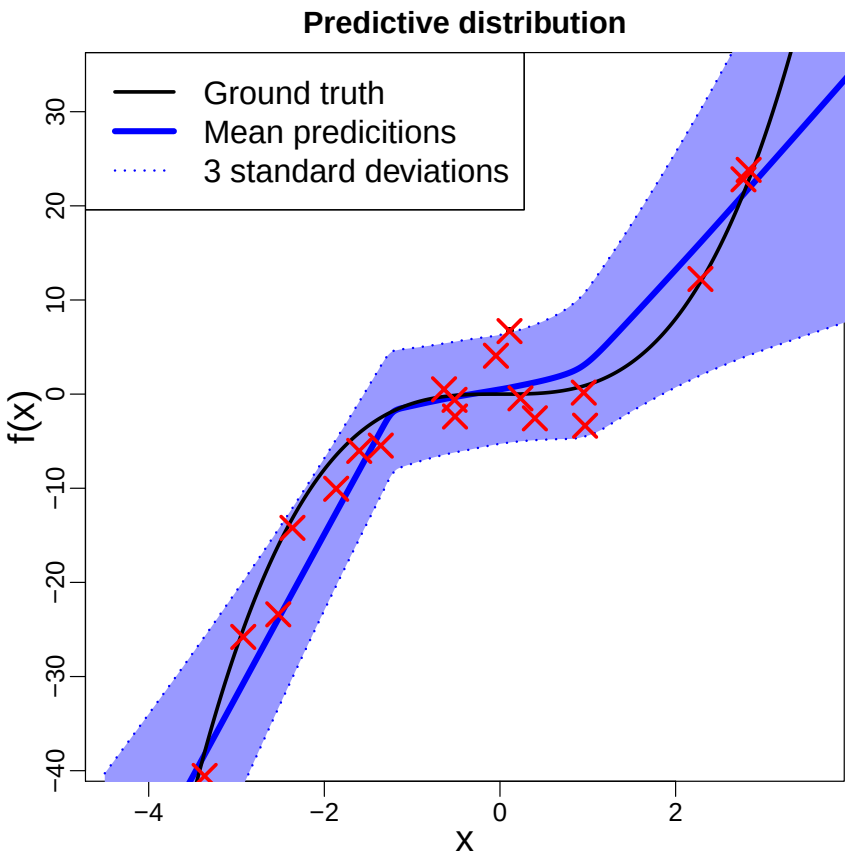
Regression with neural networks and 2D Gaussian

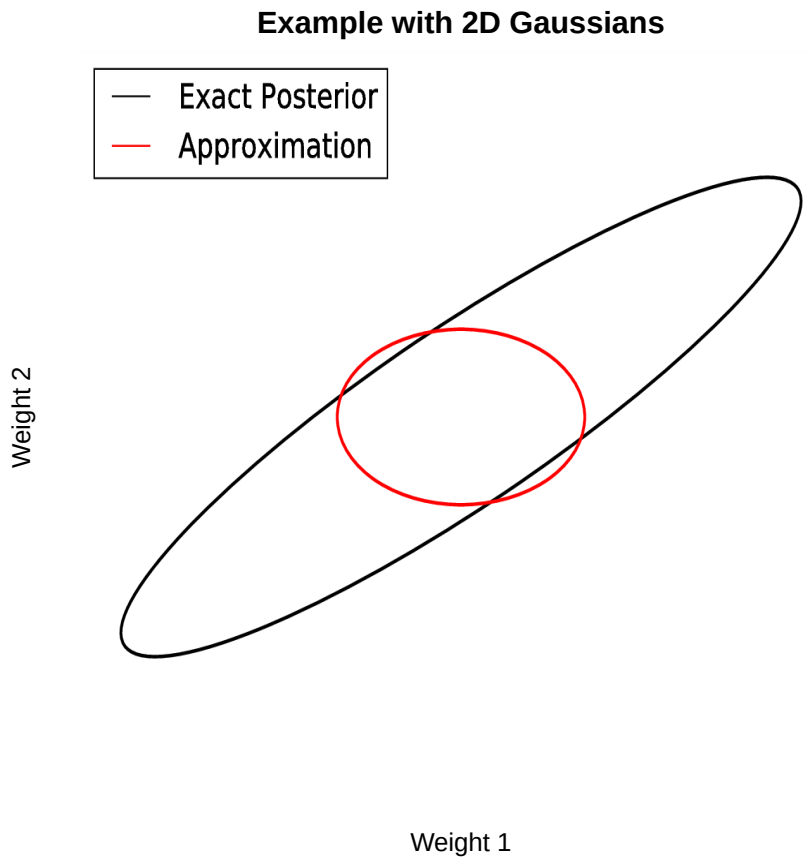




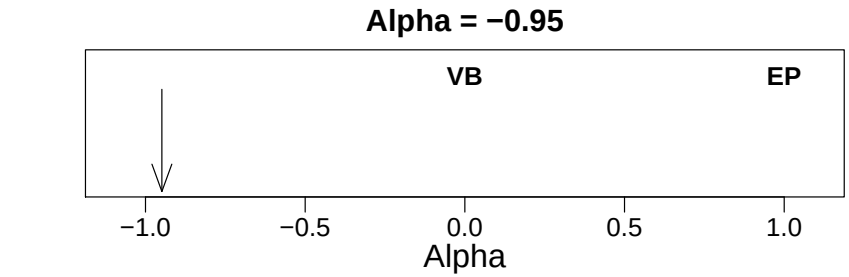
Regression with neural networks and 2D Gaussian

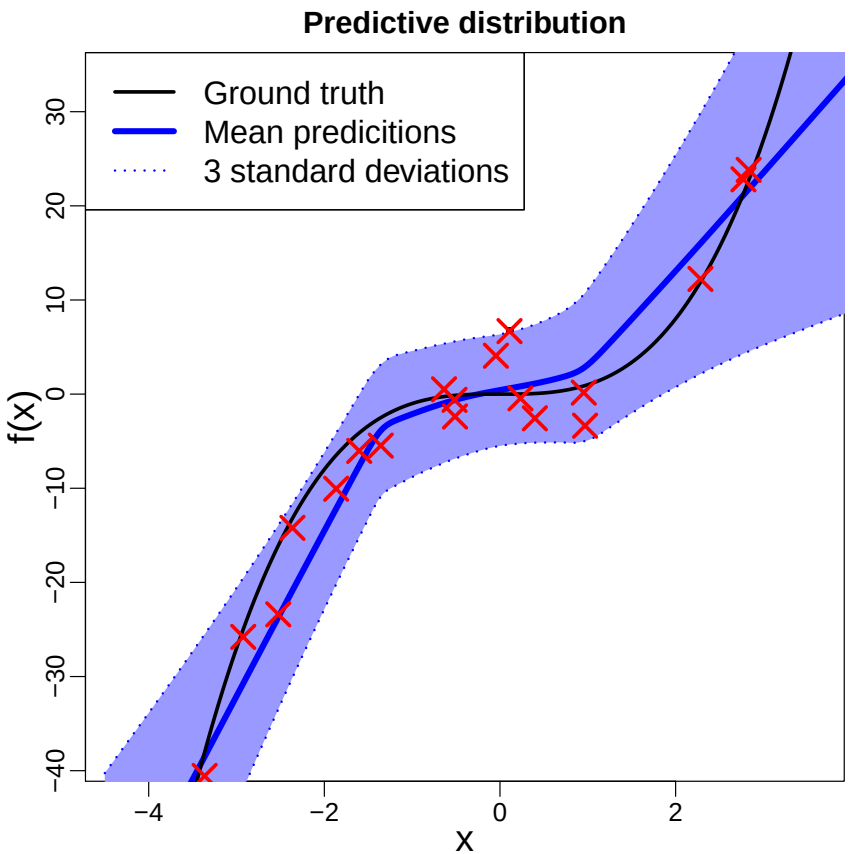


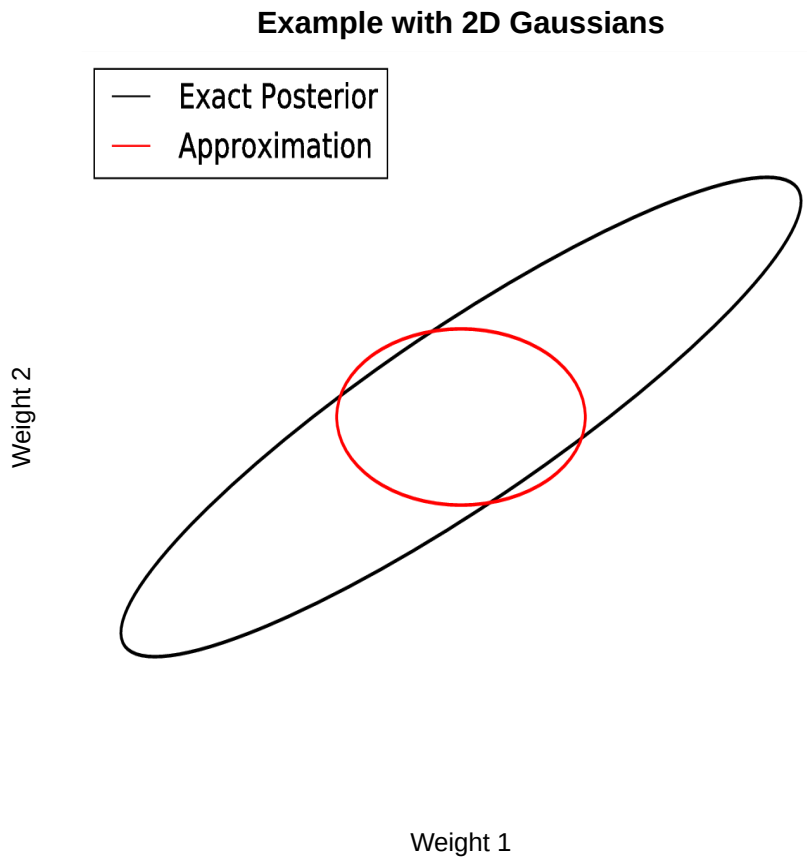




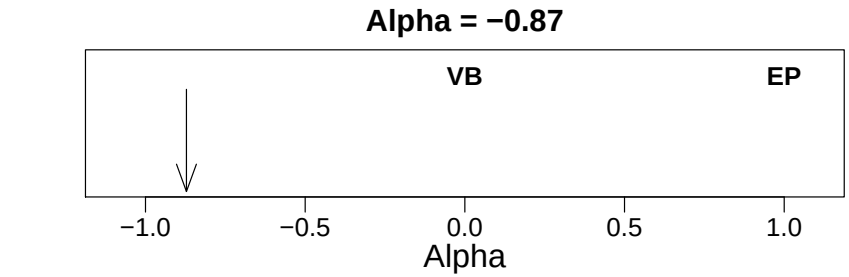
Regression with neural networks and 2D Gaussian

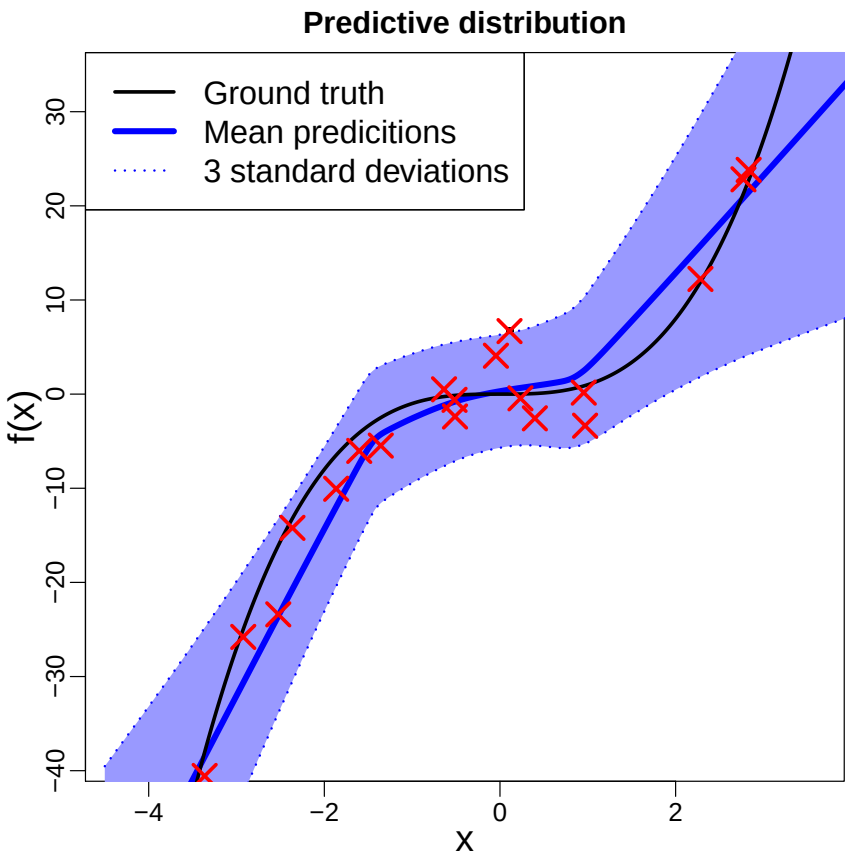


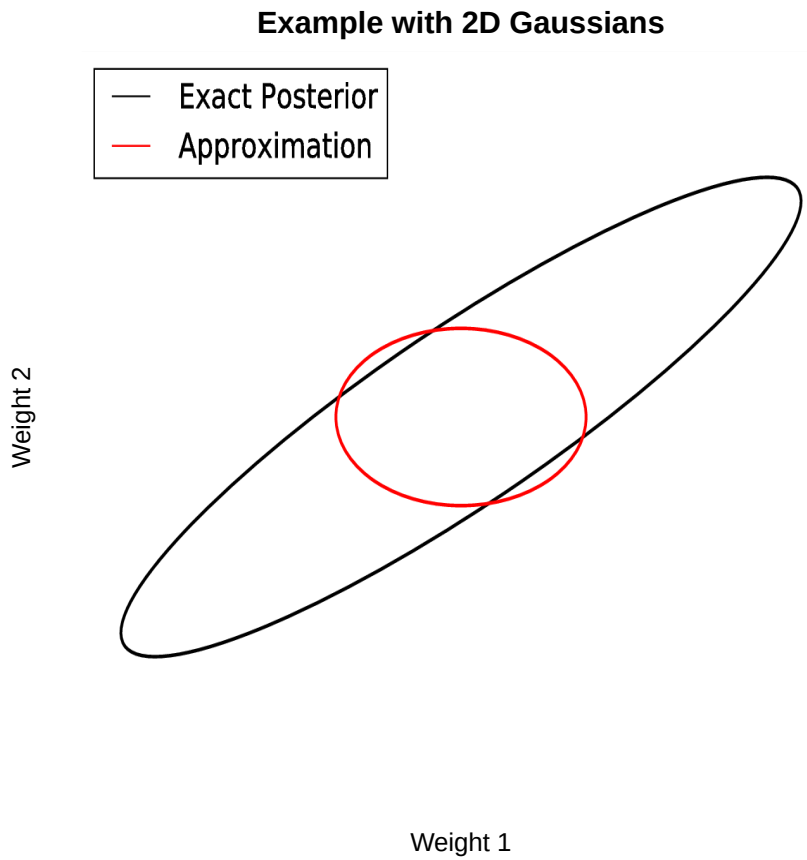




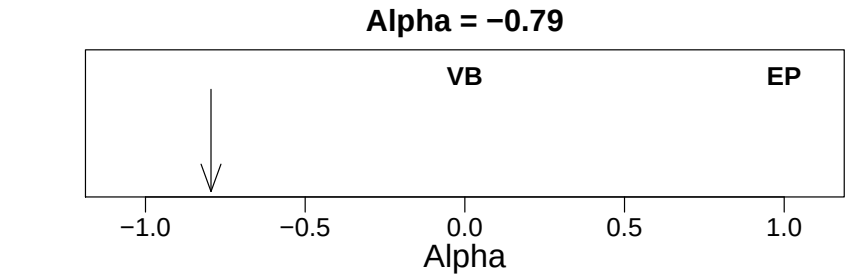
Regression with neural networks and 2D Gaussian

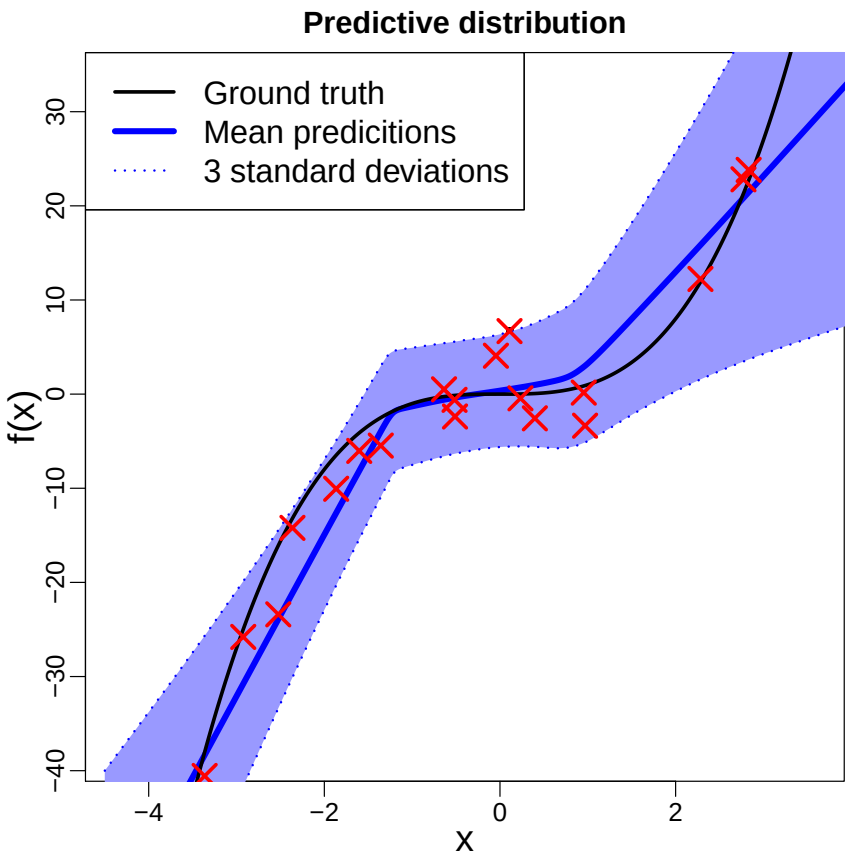


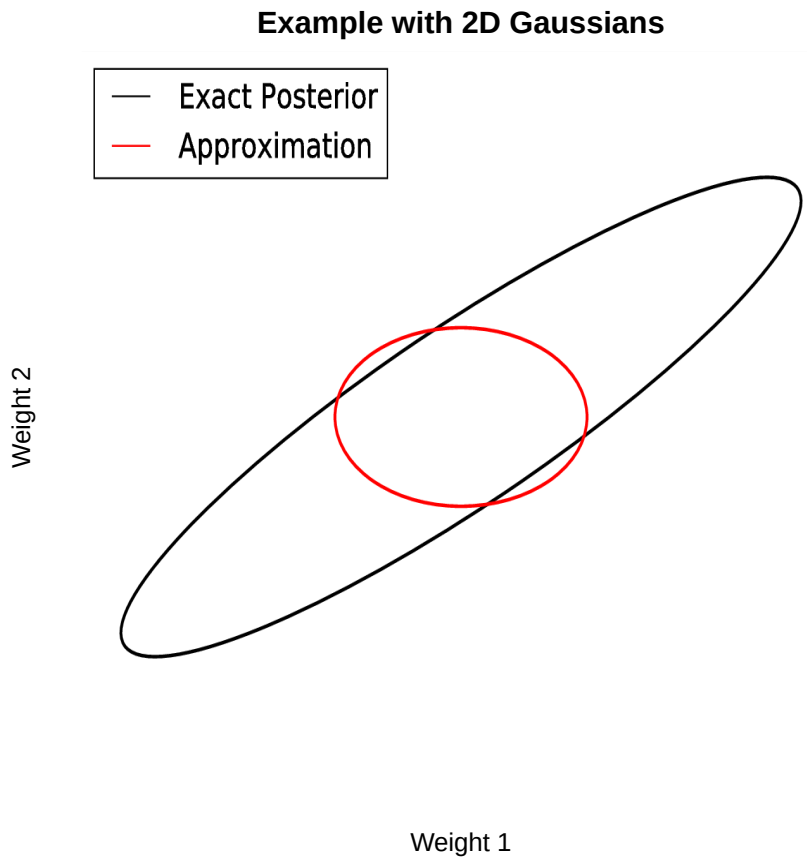




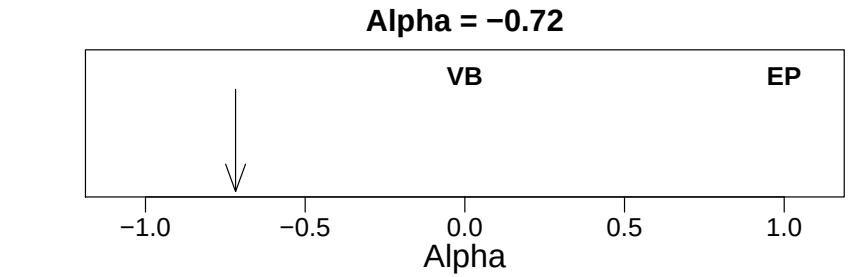
Regression with neural networks and 2D Gaussian

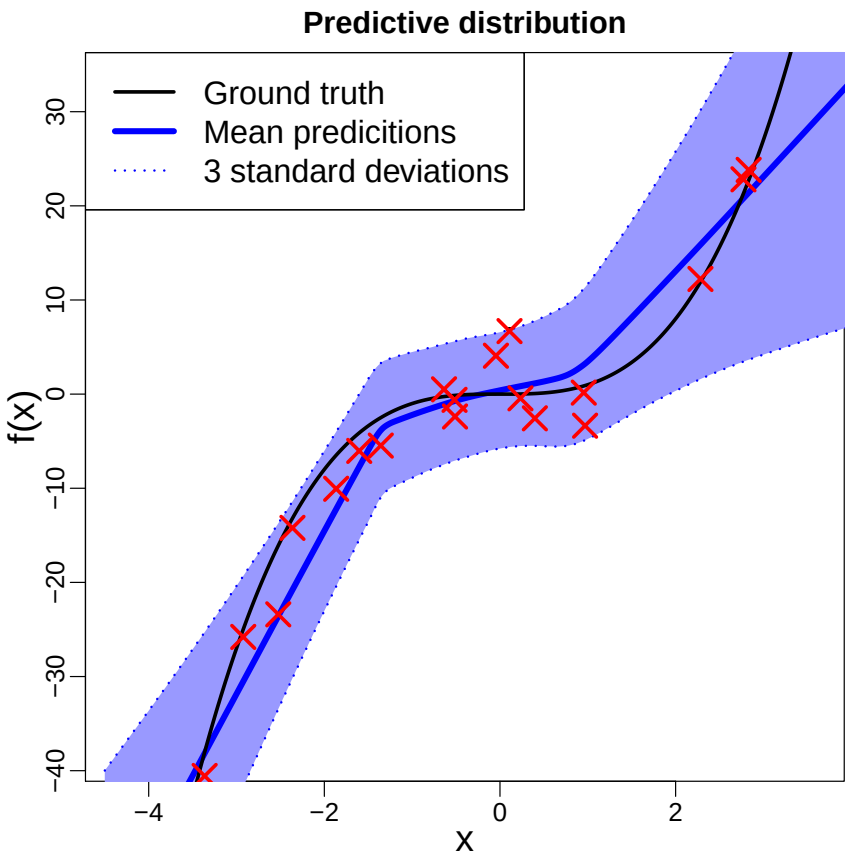


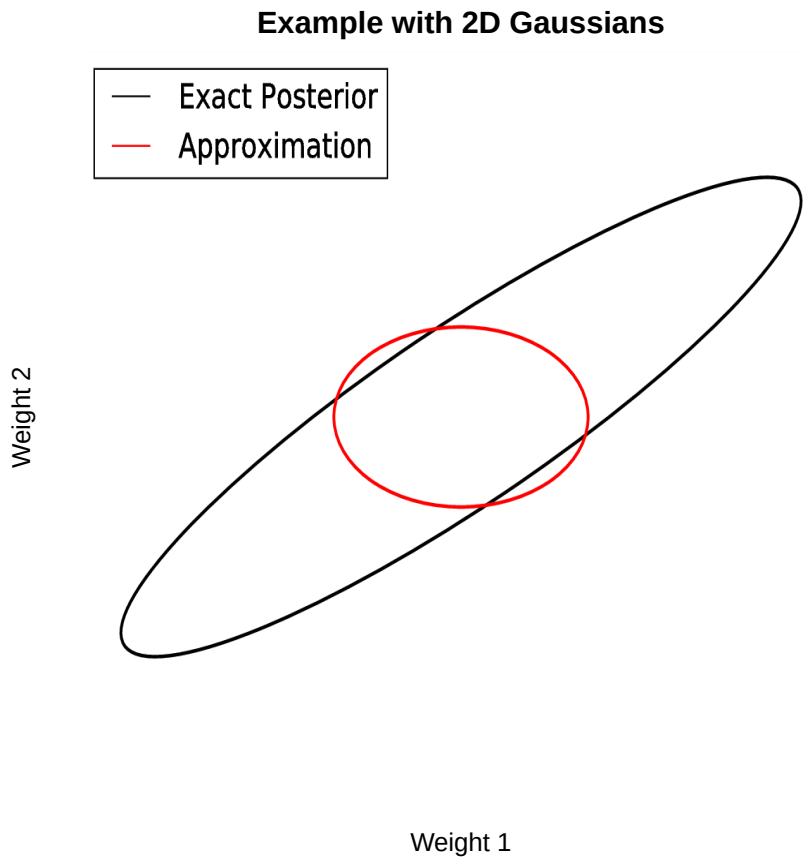




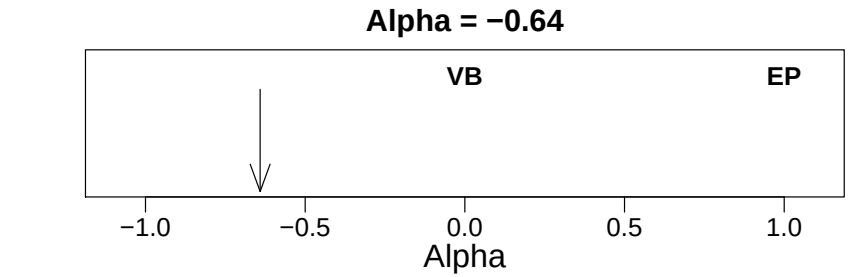
Regression with neural networks and 2D Gaussian

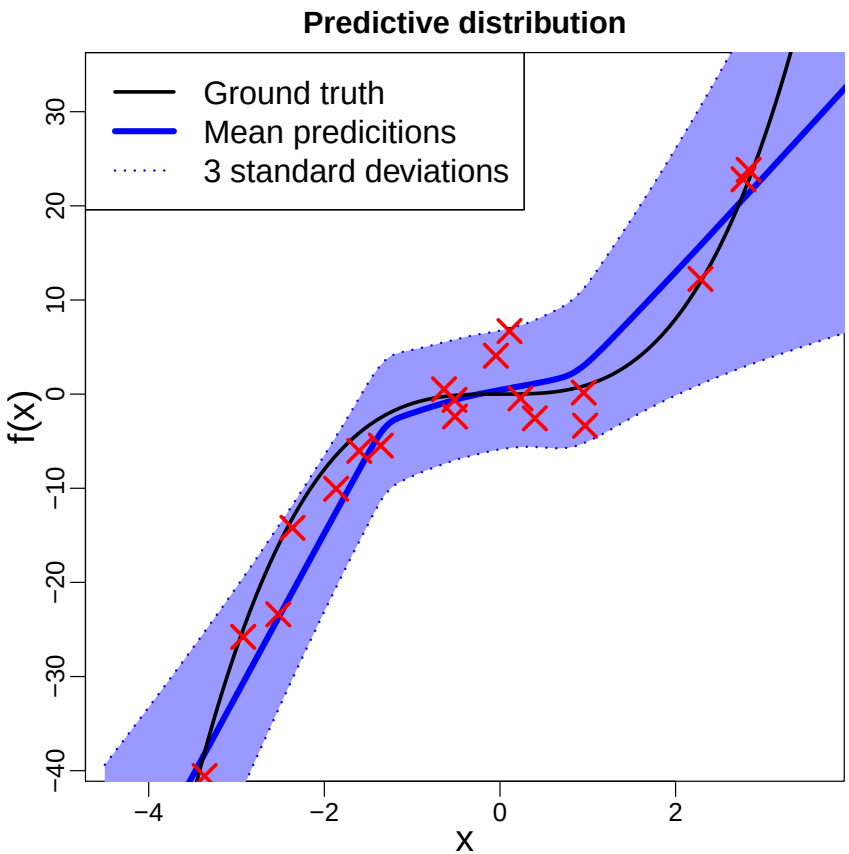


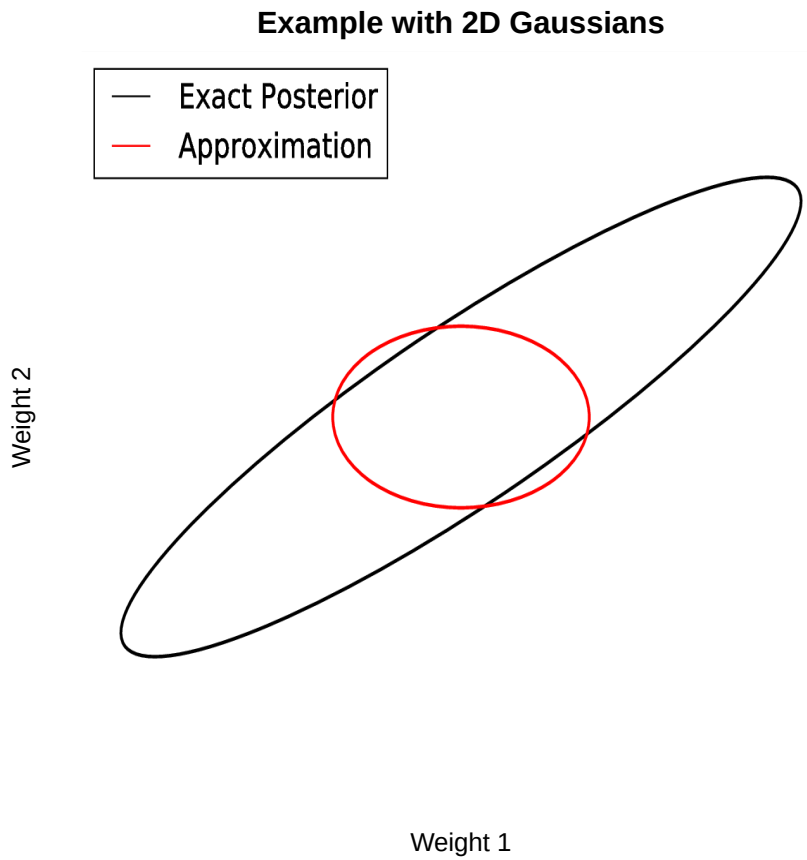




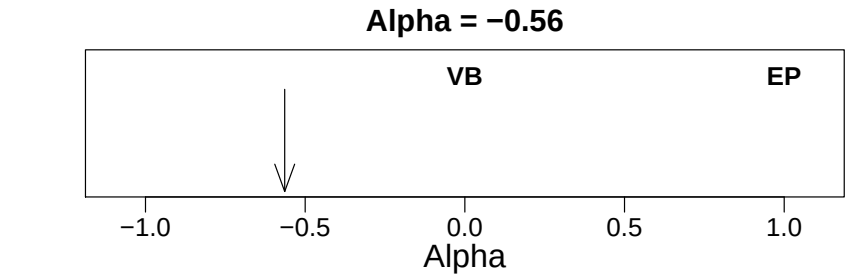
Regression with neural networks and 2D Gaussian

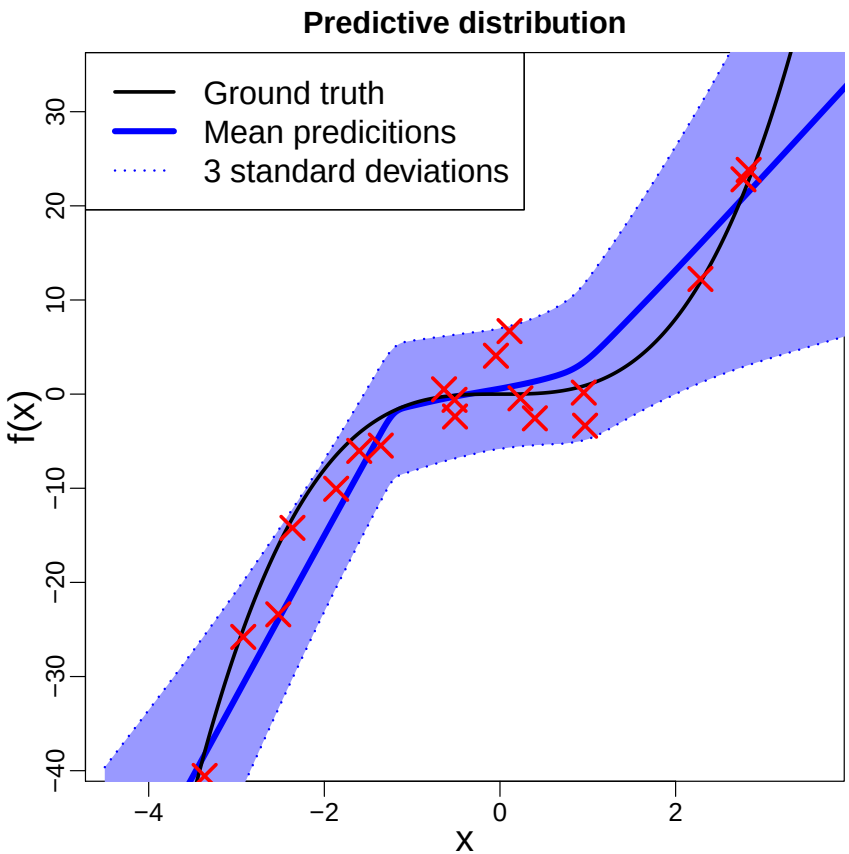


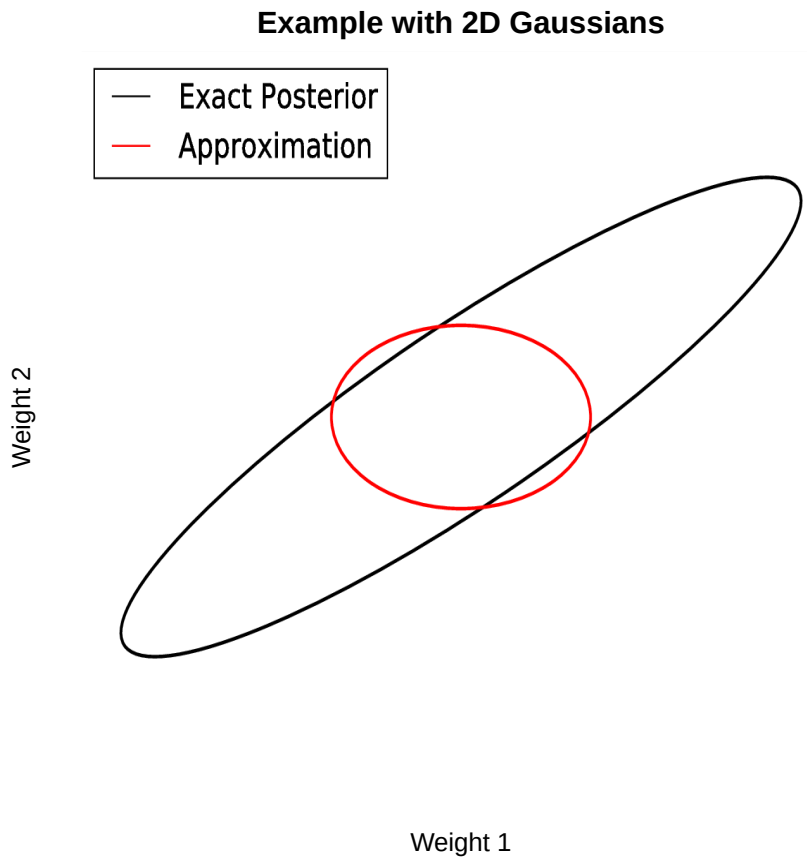




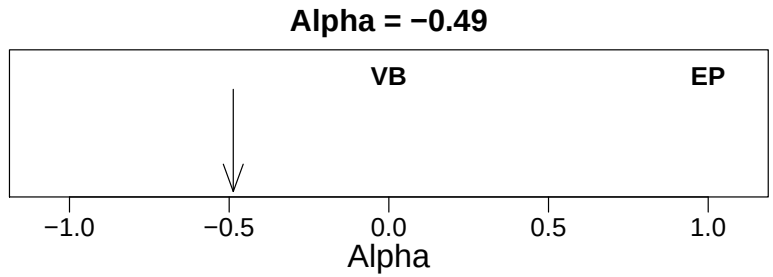
Regression with neural networks and 2D Gaussian

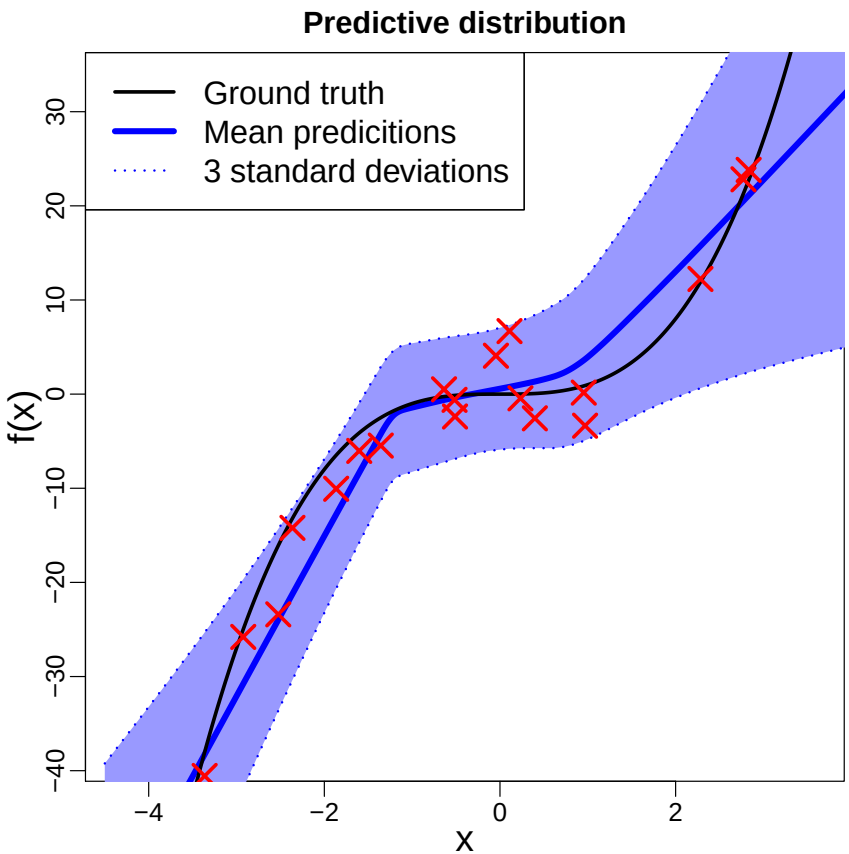


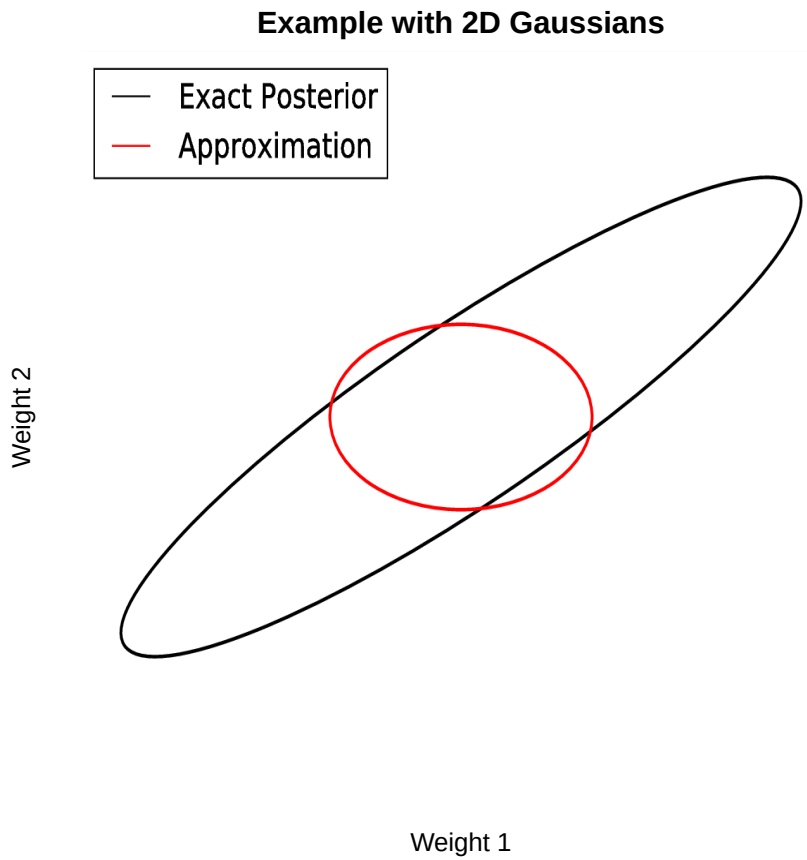




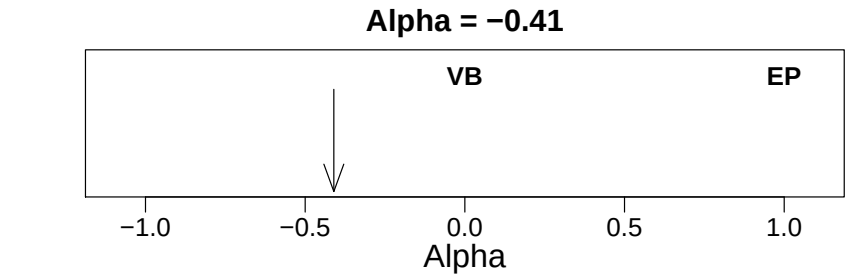
Regression with neural networks and 2D Gaussian

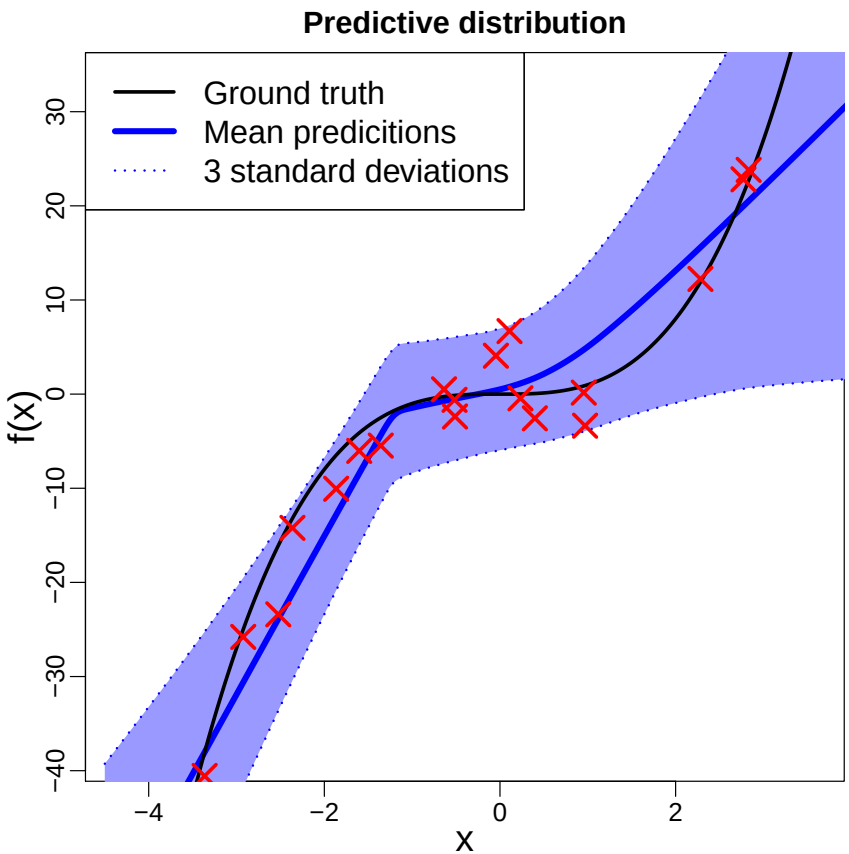


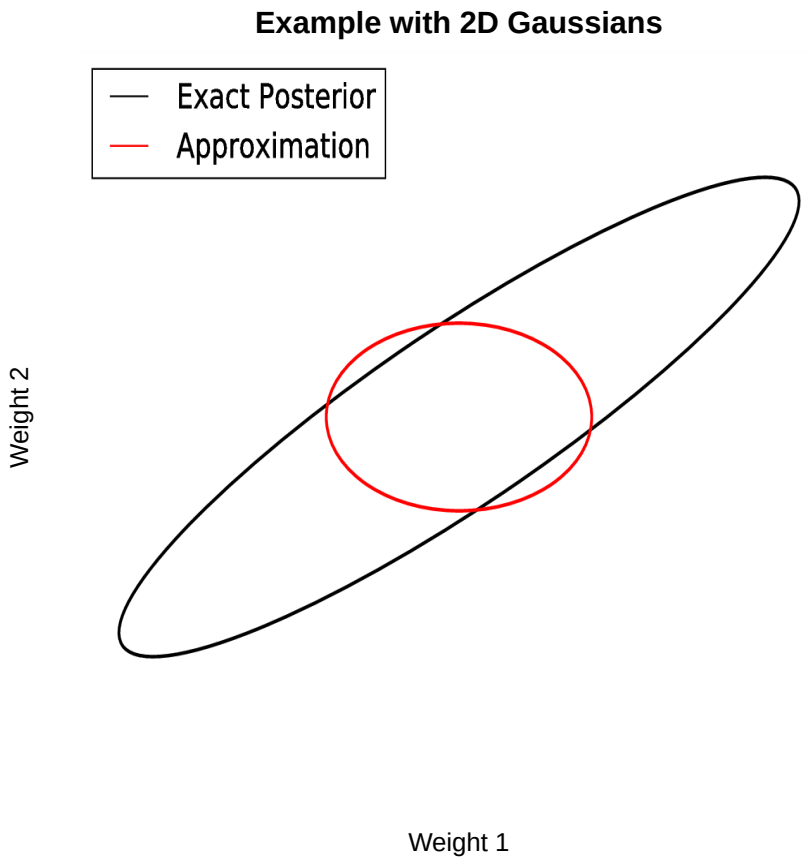




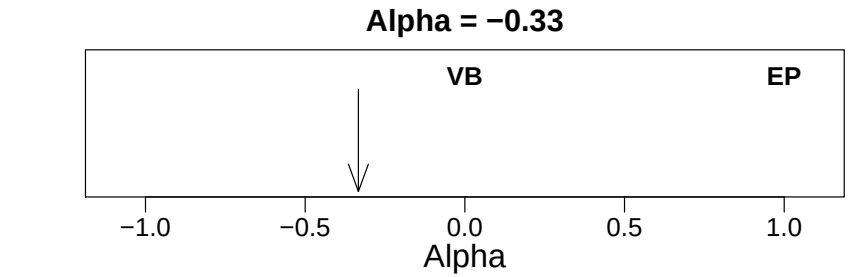
Regression with neural networks and 2D Gaussian

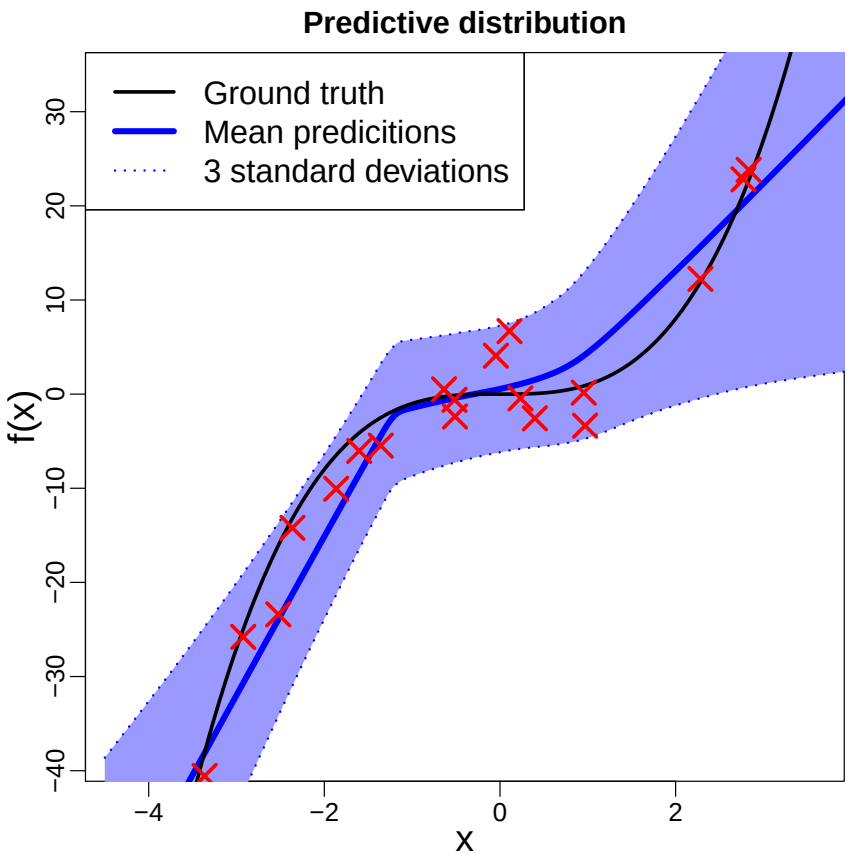


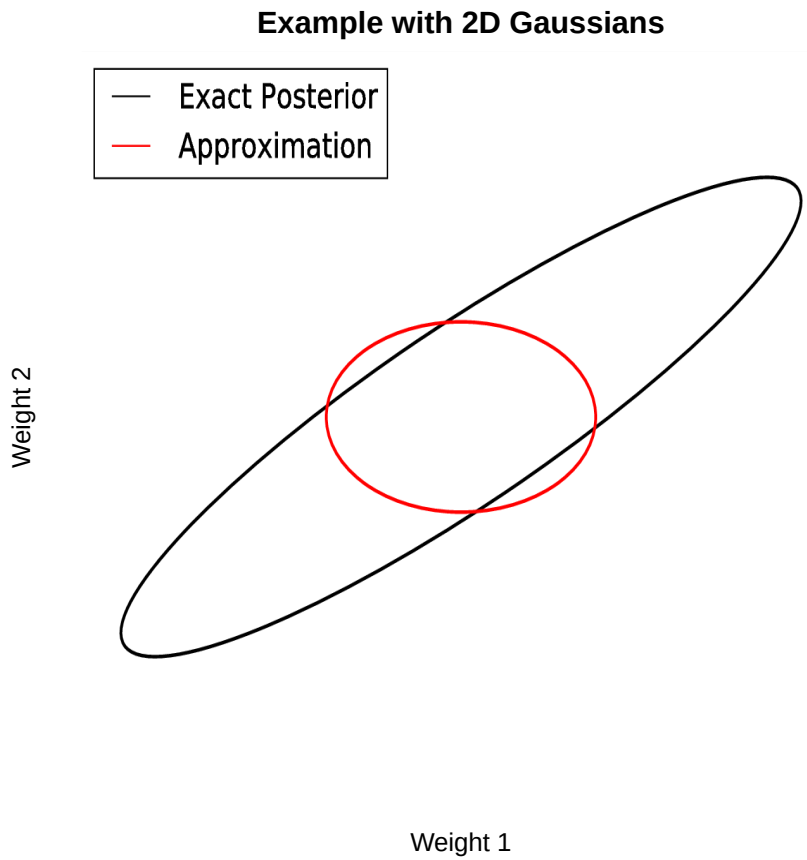




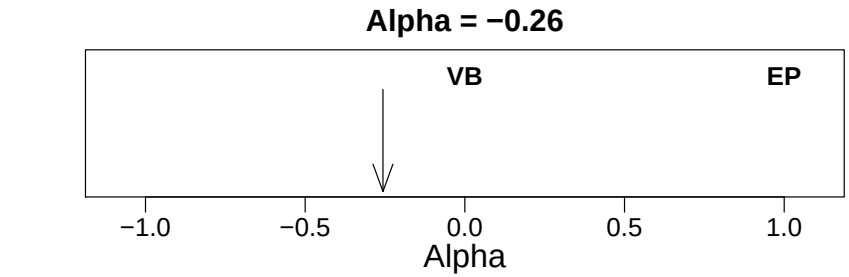
Regression with neural networks and 2D Gaussian

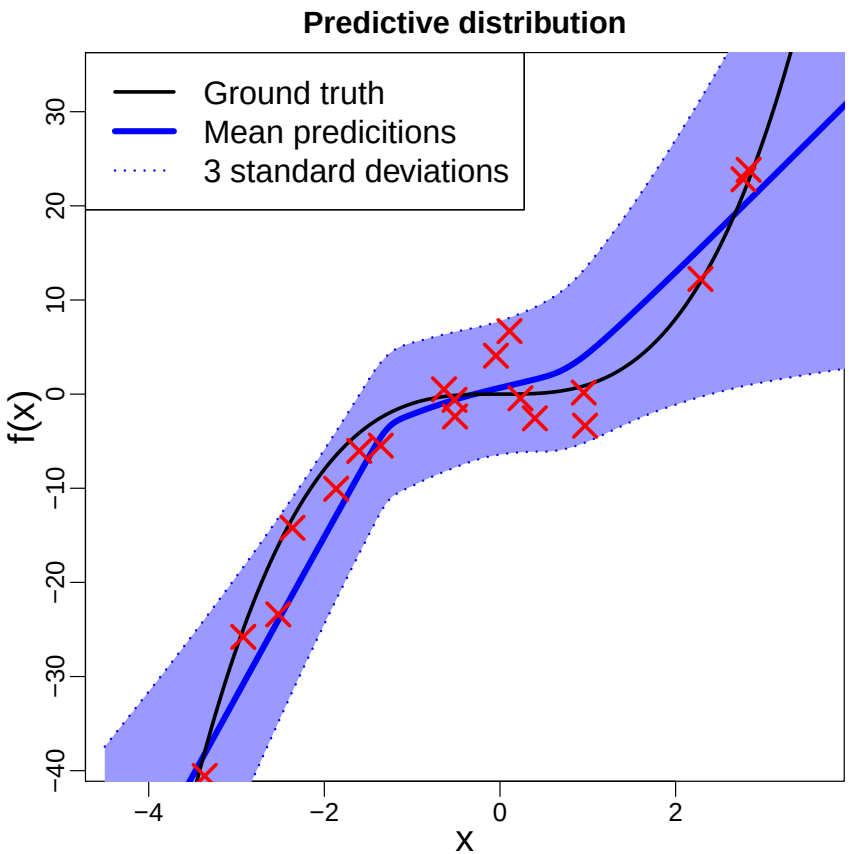


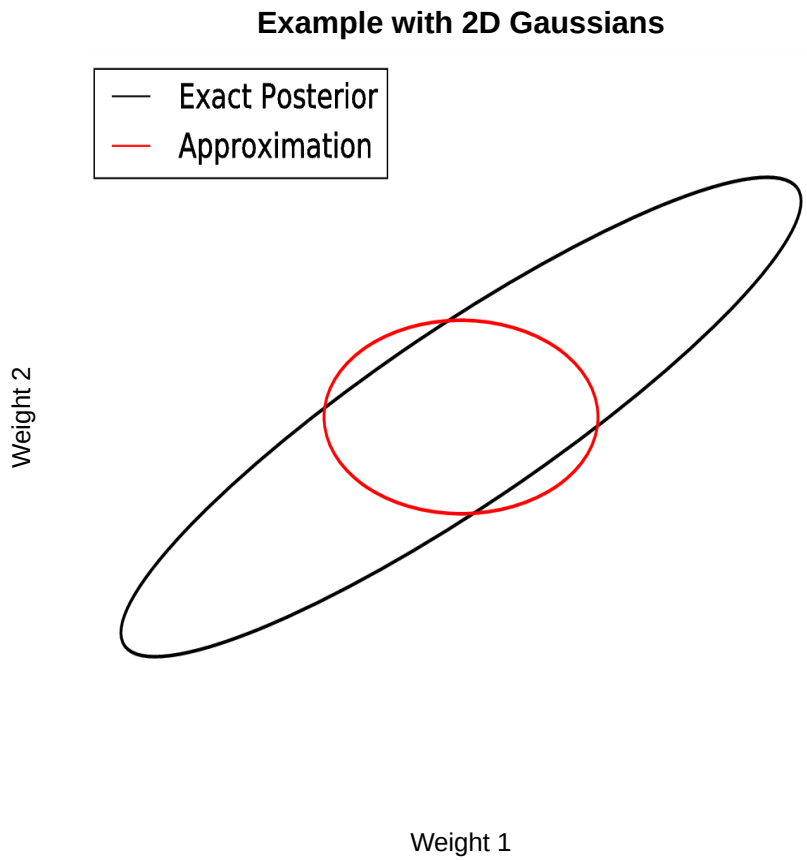




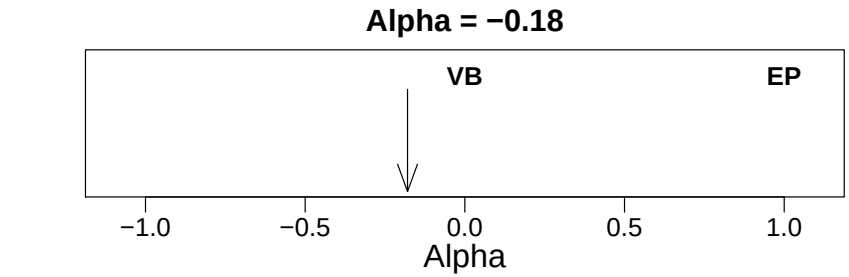
Regression with neural networks and 2D Gaussian

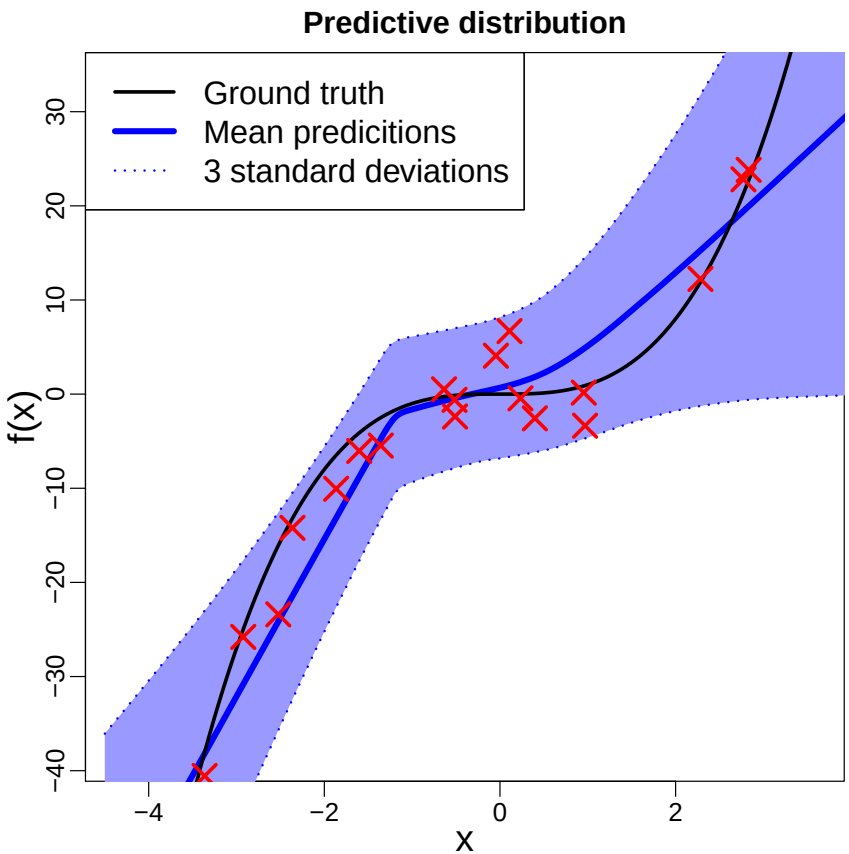


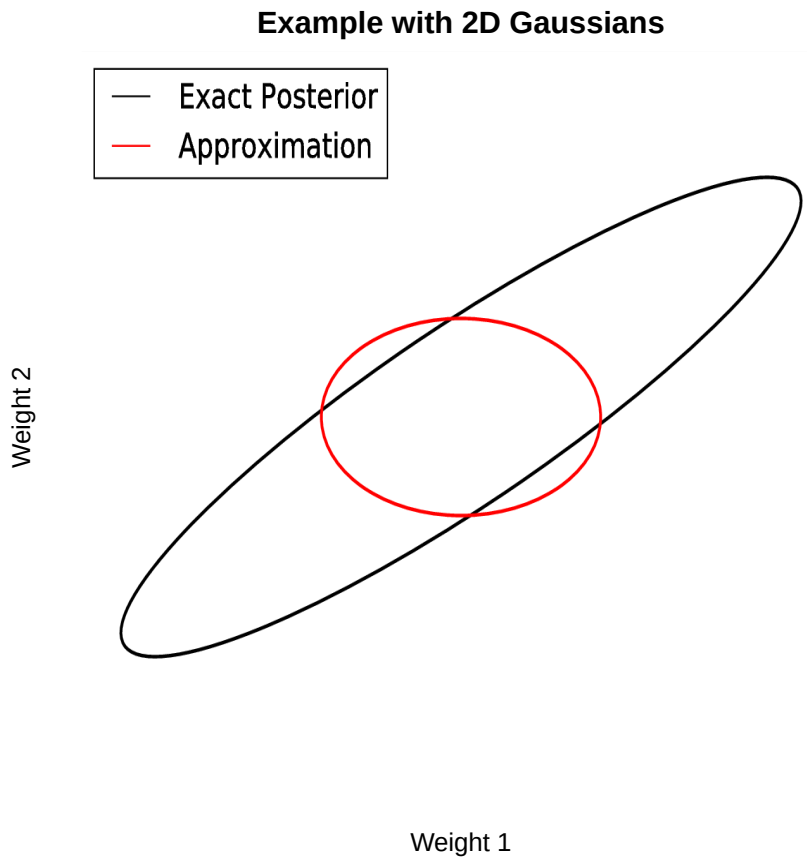




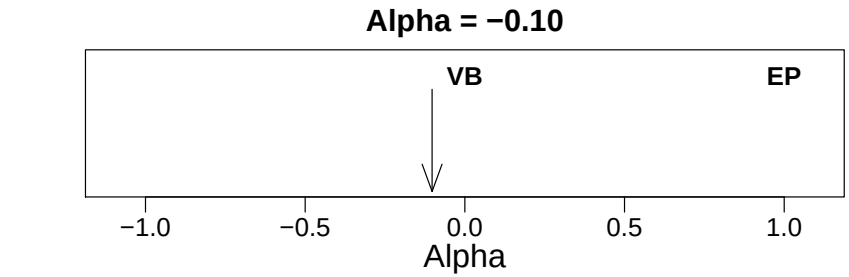
Regression with neural networks and 2D Gaussian

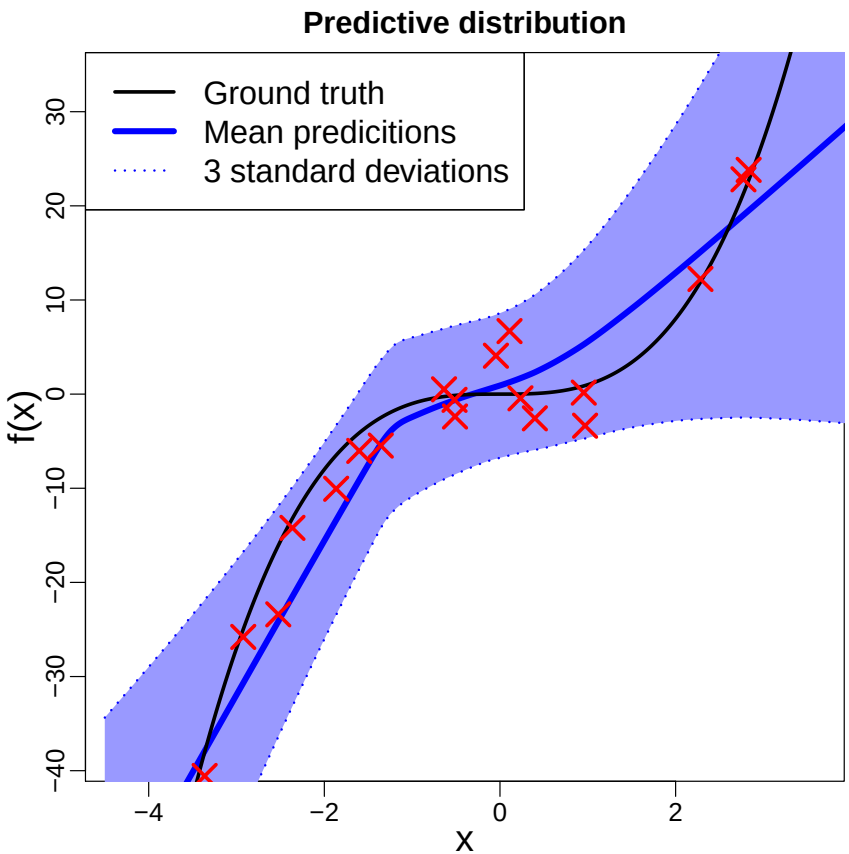


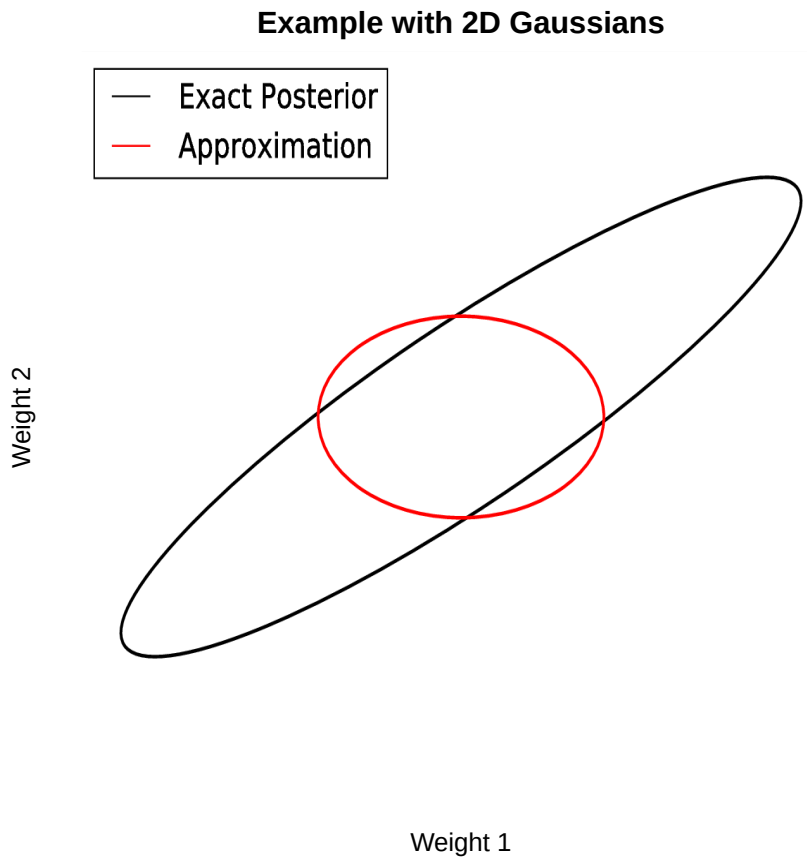




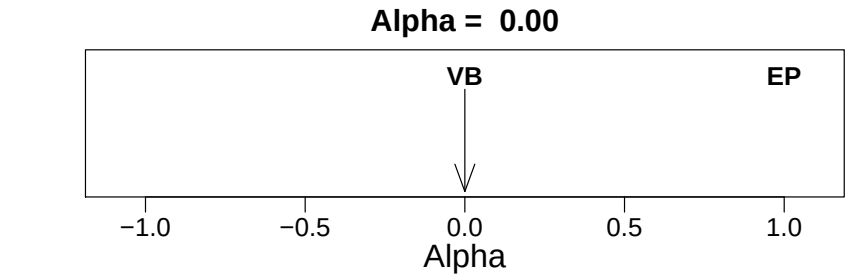
Regression with neural networks and 2D Gaussian

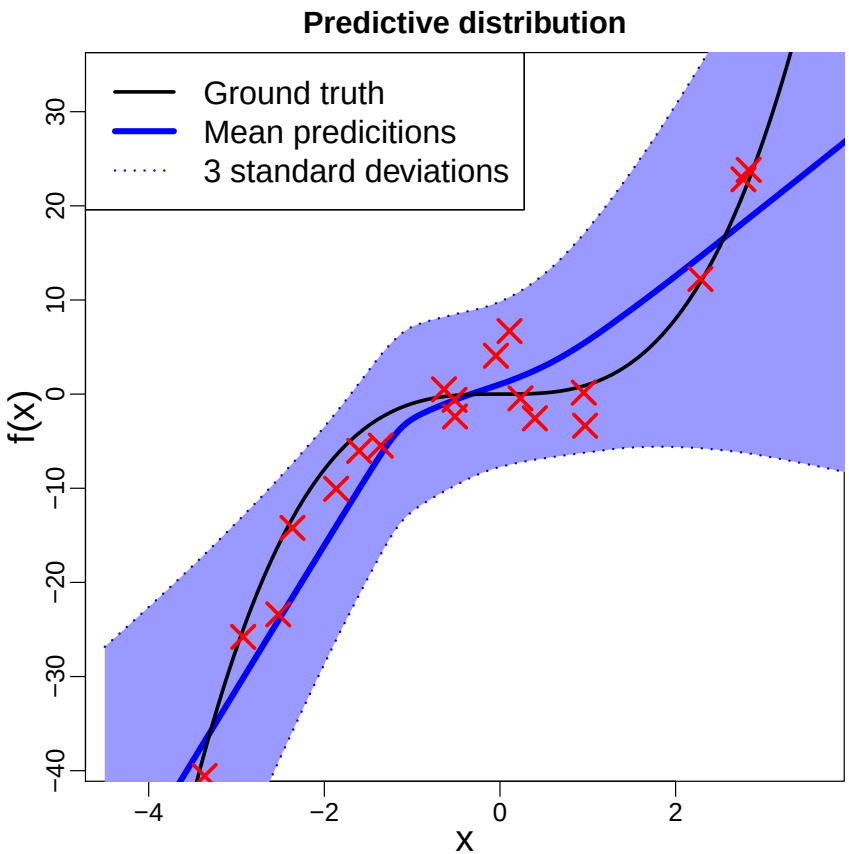


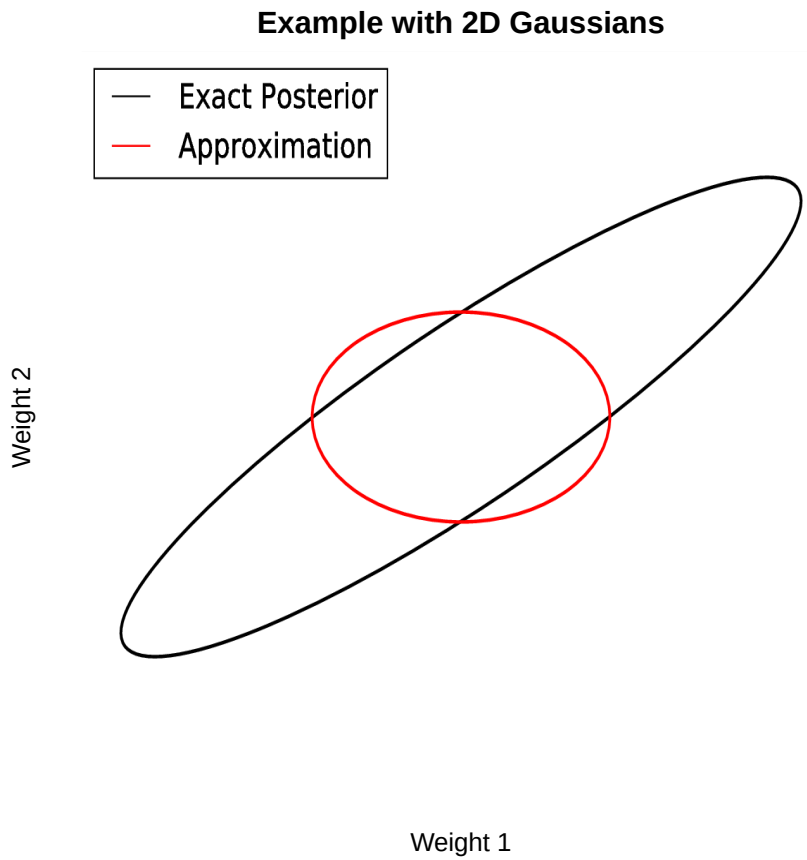




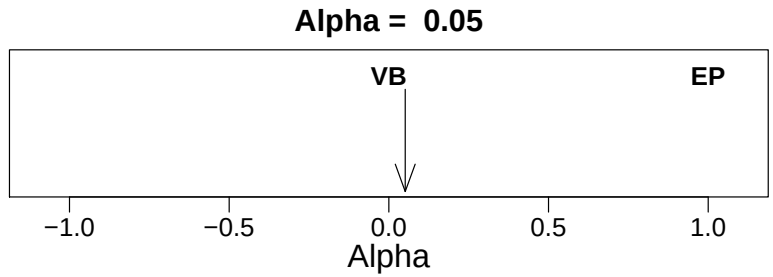
Regression with neural networks and 2D Gaussian

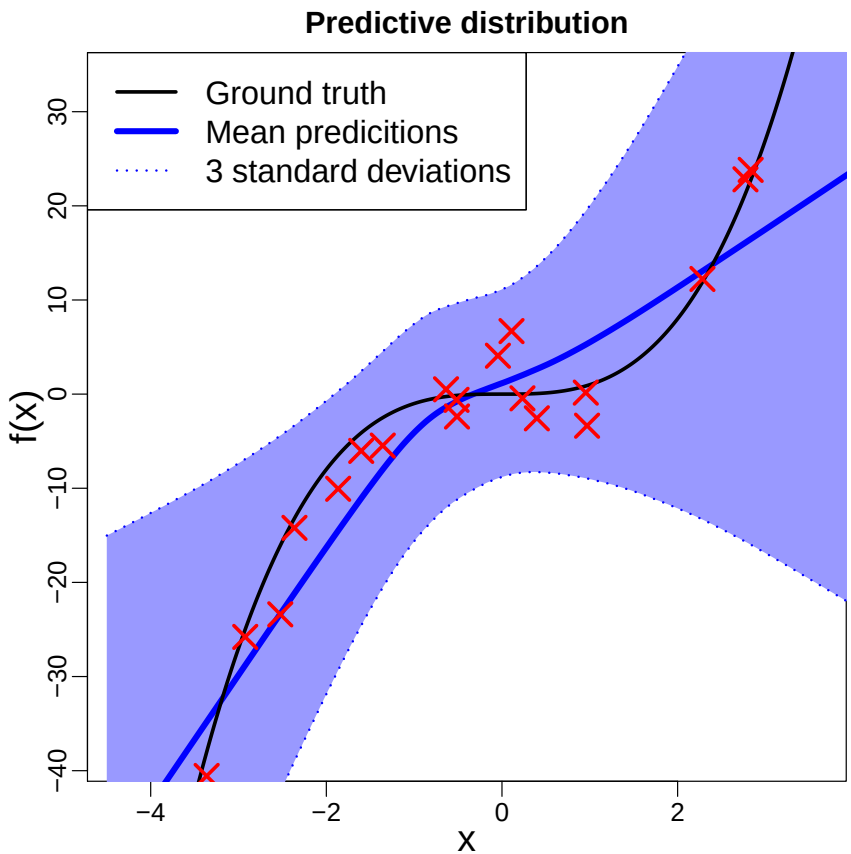


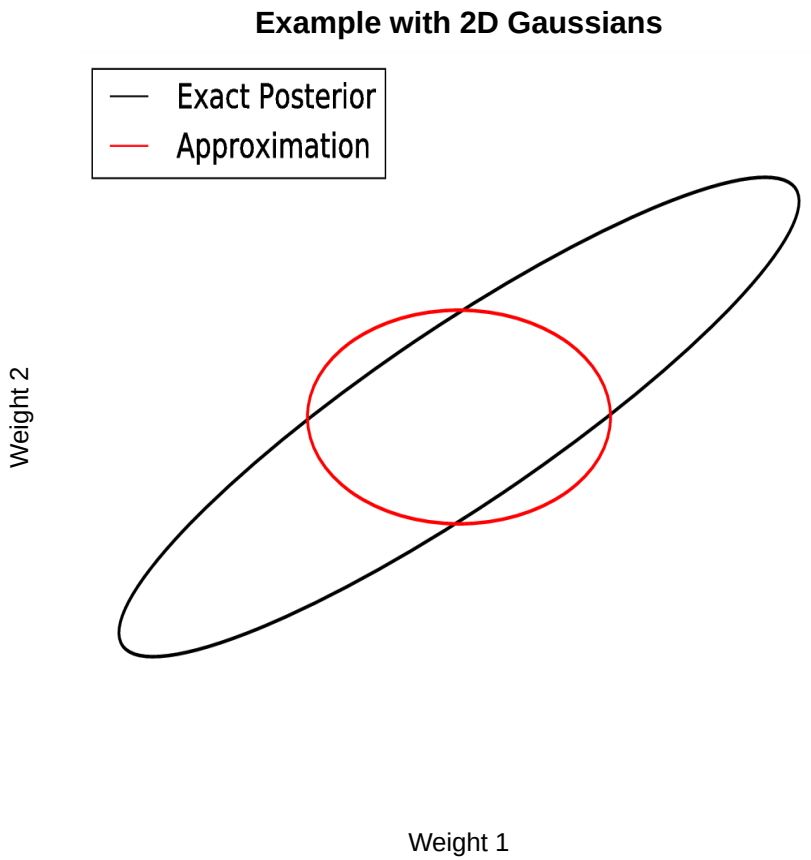




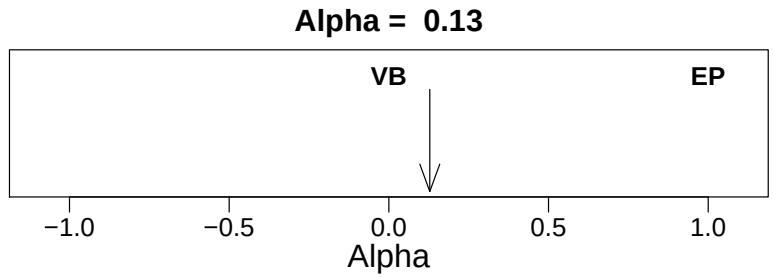
Regression with neural networks and 2D Gaussian

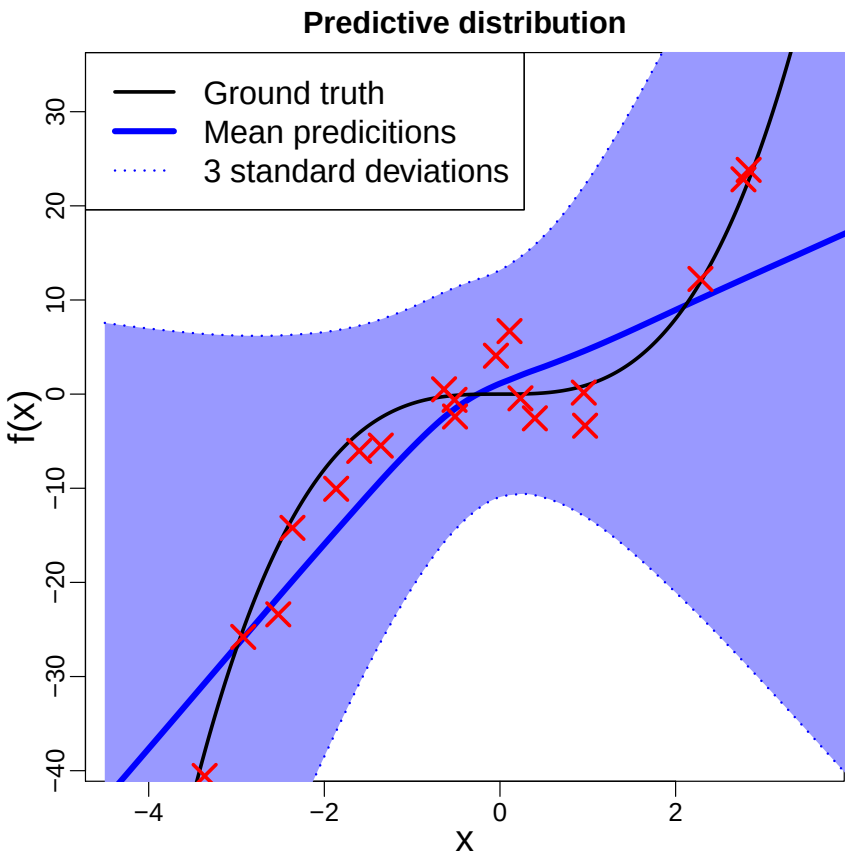


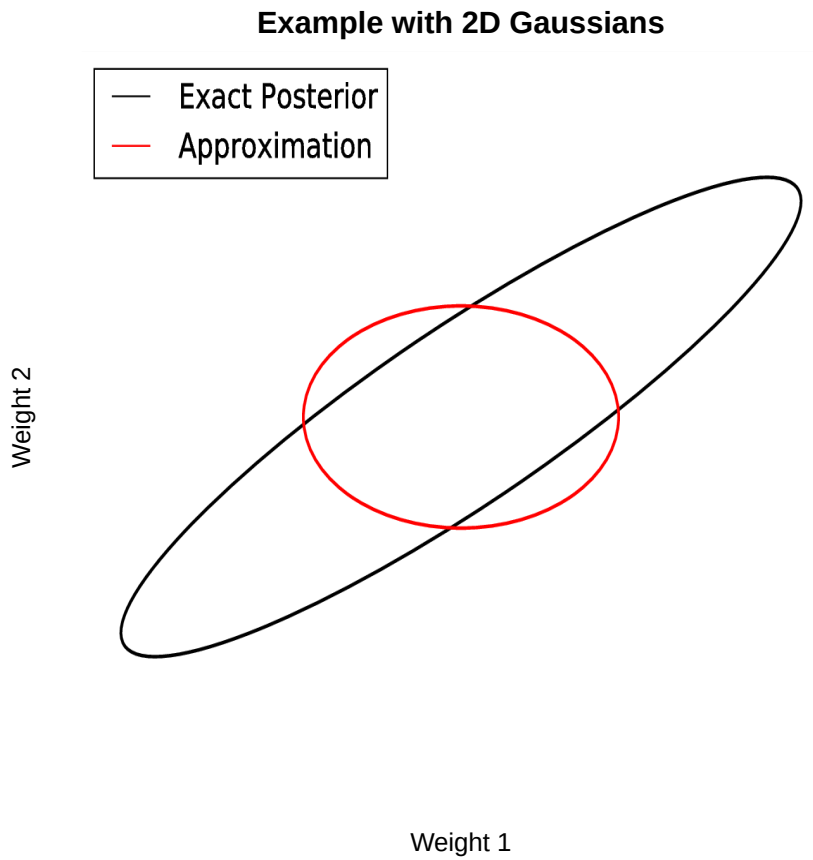




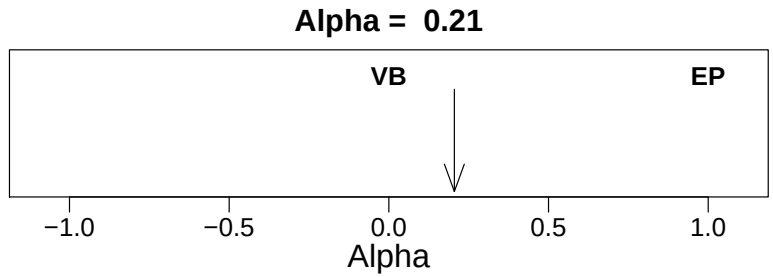
Regression with neural networks and 2D Gaussian

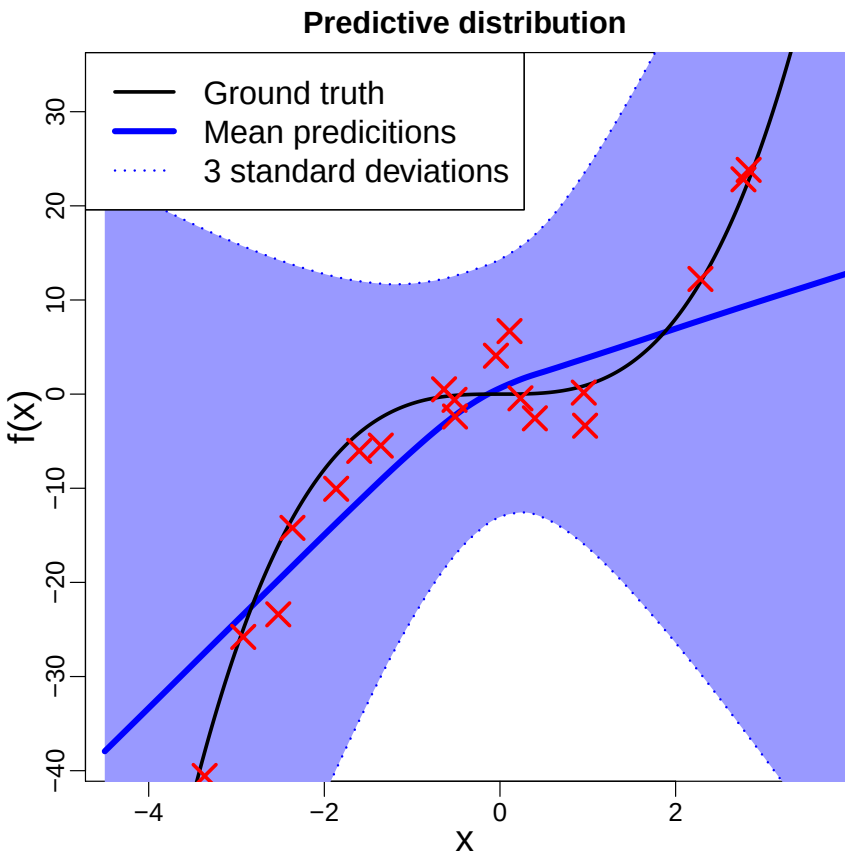


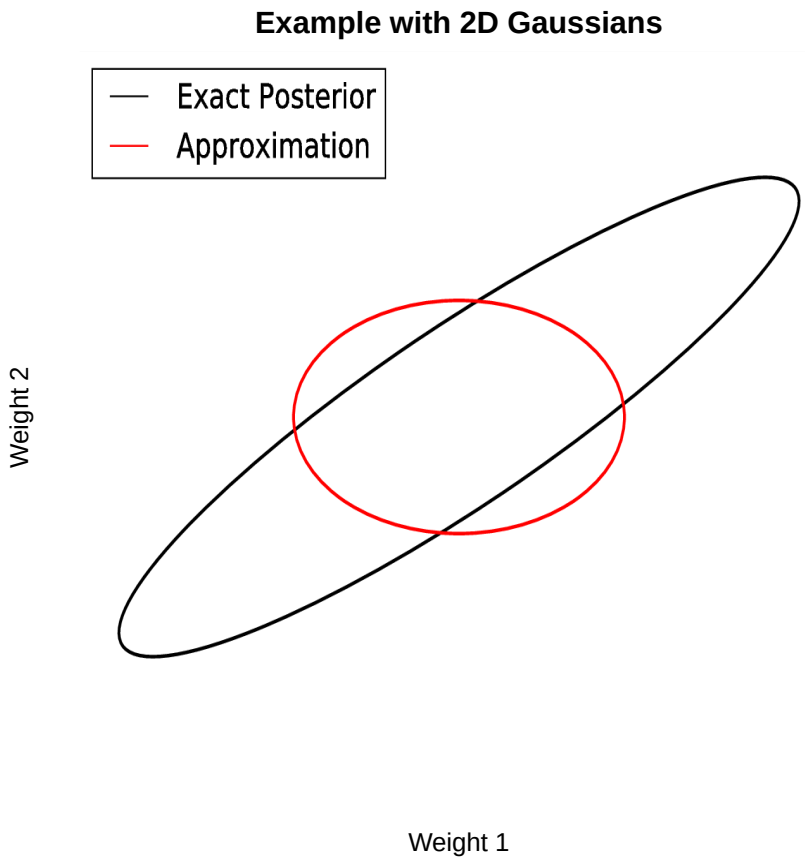




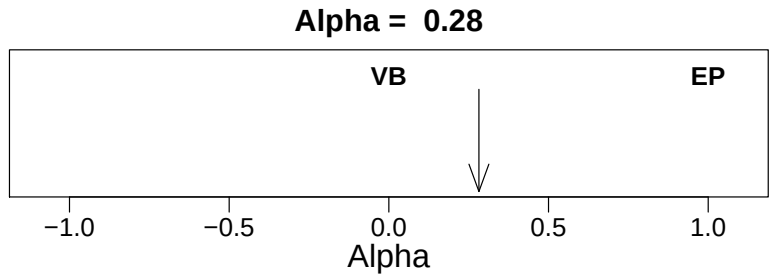
Regression with neural networks and 2D Gaussian

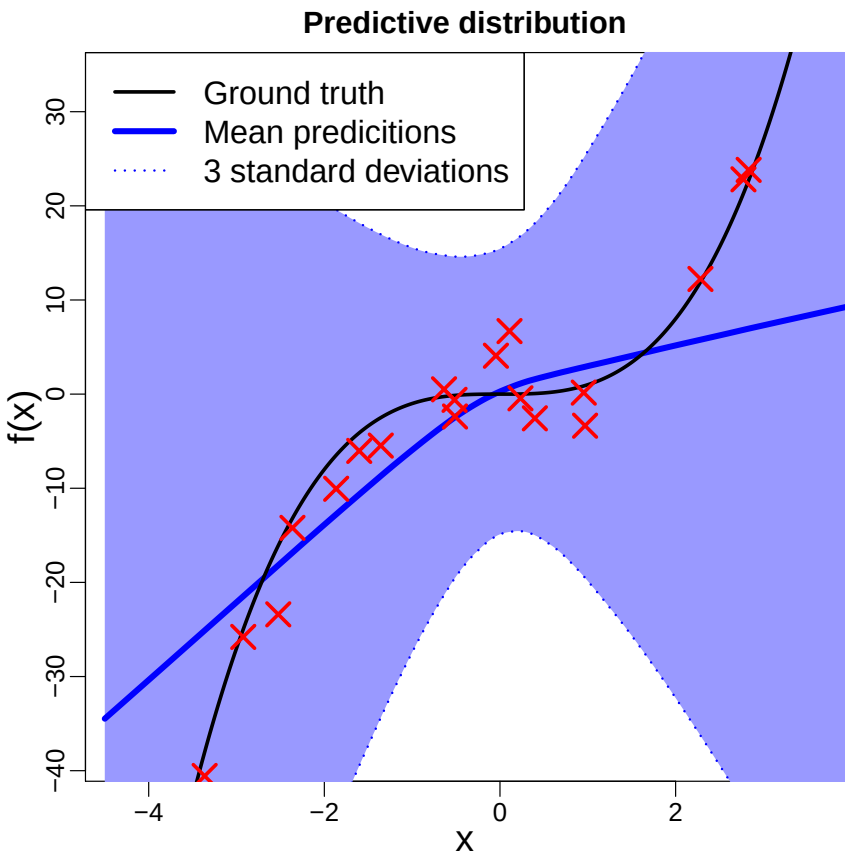


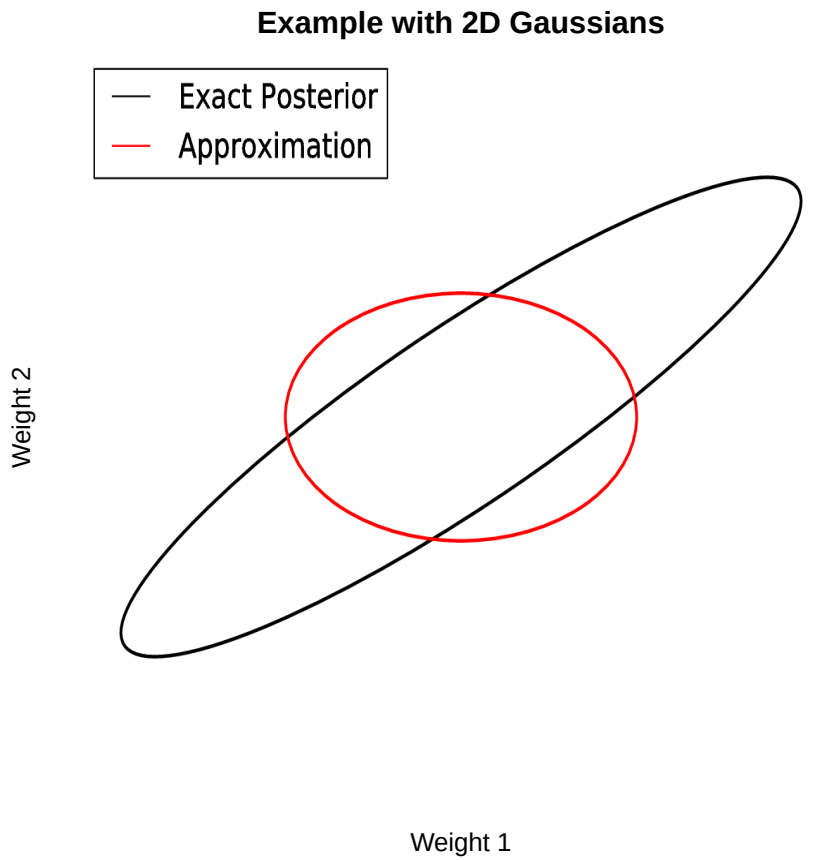




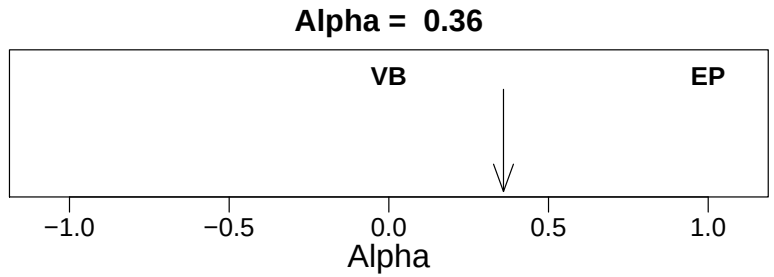
Regression with neural networks and 2D Gaussian

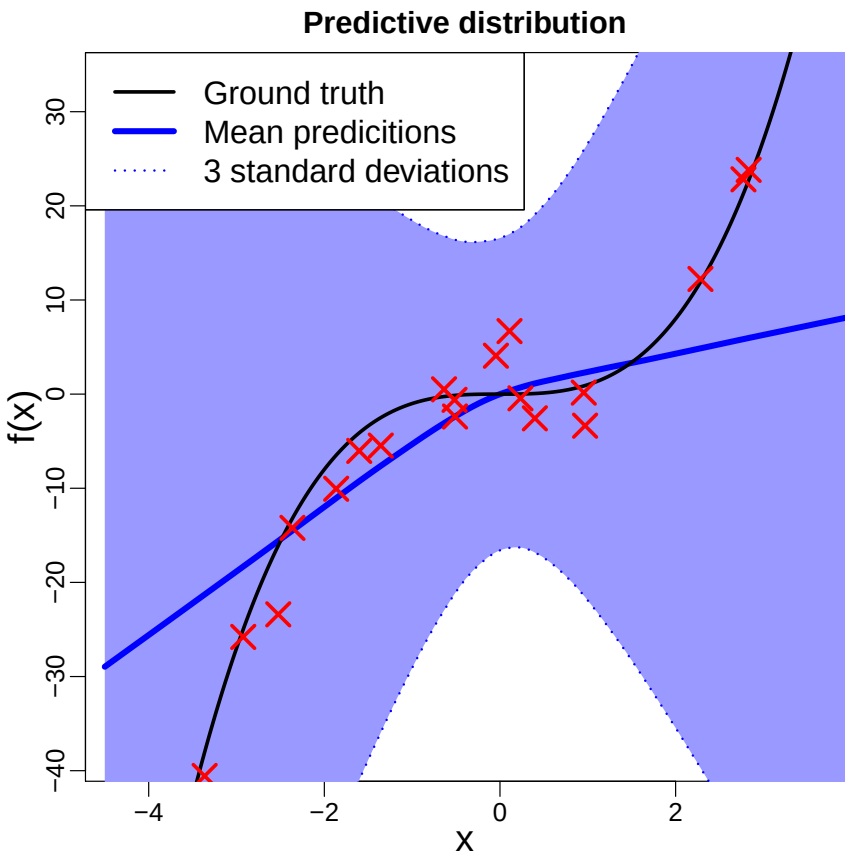


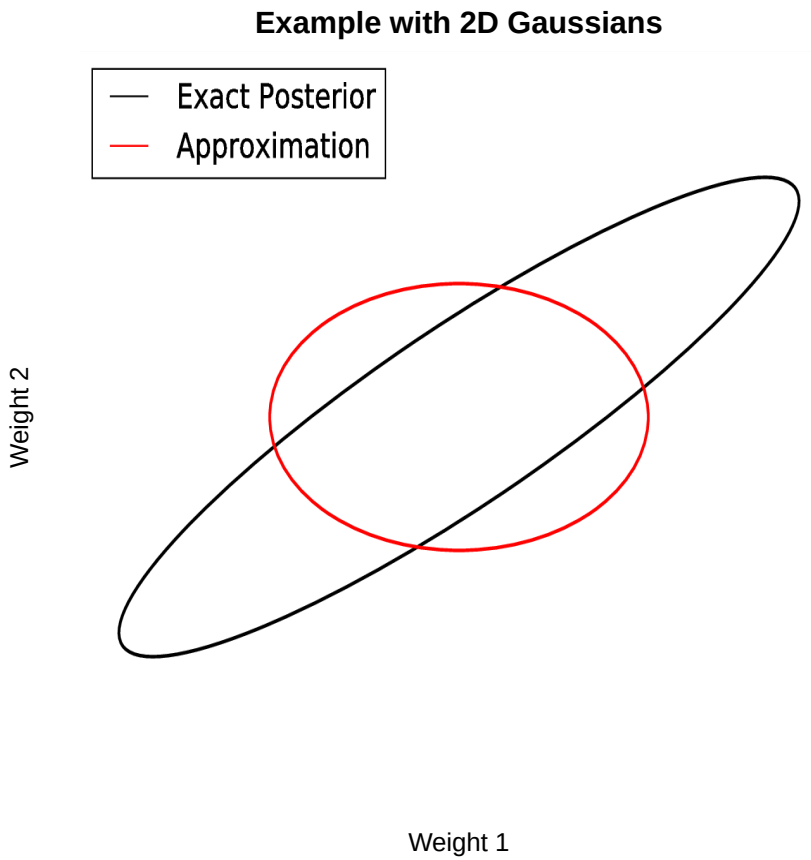




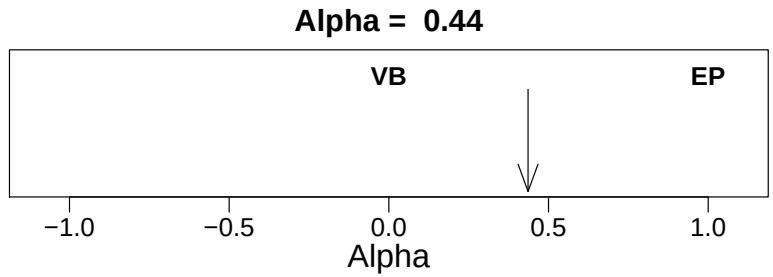
Regression with neural networks and 2D Gaussian

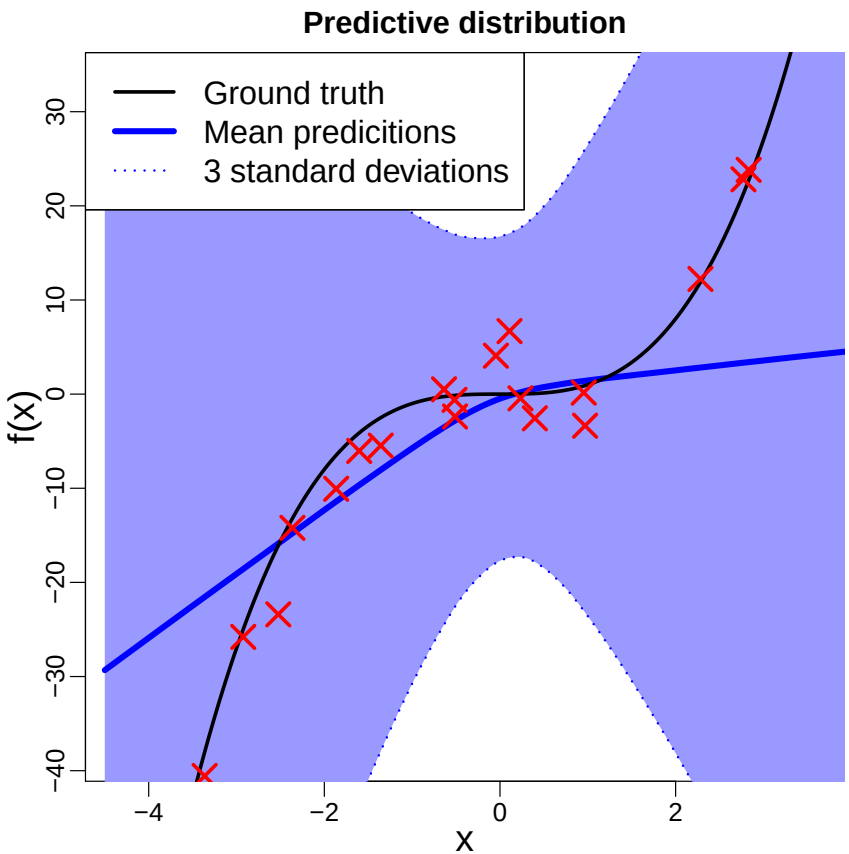


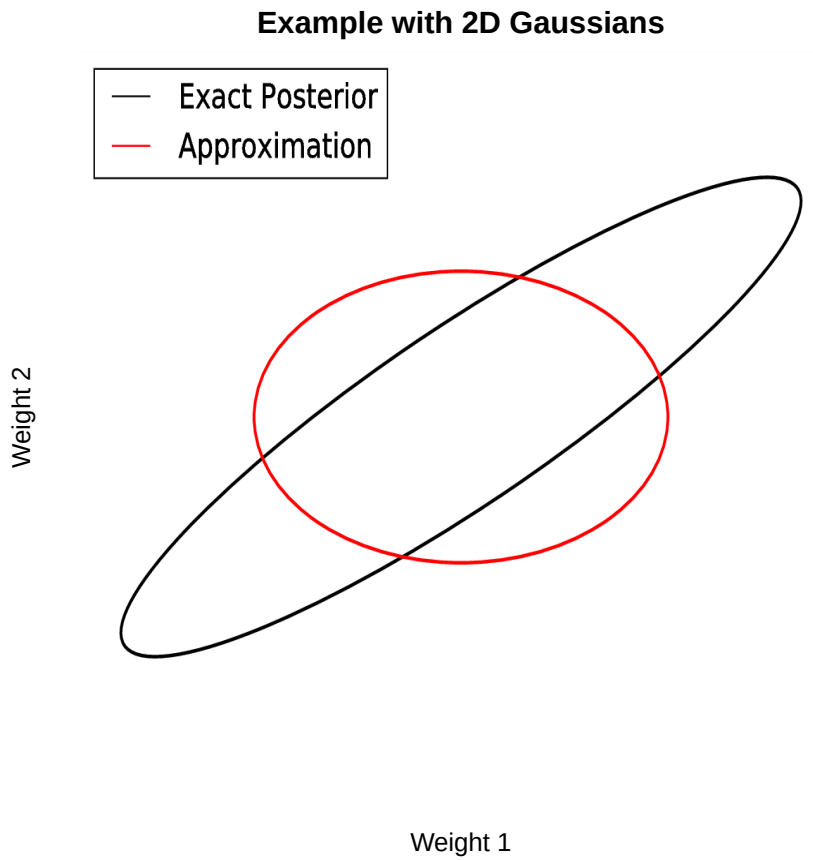




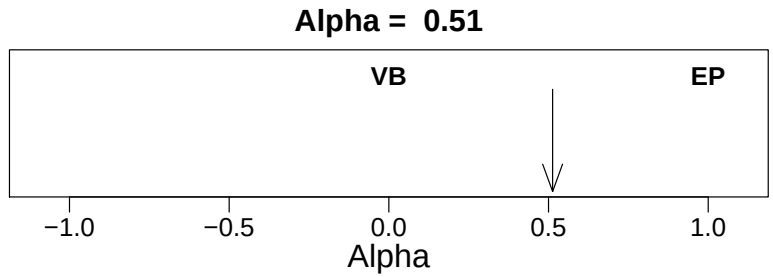
Regression with neural networks and 2D Gaussian

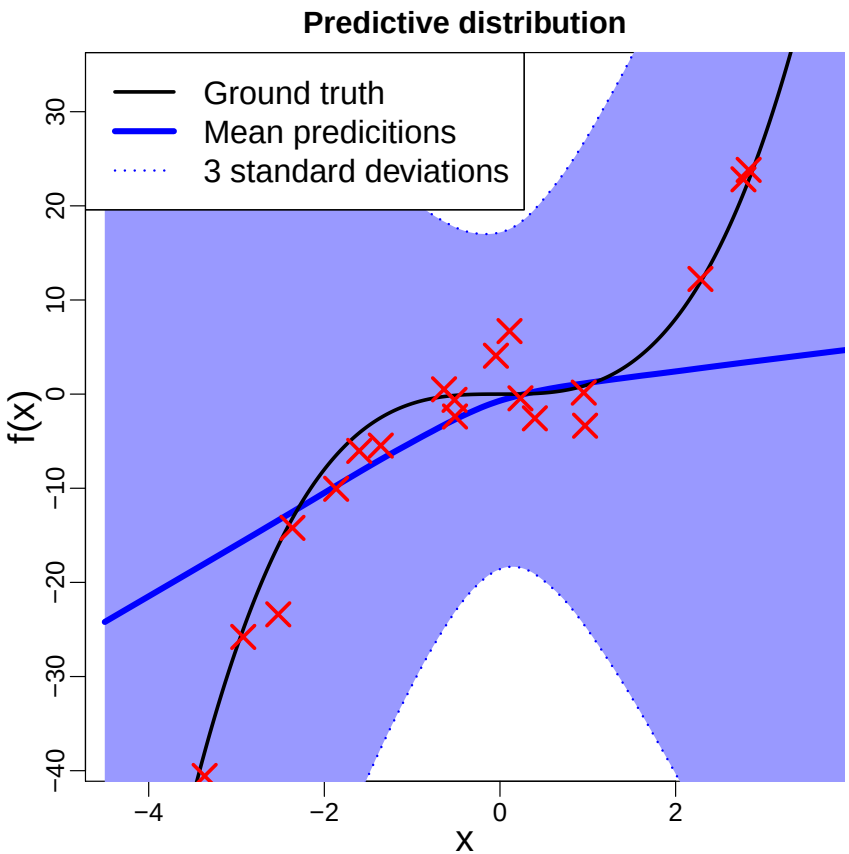


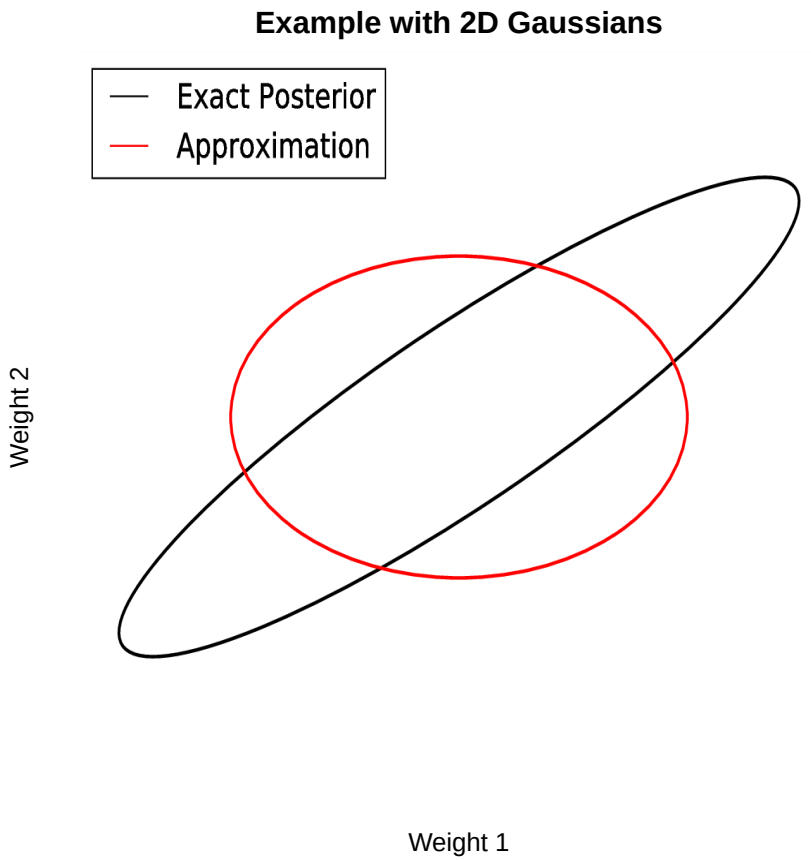




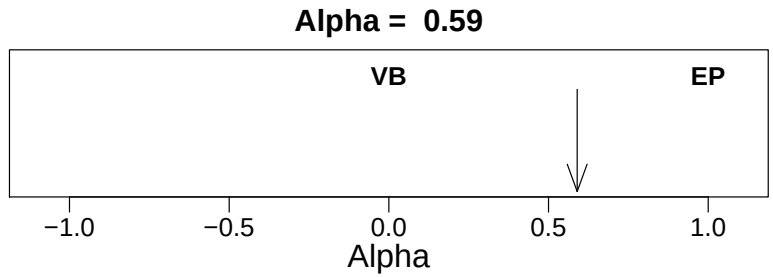
Regression with neural networks and 2D Gaussian

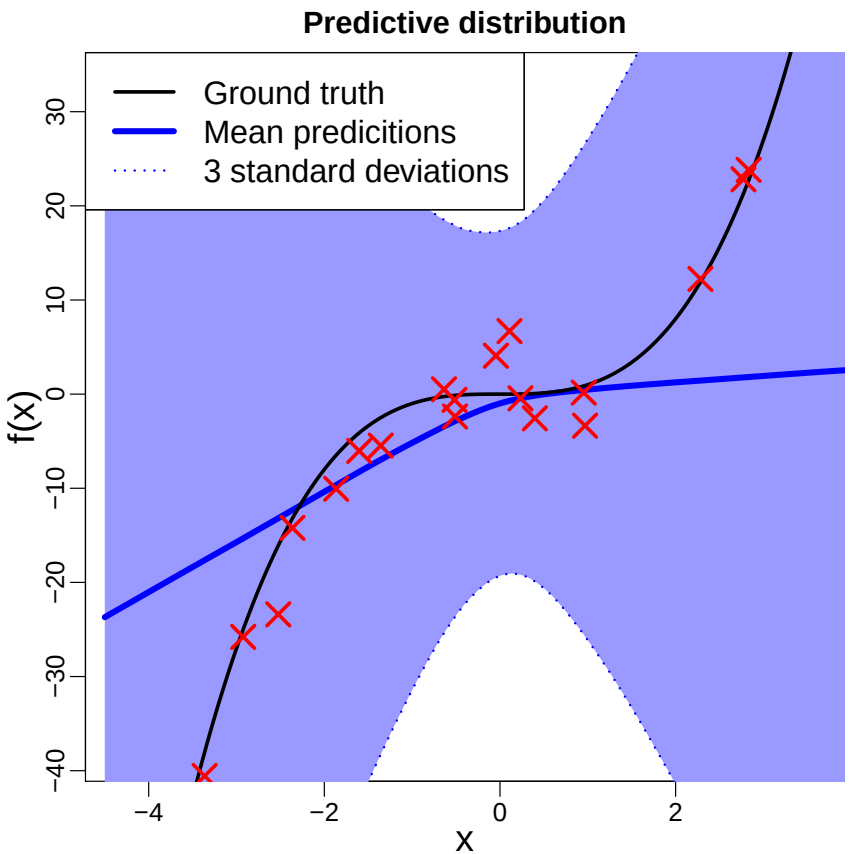


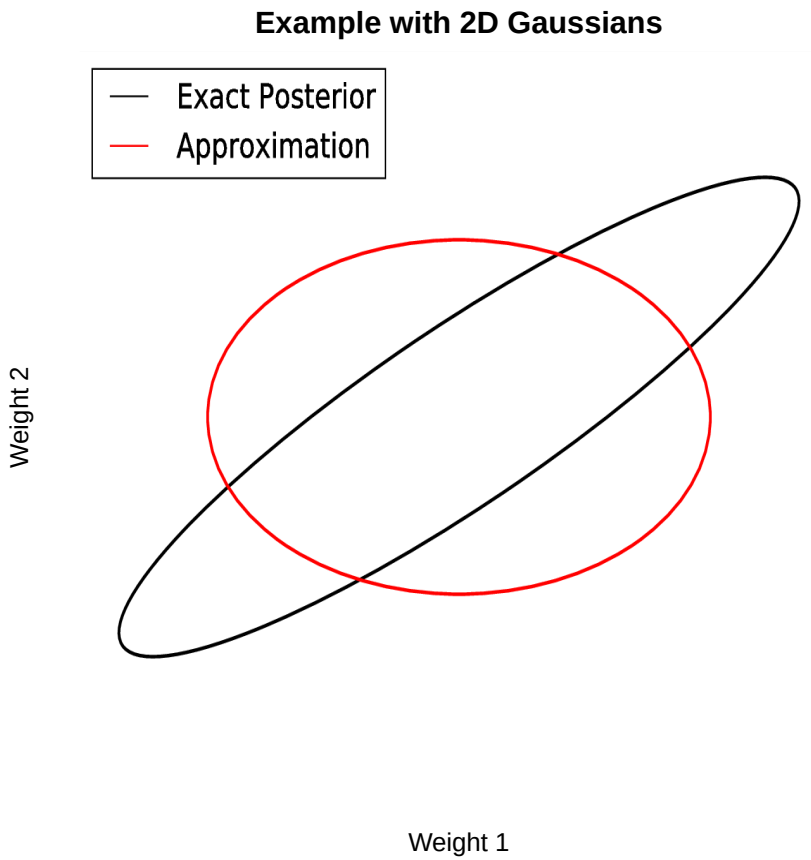




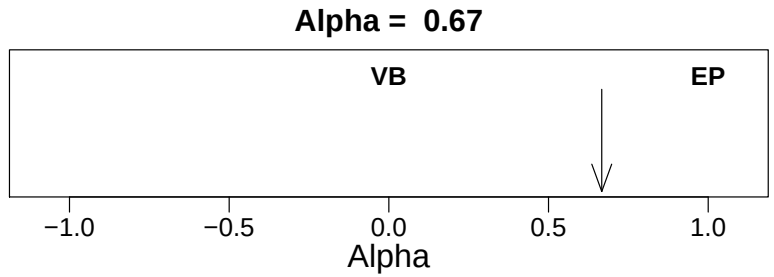
Regression with neural networks and 2D Gaussian

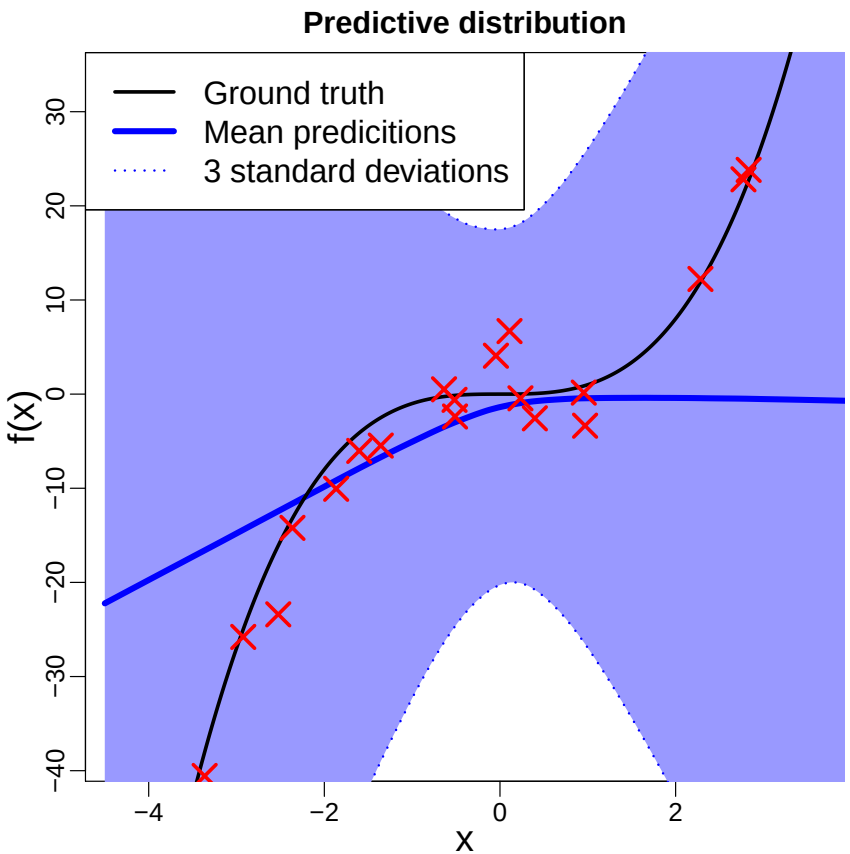


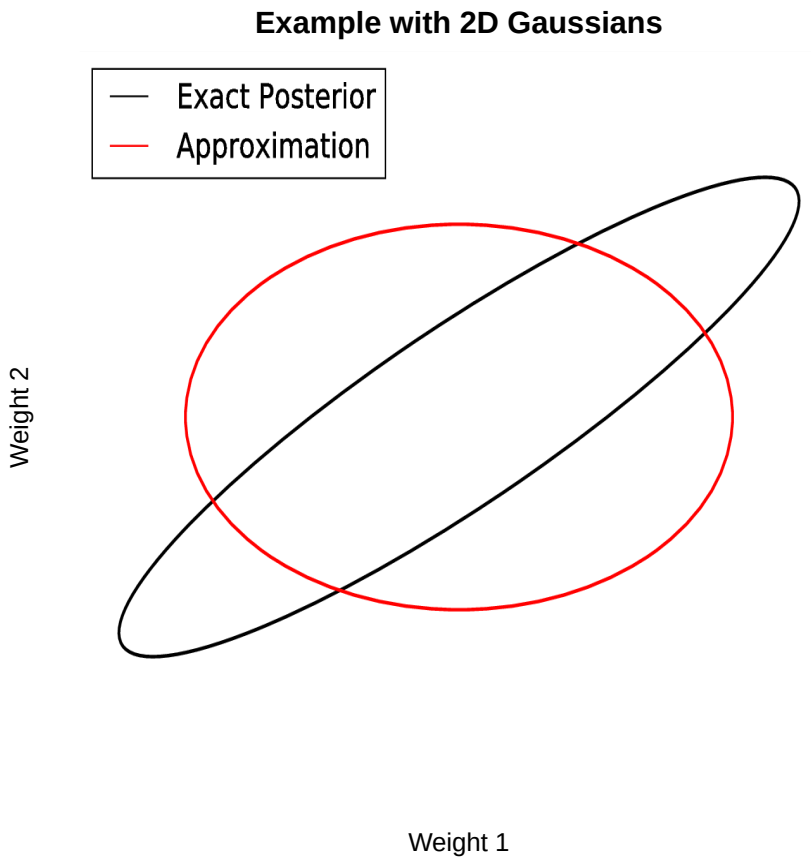




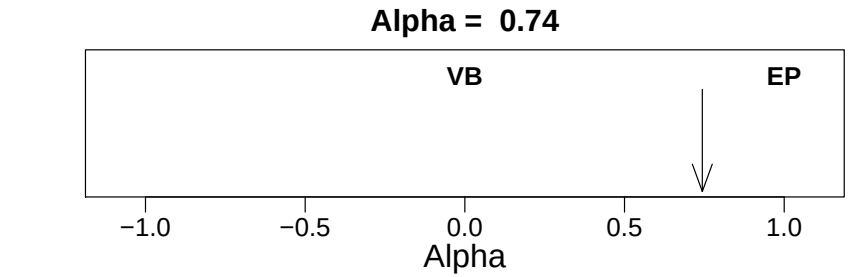
Regression with neural networks and 2D Gaussian

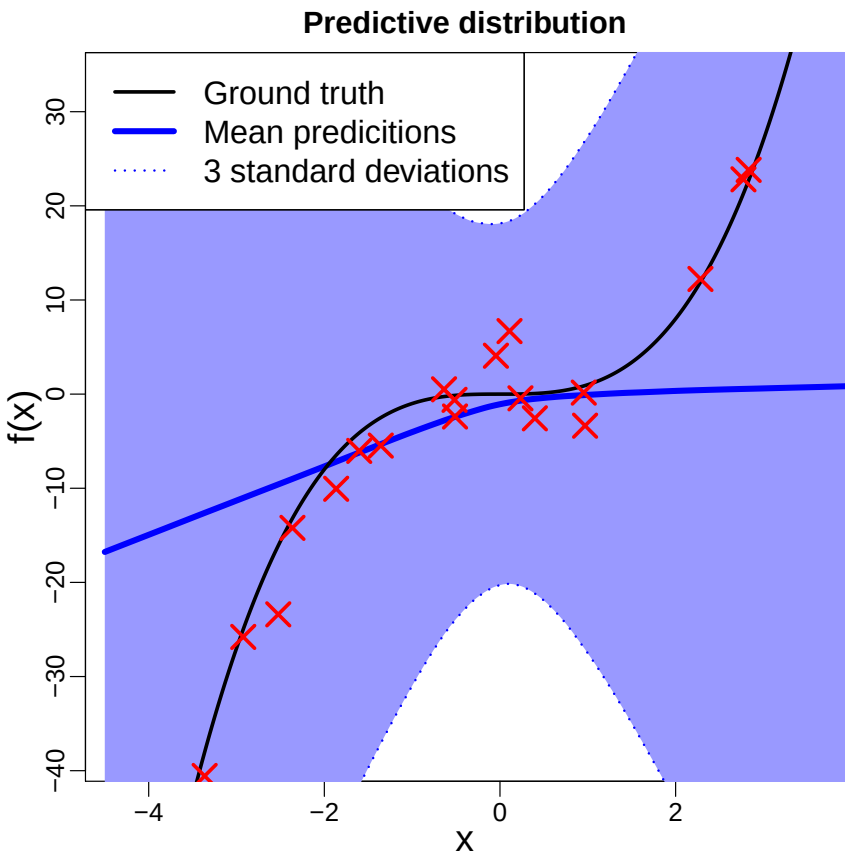


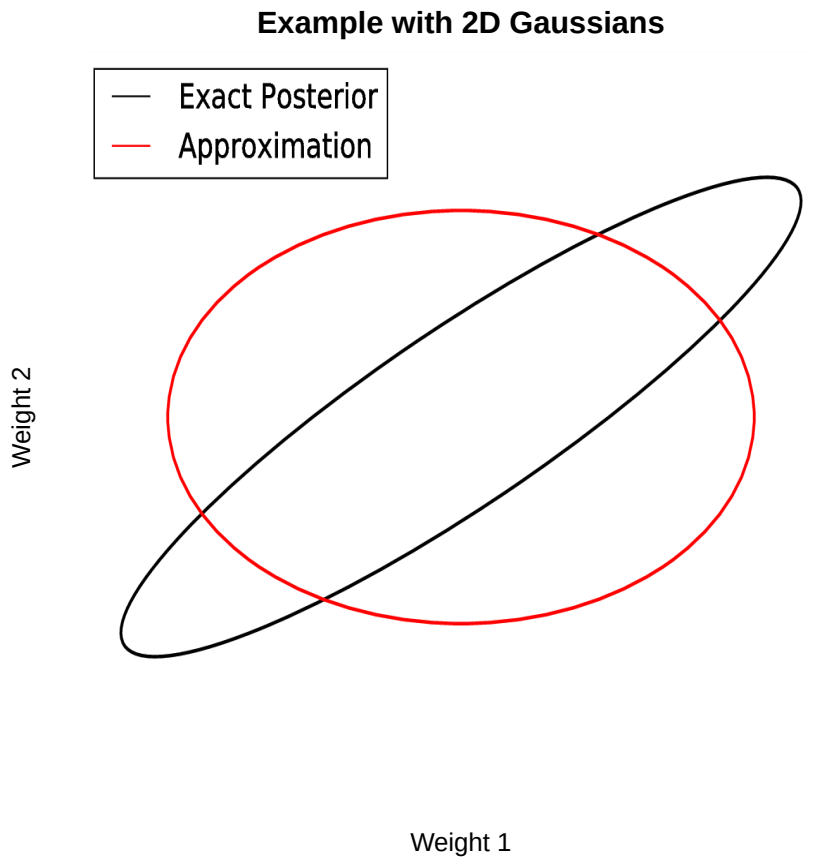




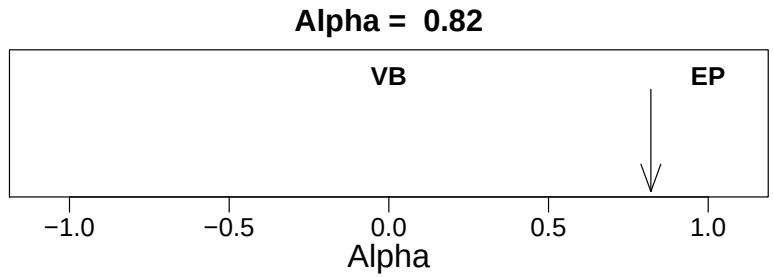
Regression with neural networks and 2D Gaussian

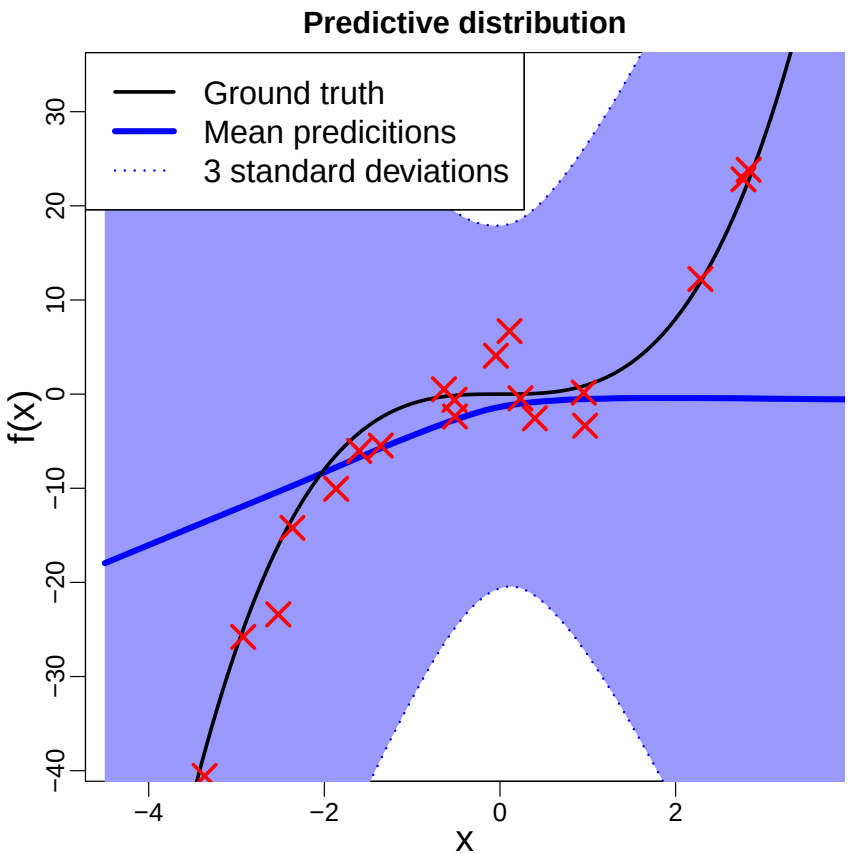


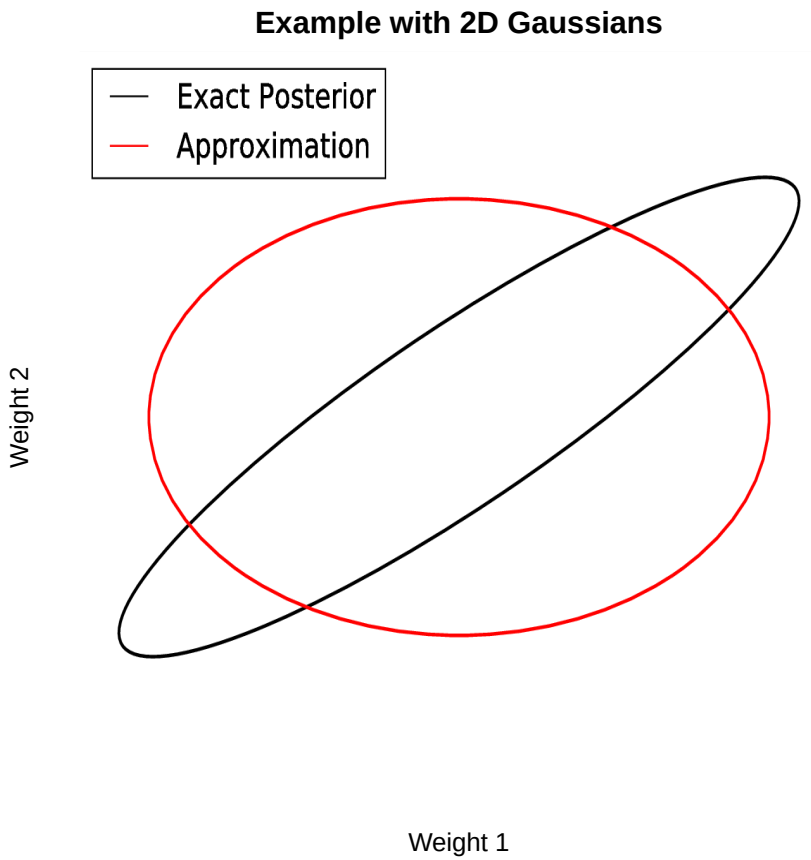




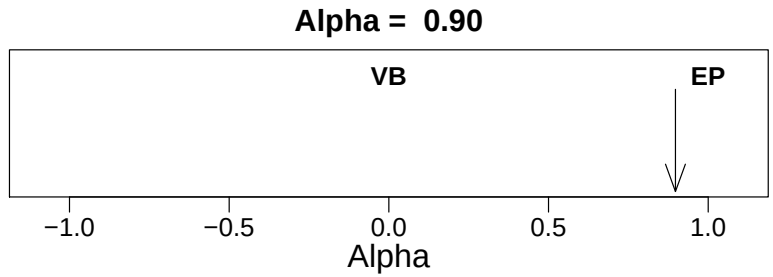
Regression with neural networks and 2D Gaussian

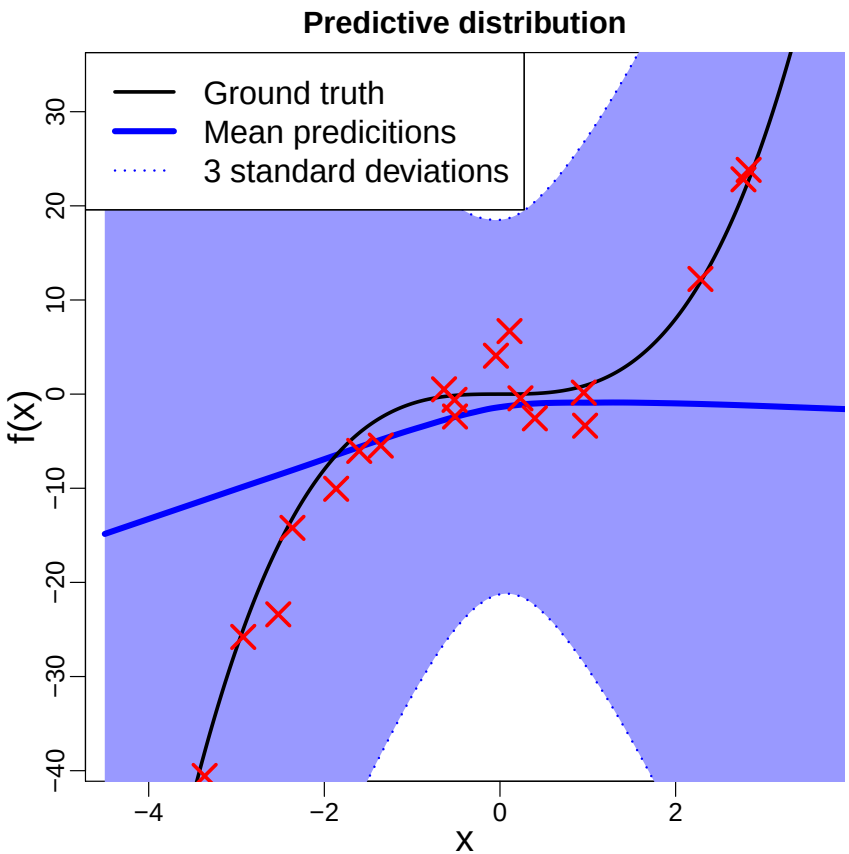


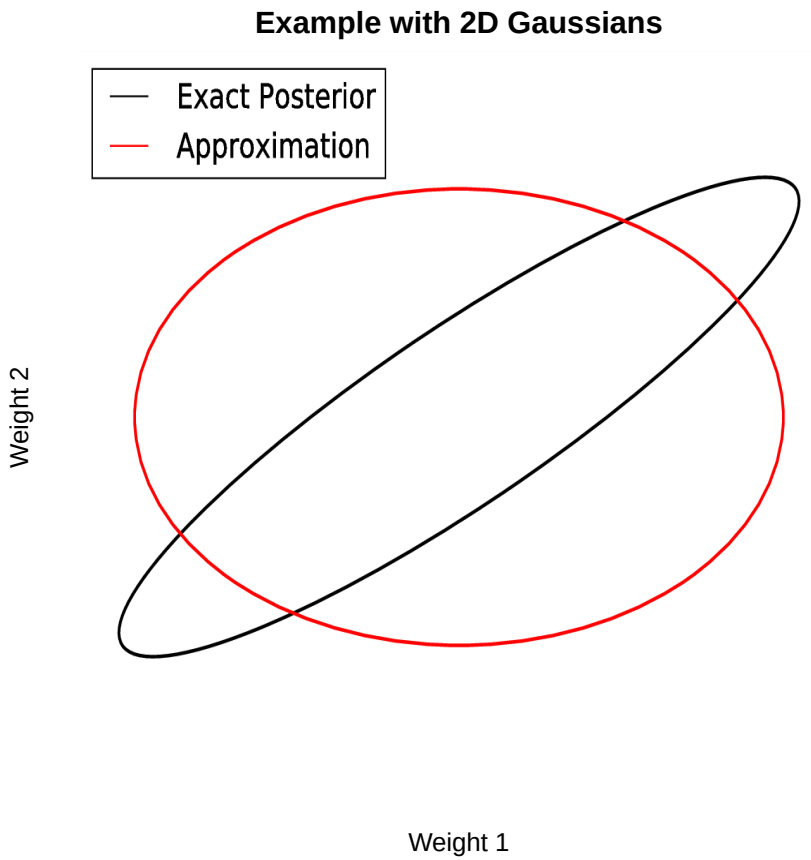




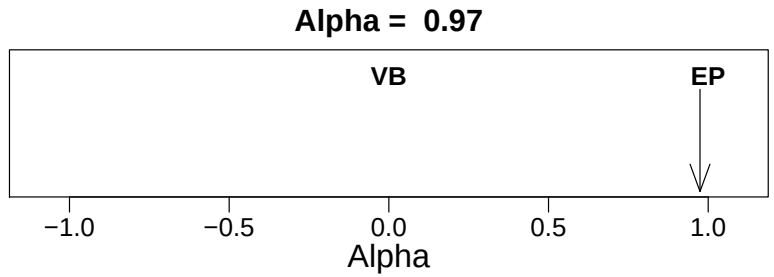
Regression with neural networks and 2D Gaussian

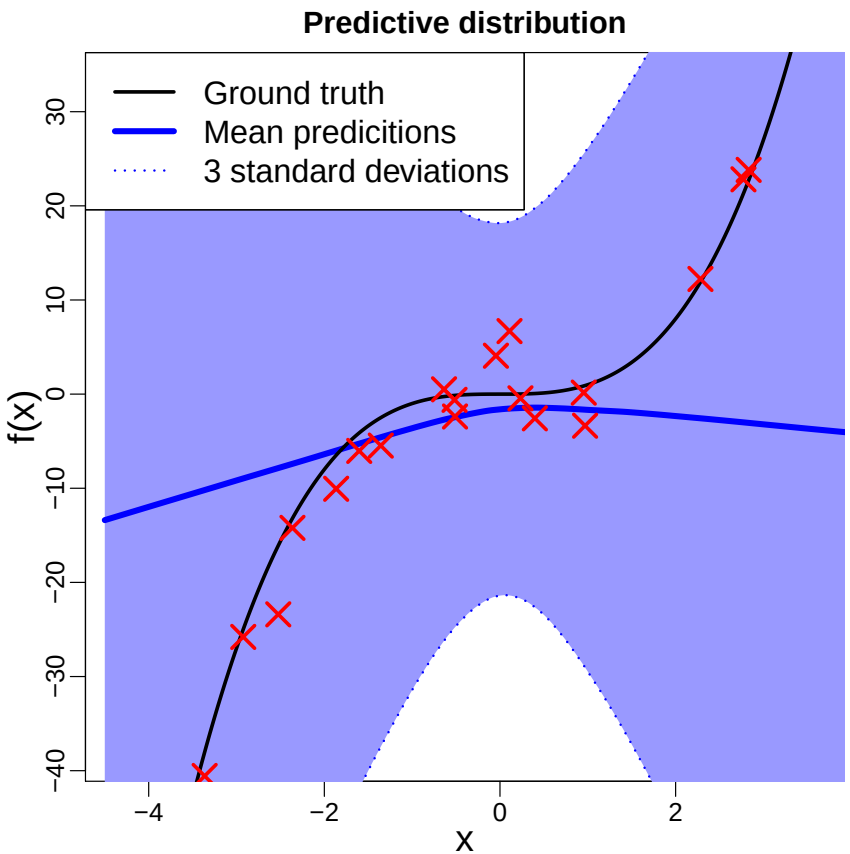


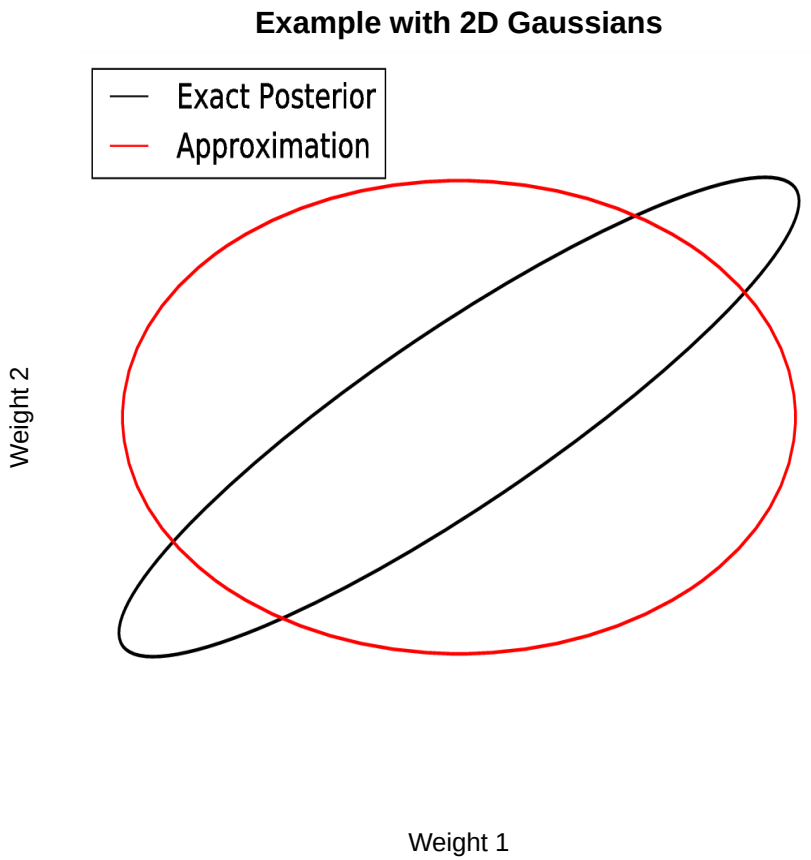




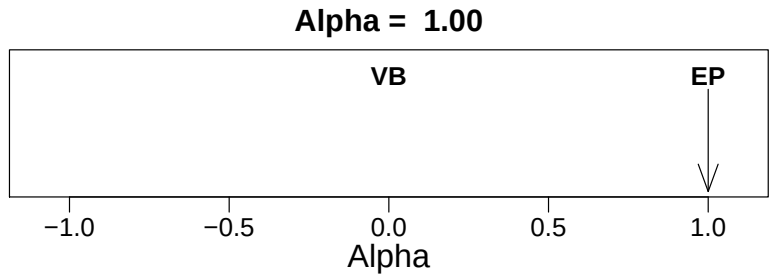
Regression with neural networks and 2D Gaussian

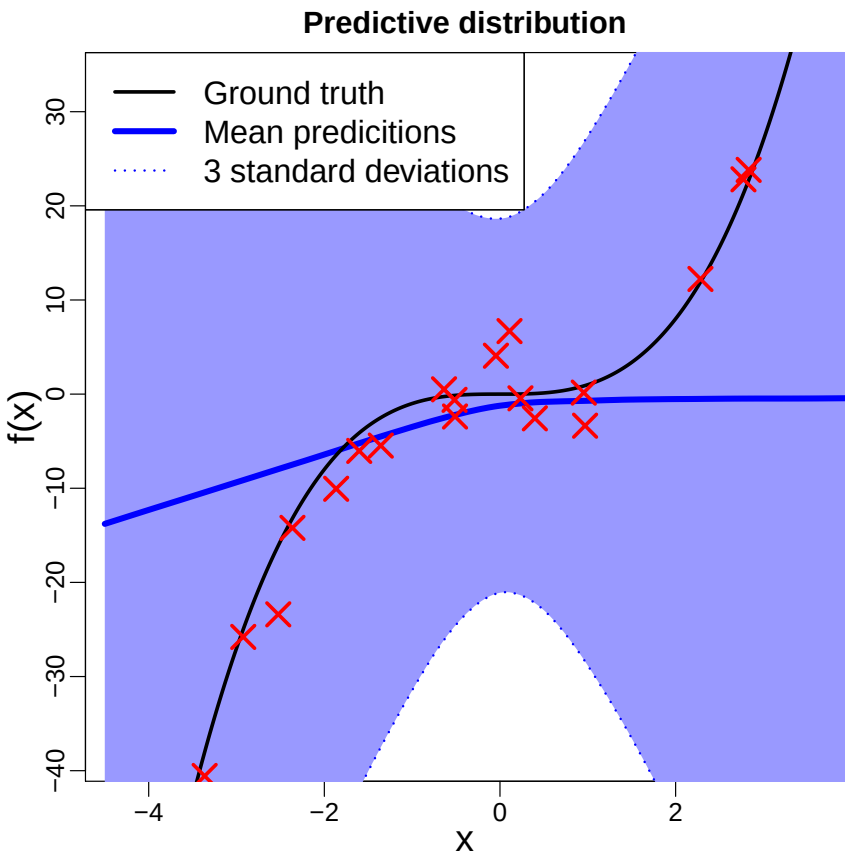


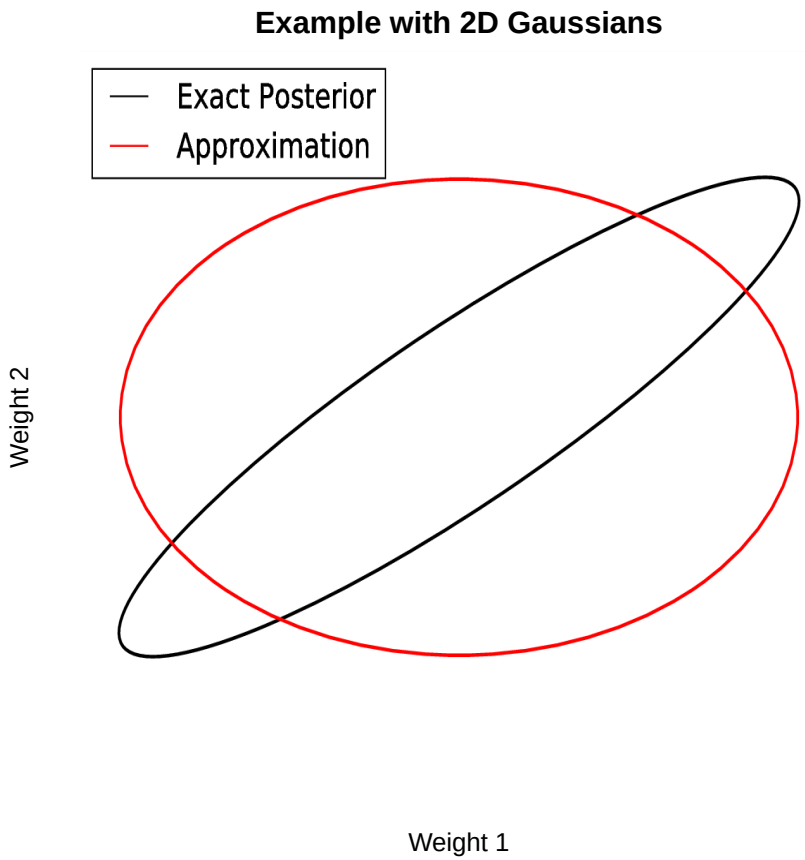




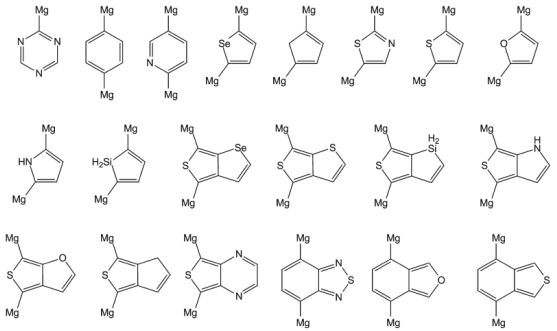
Regression with neural networks and 2D Gaussian

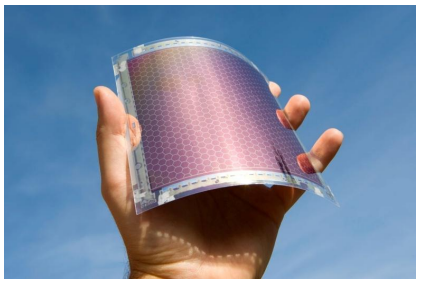






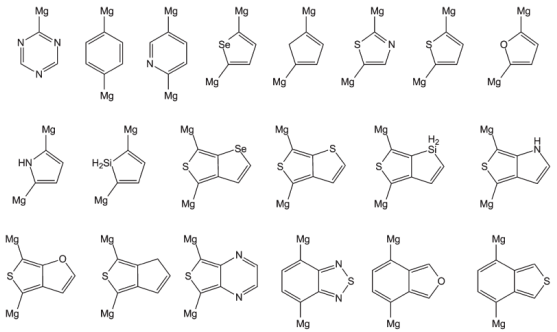
Harvard clean energy project. Data from 60.000 **organic photovoltaics**.

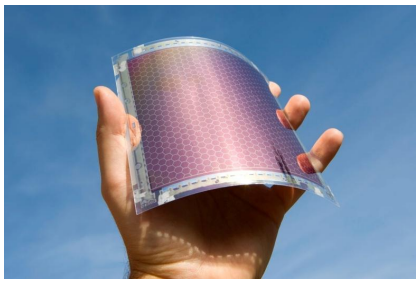




Bayesian neural networks, 2 hidden layers with 400 units each.

Harvard clean energy project. Data from 60.000 **organic photovoltaics**.



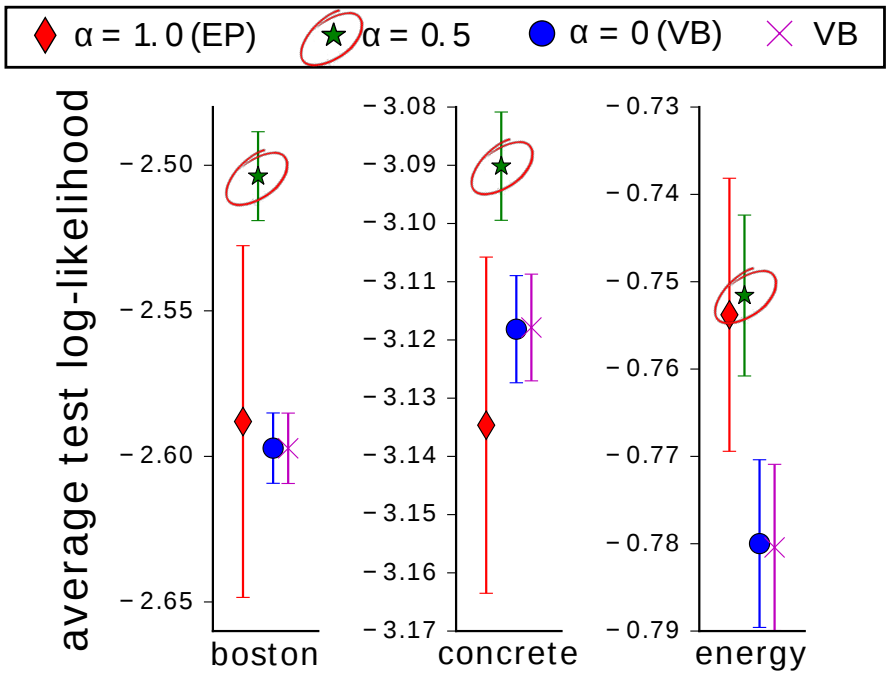


Bayesian neural networks, 2 hidden layers with 400 units each.

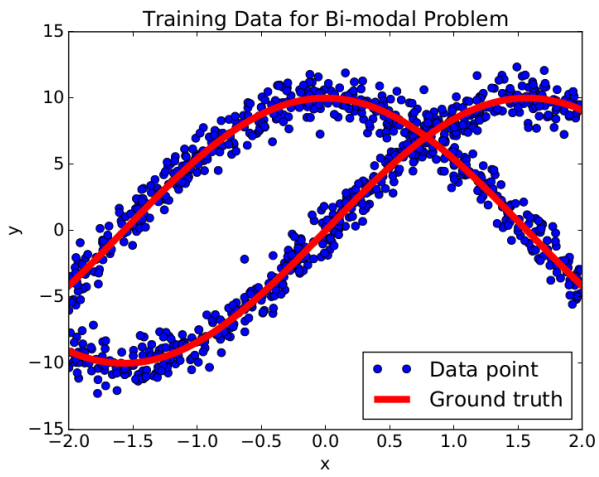
Table: Average Test Error and Test Log-likelihood.

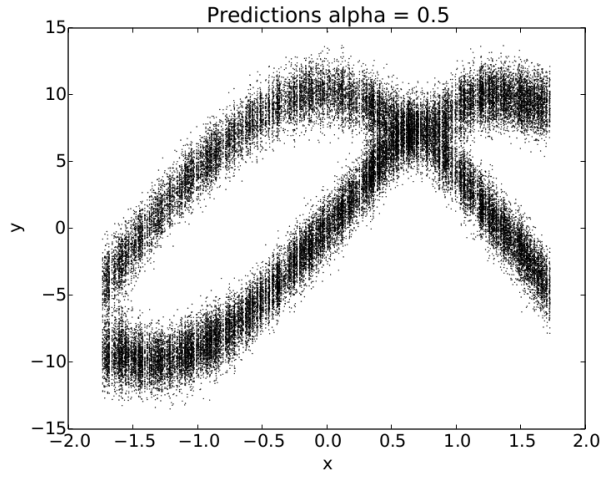
	α=1.0 **(EP)**	α=0.5	α=0	**VB**
Avg. Error	1.28±0.01	**1.08±0.01**	1.13±0.01	1.14±0.01
Avg. Log-likelihood	-0.93±0.01	**-0.74±0.01**	-1.39±0.03	-1.38±0.02

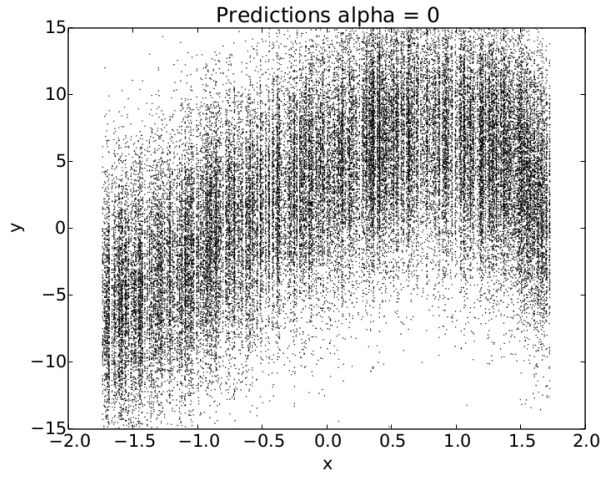
The value $\alpha = 0.5$ seems to be consistently better than $\alpha = 1$ (EP) or $\alpha = 0$ (VB).

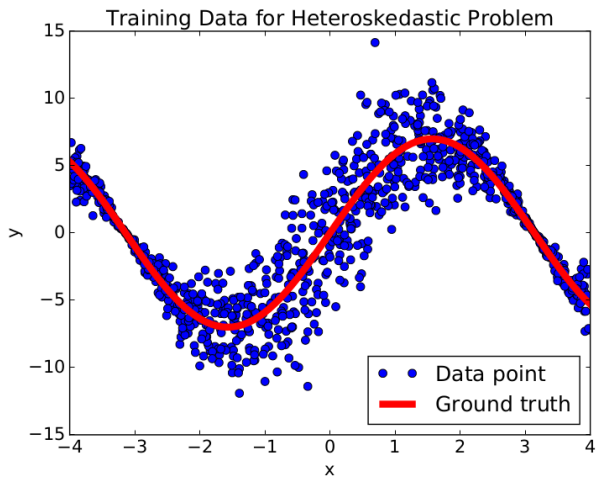


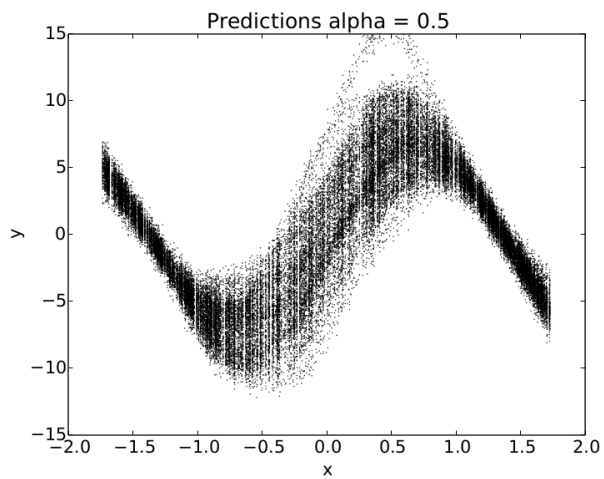
$\alpha = 0.5$ Variational Bayes

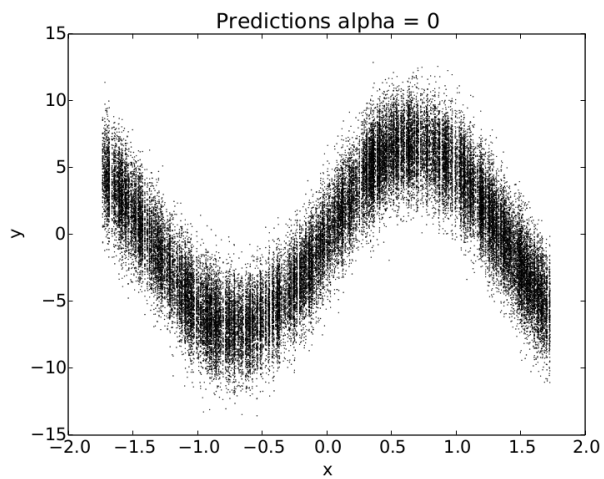












$\frac{1}{\alpha} \log \mathbf{E}_{q^{\text{cav}}}[f_n(\boldsymbol{\theta})^\alpha]$ $\mathbf{E}_q[\log f_n(\boldsymbol{\theta})]$

Depeweg et al. [2016]

Take home message

Black-box α-divergence minimization...

1. generalizes VB and an EP-like algorithm.
2. better than any of them in prediction problems with neural nets.
3. straightforward to apply in very complex models.

- Important applications in **probabilistic programming** to go beyond MCMC and VB.

References I

S. Amari. *Differential-geometrical methods in statistics*, volume 28. Springer Science & Business Media, 1985.

S. Depeweg, J. M. Hernández-Lobato, F. Doshi-Velez, and S. Udluft. Learning and policy search in stochastic dynamical systems with bayesian neural networks. *arXiv preprint arXiv:1605.07127*, 2016.

Heskes et al. Expectation propagation for approximate inference in dynamic bayesian networks. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, 2002.

A. Kucukelbir, R. Ranganath, A. Gelman, and D. Blei. Automatic variational inference in stan. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 568–576. Curran Associates, Inc., 2015.

T. Minka. Divergence measures and message passing. Technical report, Microsoft Research, 2005.

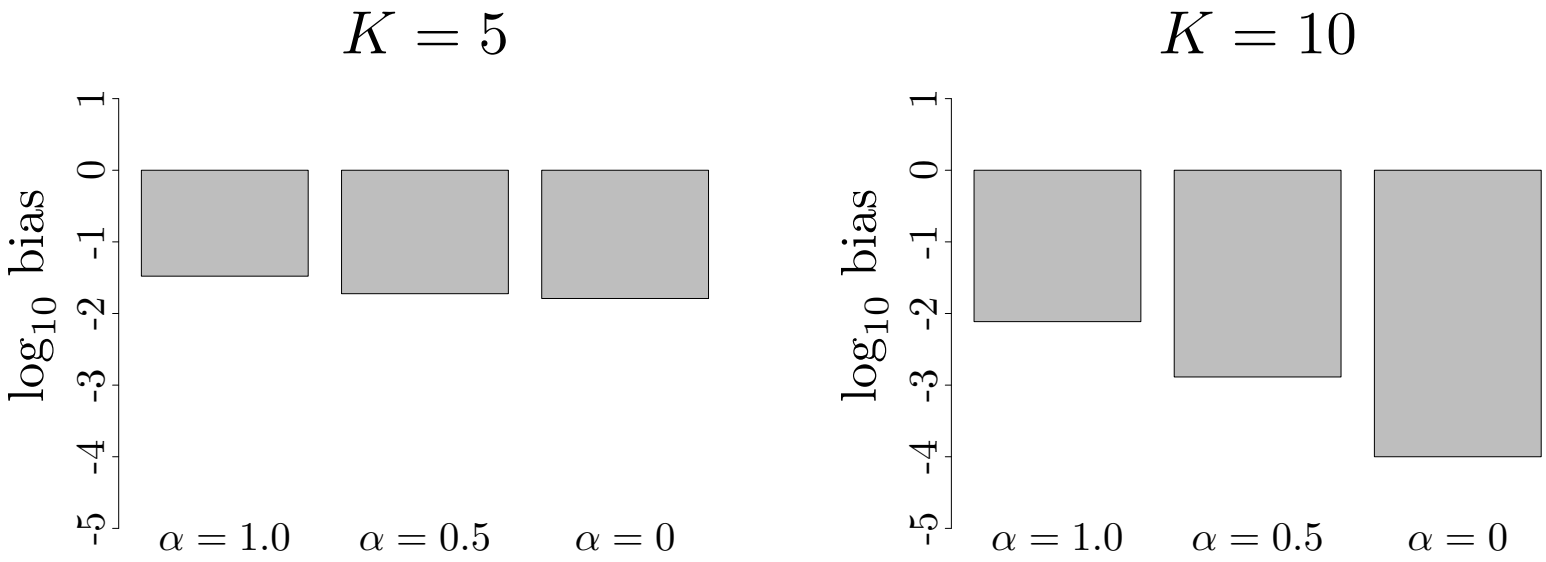
References II

T. P. Minka. Power ep. Technical report, Technical report, Microsoft Research, Cambridge, 2004.

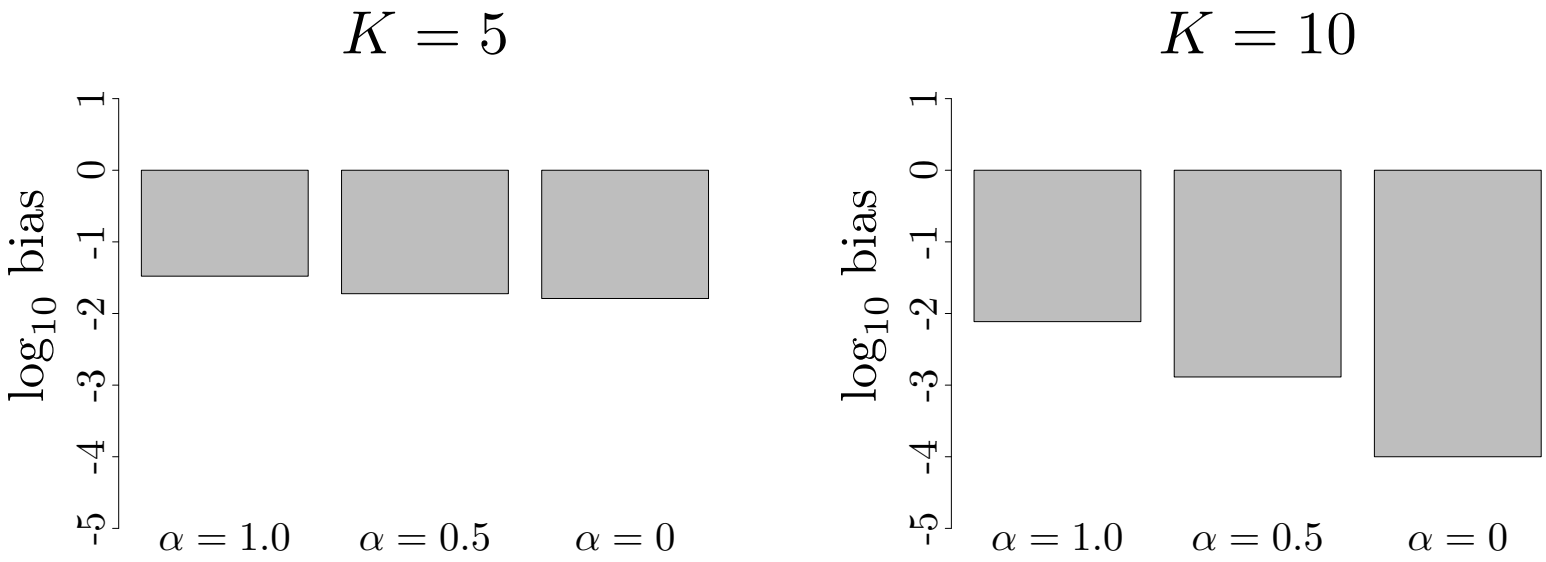
R. Ranganath, S. Gerrish, and D. Blei. Black box variational inference. In *AISTATS*, pages 814–822, 2014.

T. Salimans, D. A. Knowles, et al. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

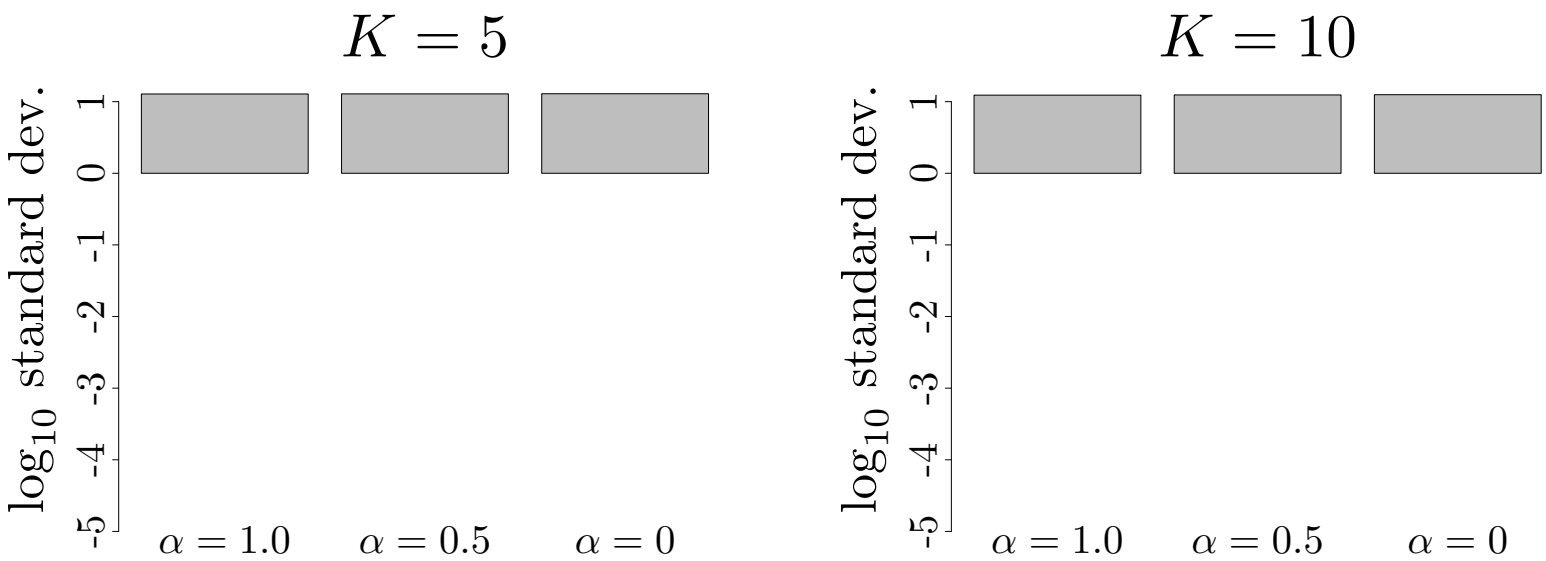
Average Bias in the Gradient.



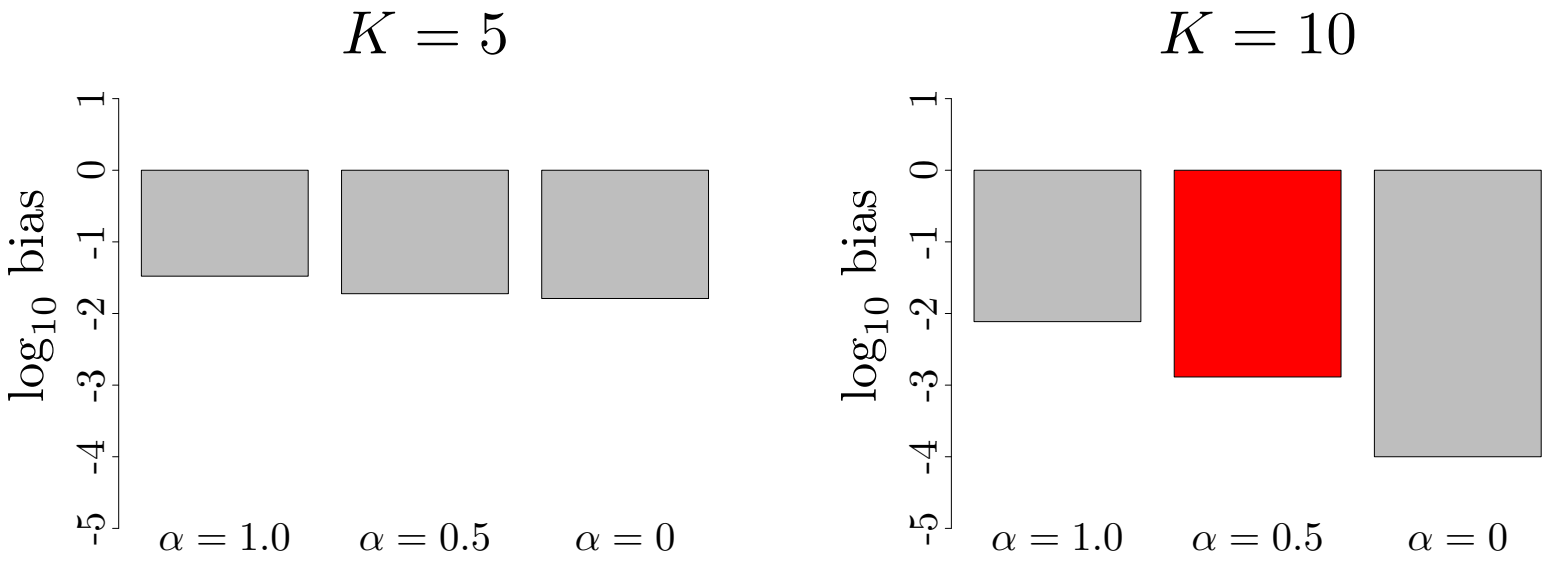
Average Bias in the Gradient.



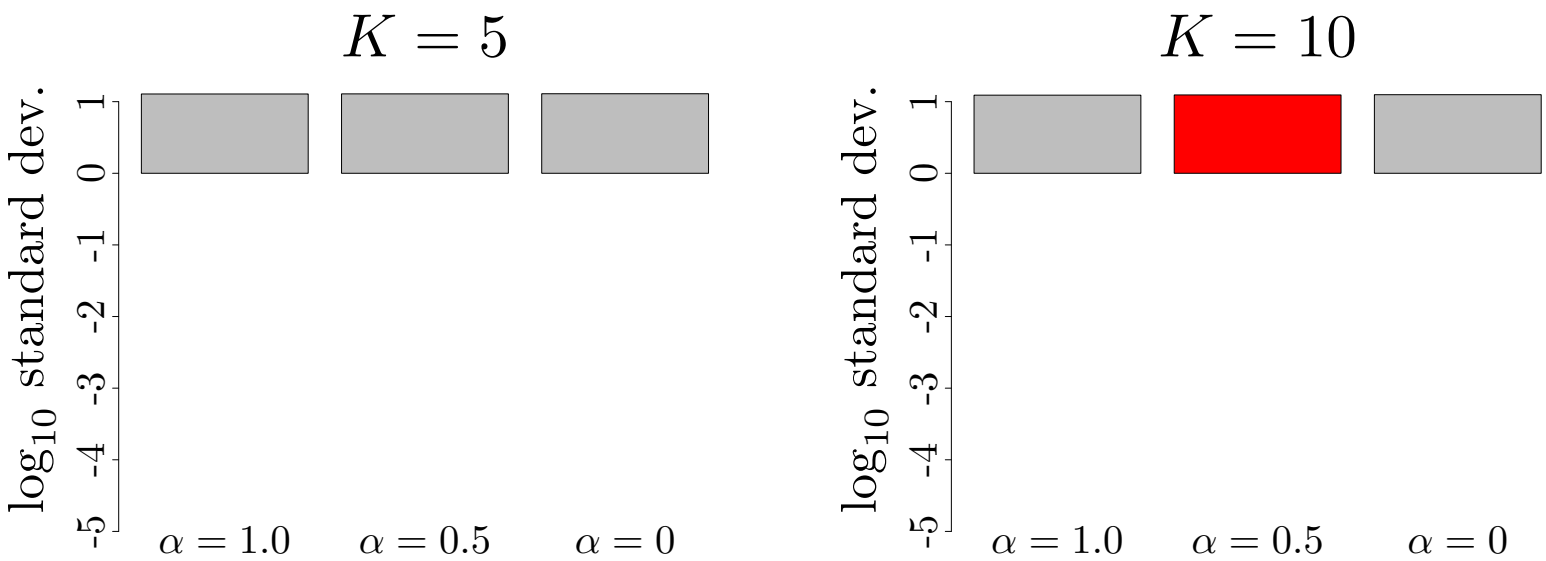
Average Standard Deviation in the Gradient.



Average Bias in the Gradient.



Average Standard Deviation in the Gradient.



Multi-class classification

Bayesian neural networks, 2 hidden layers with 400 units each.

Table: Average Test Error and Test Log-likelihood in MNIST.

MNIST	α=1.0	α=0.5	α=0	**VB**	α=$-$1.0
Error	0.0151±0.0001	0.0144±0.0001	0.0136±0.0001	0.0136±0.0001	**0.0133±0.0001**
Log-likelihood	-0.0551±0.0004	-0.0509±0.0003	-0.0468±0.0002	-0.0468±0.0002	**-0.0447±0.0002**