Ensemble Methods in Machine Learning

José Miguel Hernández-Lobato

Department of Engineering, University of Cambridge,
Trumpington Street, Cambridge, CB2 1PZ, UK,
jmh233@cam.ac.uk

December 19, 2012

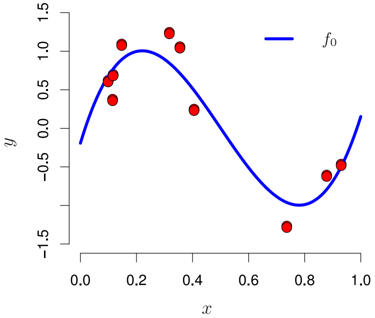
Motivation

Machine learning is about making **predictions** from data.

Prediction is often implemented using a **single** learning **method** which assumes a specific **model** that has to be **adjusted** to the data.

Two main difficulties in this approach:

- What **method** should we use to make predictions?
- What **values** of the model parameters are the optimal ones?



1. Decision tree?
2. Neural network?
3. Gaussian process?
4. ...

Ensemble methods can be used to address these difficulties!

Ensemble Methods

Instead of a single predictor, consider a **collection** of different predictors.

The ensemble prediction is a **combination** of the individual responses.

Two different types of ensembles:

Homogeneous

The same method is replicated several times with different parameter values.

Heterogeneous

Different learning methods are applied to the same training data.

Ensembles are often **better** than single predictors [Plikar, 2006]. For this,

- Predictors must be better than random guessing.
- Predictors must make complementary errors.

State of the Art Performance

Some applications in which ensembles obtain state of the art performance:

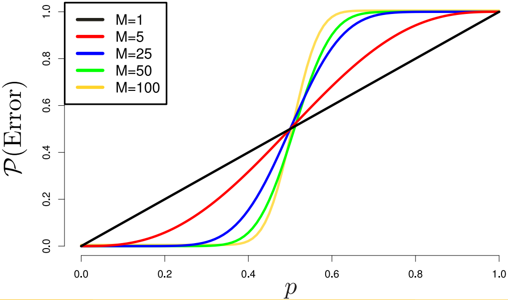
1. Recommendation (Netflix prize) [Koren and Bell, 2011].
2. Weather forecast [Gneiting and Raftery, 2005].
3. Real-time human pose recognition (Kinect) [Shotton et al. 2011].
4. Robust real-time face detection [Viola and Jones, 2004].
5. Gene function prediction [Ré and Valentini, 2010].
6. Reverse-engineering of biological networks [Marbach et al. 2009].
7. Credit card fraud detection [Bhattacharyya et al. 2011].

Why do Ensemble Methods Work?

Error probability in an ensemble of independent classifiers:

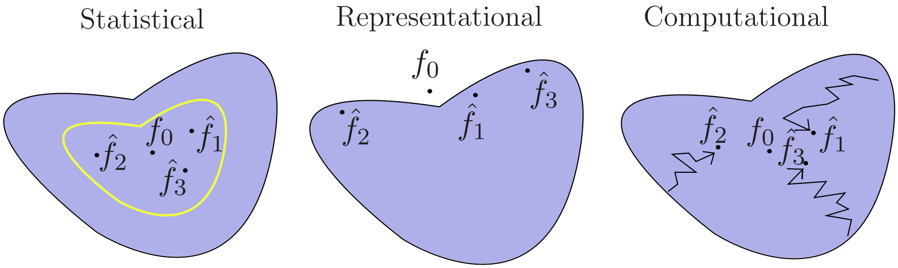
Binary classification problem. Ensemble of size M.
Predictors make **independent** errors with probability $p < 0.5$.
Individual predictions combined by **majority voting**.

$$\mathscr{P}(\text{Error}) = \sum_{m=\lceil \frac{M}{2} \rceil}^{M} \binom{M}{m} p^m (1-p)^{M-m} = I_p(\lfloor \frac{M}{2} \rfloor + 1, M - \lfloor \frac{M}{2} \rfloor)$$



Other Reasons for Using Ensemble Methods

The assumption of independent errors **does not hold** in practice. But there are other reasons for using ensembles:



[Dietterich, 2000]

Additionally,

- Empirical reduction in the bias and variance components of the error.
- Improved robustness to noise in the training data.
- Reduction in overfitting.

How to Generate Homogeneous Ensembles?

The ensemble components should be **accurate** and make **different** errors. For the latter, we can

- Use different perturbed versions of the training set by
  1. Manipulating training examples.
  2. Manipulating input features.
  3. Manipulating target variables.
- Induce some randomness in the training process. For example,
  1. Random initializations in neural networks.
  2. Random splits in decision trees.

Performance usually depends on a set of **parameters** that determine the amount of perturbation or randomization.

How to Combine the Individual Responses?

Parallel Combination:

Predictors are queried independently. Their responses are then combined.

- Majority voting and simple averaging.
- Weighted majority voting and weighted averaging.
- Input dependent weighted combination (mixture of experts).
- Stacking.

Cascade Generalization:

The input to each predictor in a sequence is the output of previous predictors and the original data instance.

Dynamic Integration:

A meta-level predictor selects the element with lowest expected error.

Hierarchichal Combination:

Predictors located at the leaves of a tree. Input dependent gating networks at the internal nodes compute response probabilities.

Bagging I

Bagging comes from *bootstrap* + *aggregation* [Breinman, 1996]. Improvements are obtained from a reduction in **variance**.

Consider a **regression** problem with M **independent** training sets $\mathscr{D}_1, \ldots, \mathscr{D}_M$, corresponding predictors $\hat{f}_1, \ldots, \hat{f}_M$ and **aggregated** predictor

$$\hat{f}_{\text{avg}}(\mathbf{x}) = \sum_{i=1}^{M} \frac{1}{M} \hat{f}_i(\mathbf{x}),$$

The expected prediction error of $\hat{f}_{\text{avg}}$ is

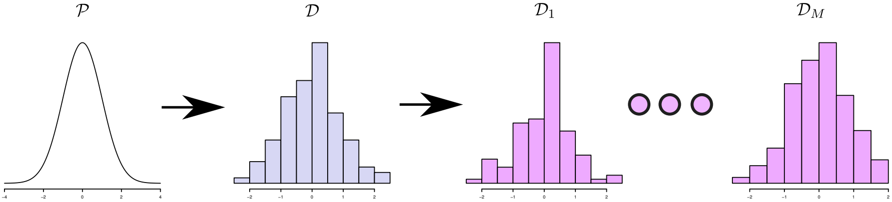
$$\mathbb{E}_{\{\mathscr{D}_i\}_{i=1}^{M}}[(\hat{f}_{\text{avg}}(\mathbf{x}) - y)^2] = \frac{1}{M}\overline{\text{Var}} + (1 - \frac{1}{M})\overline{\text{Cov}} + \overline{\text{Bias}}^2,$$

where $\overline{\text{Var}}$, $\overline{\text{Cov}}$ and $\overline{\text{Bias}}$ are respectively the average variance, covariance and bias of the individual predictors.

Bagging II

In practice we only have a single dataset $\mathscr{D}$ for training.

Solution: $\mathscr{D}_1, \ldots, \mathscr{D}_M$ are bootstrap samples from $\mathscr{D}$.



Problems:

- Dependencies among the different predictors.
- Increment in the bias and variance of the members of the ensemble.

Nevertheless, the reduction of variance in the final ensemble often compensates for these problems!

Bagging III

Illustrative Example with simulated data [Hernandez-Lobato, 2009].

Each $\mathbf{x}_i$ sampled from $\mathcal{N}(\mathbf{m}_i, \mathbf{I})$ where $\mathbf{m}_i = (r_i, \ldots, r_i)$ and $r_i \sim U[0,3]$.
$\mathbf{x}_i$ is 20-dimensional.
Each y_i satisfies $y_i = 25\sin(r_i) r_i^{-1} + \varepsilon_i$, with $\varepsilon_i \sim \mathcal{N}(0,1)$.
100 training sets with 25 instances and single test set with 1000 instances.

We construct Bagging ensembles with 100 un-prunned **CART** trees.

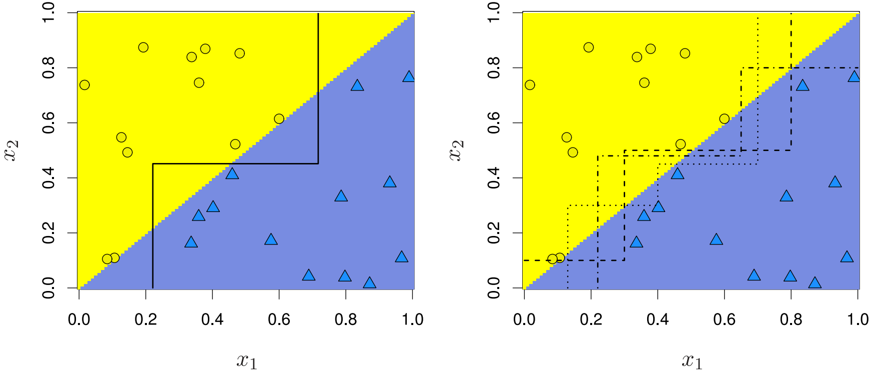
Method	**MSE**	**Bias^2**	**Variance**
Bagging	15.75	11.16	4.60
Single Tree	36.80	9.49	27.30

The reduction in the variance of the ensemble is larger than the increment in bias, in individual variance and in covariance!

Random Forests I

Bagging with un-prunned randomized CART trees [Breinman, 2001].

Besides the splits found by CART, other splits may also explain the data.



Random forests aim to take into account these alternative partitions.

Random Forests II

RF introduce some **randomization** in the construction of CART trees.
At each split, only a **subset** of m randomly chosen **features** are examined.
This increases the **variance** but also reduces the **covariance** of the trees.

Example on the same simulated data used before.
Results of a random forest ensemble of size 100 with $m = 1$:

Ensemble Method	**MSE**	$\overline{\textbf{Bias}}^2$	$\overline{\textbf{Var}}$	$\overline{\textbf{Cov}}$
Bagging	15.75	11.15	31.82	4.32
Random Forest	12.07	10.70	34.27	1.04

The **reduction** in covariance leads to lower predictive error!

OOB samples allow to estimate **test error** and do **feature selection** [Díaz-Uriarte et al. 2006].

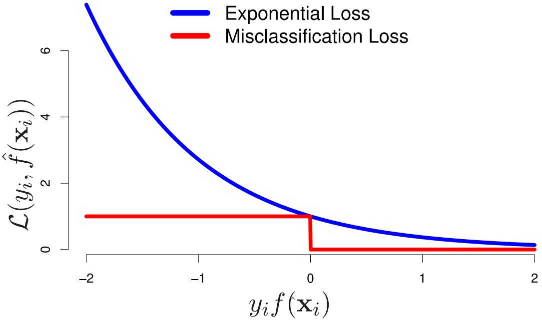
Adaboost I

Generates powerful and expressive predictors by combining **weak** learners [Freund and Schapire, 1996].

The weak method is applied to repeatedly **modified** versions of the data.

The final adaboost decision is obtained by **weighted majority voting**.

Adaboost minimizes the **exponential loss**: $\ell[y_i, \hat{f}(\mathbf{x}_i)] = \exp[-y_i \hat{f}(\mathbf{x}_i)]$.



$$\hat{f}(\mathbf{x}_i) = \sum_m \beta_m \hat{f}_i(\mathbf{x}_i)\,,$$

$$y_i \in \{-1, 1\}\,,$$

$$\hat{f}_m(\mathbf{x}_i) \in \{-1, 1\}\,.$$

Adaboost II

At iteration k adaboost adds the predictor $\hat{f}_k$ with weight β_k such that

$$
\begin{aligned}
(\hat{f}_k, \beta_k) &= \underset{(\tilde{f},\tilde{\beta})}{\arg\min} \sum_{i=1}^n \exp[-y_i \sum_{m=1}^{k-1} \beta_m \hat{f}_m(\mathbf{x}_i) - y_i \tilde{\beta} \tilde{f}(\mathbf{x}_i)] \\
&= \underset{(\tilde{f},\tilde{\beta})}{\arg\min} \sum_{i=1}^n w_i^k \exp[-y_i \tilde{\beta} \tilde{f}(\mathbf{x}_i)]\,,
\end{aligned}
$$

where $w_i^k = \exp[-y_i \sum_{m=1}^{k-1} \beta_m \hat{f}_m(\mathbf{x}_i)]$ is the weight of the i-th instance.

The solution is

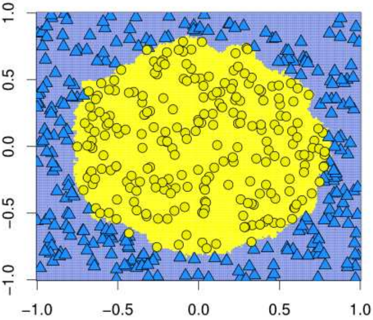
$$
\hat{f}_k = \underset{\tilde{f}}{\arg\min} \sum_{i=1}^n w_i^k \mathbb{I}[y_i \neq \tilde{f}(\mathbf{x}_i)]\,, \qquad \beta_k = \frac{1}{2} \log \frac{1 - \epsilon_k}{\epsilon_k}
$$

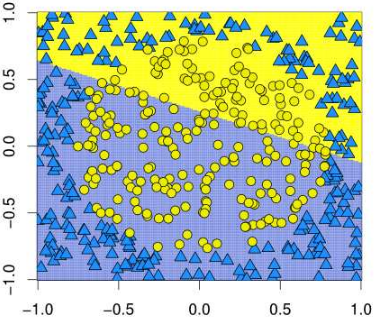
$$
\epsilon_k = \frac{1}{\sum_{i=1}^n w_i^k} \sum_{i=1}^n w_i^k \mathbb{I}[\hat{f}_k(\mathbf{x}_i) \neq y_i]\,. \tag{1}
$$

Adaboost III

Illustrative Example with simulated data [Hernandez-Lobato, 2009].

Each $\mathbf{x}_i$ sampled uniformly from $[-1,1]^2$, $y_i = \mathrm{sign}(x_1^2 + x_2^2 - 2\pi^{-1})$.
Training sets with 500 instances.
Logistic regression model. Slightly better than random guessing!
Adaboost ensemble of size 1000.
On each iteration a new training set is sampled using weights $w_1, \ldots, w_{500}$.





Adaboost IV

The first iterations reduce **bias**, while the last ones reduce **variance**.

Adaboost can be affected by significant **overfitting** problems if some of the data instances are **mislabeled**.

Extensions:

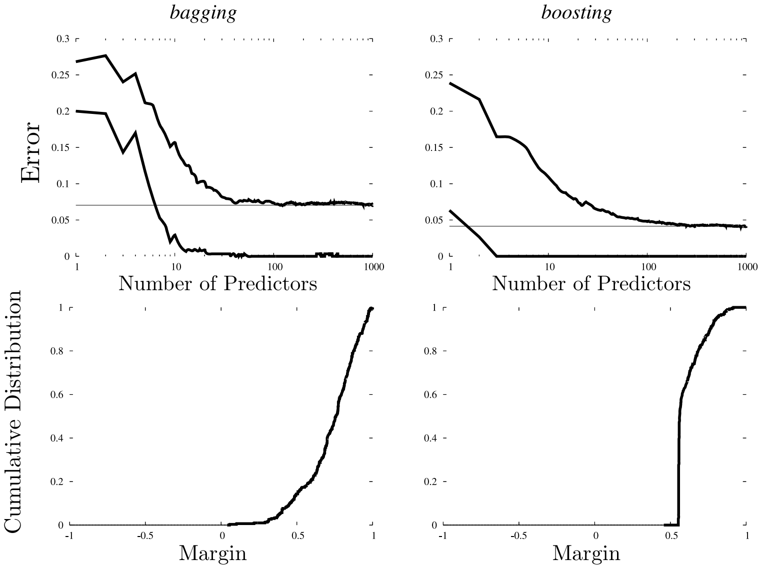
A boosting method for **regression** can be obtained by fitting each weak learner to the residual errors of the current ensemble [Friedman, 2001]. Used for blending in the netflix prize solution [Koren, 2009].

Boosting for **multi-class** problems [Saberian and N. Vasconcelos, 2011].

Margin Maximization:

Adaboost maximizes the margin of the most difficult training examples. **Illustrative example** comparing boosting and bagging with CART trees on the two-norm dataset [Martínez-Muñoz, 2006].

Adaboost V



bagging
boosting
Error
0.3
0.25
0.2
0.15
0.1
0.05
0
1
10
100
1000
Number of Predictors
Number of Predictors
Cumulative Distribution
1
0.8
0.6
0.4
0.2
0
-1
-0.5
0
0.5
1
Margin
Margin

How Large Should Binary Classification Ensembles Be? I

We focus on **parallel** ensembles such as **bagging** and **random forests**.

The error of the ensemble **decreases** with its size M [Breinman, 2001].

How to choose the optimal value of M?

- If M is too **large** we waste computational resources.
- If M is too **small** we loose prediction accuracy.

Practical solution proposed in [Hernández-Lobato, 2010]:

Stop including classifiers to the ensemble when it is **unlikely** that adding extra classifiers will **change** the ensemble prediction.

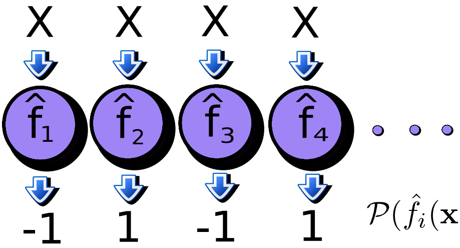
For fixed $\mathbf{x}$, the predictions of classifiers $\hat{f}_i$ and $\hat{f}_j$ are **independent**:

$$\mathscr{P}[\hat{f}_i(\mathbf{x}) = y', \hat{f}_j(\mathbf{x}) = y''] = \mathscr{P}[\hat{f}_i(\mathbf{x}) = y']\mathscr{P}[\hat{f}_j(\mathbf{x}) = y''] .$$

where y' and y'' are any class labels.

How Large Should Binary Classification Ensembles Be? II

For fixed **x**, the ensemble prediction is the result of a **binomial experiment**:



$$\mathcal{P}(\hat{f}_i(\mathbf{x}) = 1) = \pi_{\mathbf{x}}$$

The probability of the ensemble predicting class 1 is:

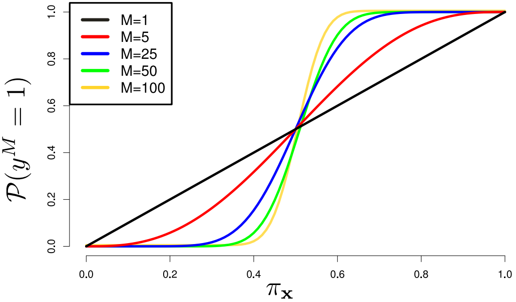
$$\mathscr{P}(y^M = 1) = \sum_{m=\lceil \frac{M}{2} \rceil} \binom{M}{m} \pi_{\mathbf{x}}^m (1 - \pi_{\mathbf{x}})^{M-m} = I_{\pi_{\mathbf{x}}}(\lfloor \frac{M}{2} \rfloor + 1, M - \lfloor \frac{M}{2} \rfloor)$$

How Large Should Binary Classification Ensembles Be? III

Asymptotically, $\mathscr{P}(y^M = 1)$ converges to a step function:

$$\lim_{M\to\infty} \mathscr{P}(y^M = 1) = \begin{cases} 1 & \text{if } \pi_{\mathbf{x}} > 1/2, \\ 1/2 & \text{if } \pi_{\mathbf{x}} = 1/2, \\ 0 & \text{if } \pi_{\mathbf{x}} < 1/2. \end{cases}$$

This can be observed in this plot of $I_{\pi_{\mathbf{x}}}(\lfloor \frac{M}{2} \rfloor + 1, M - \lfloor \frac{M}{2} \rfloor)$:



How Large Should Binary Classification Ensembles Be? IV

The M-size ensemble agrees with the infinite ensemble with probability

$$\mathscr{P}(y^M = y^\infty) = I_{\max\{\pi_{\mathbf{x}}, 1-\pi_{\mathbf{x}}\}}(\lfloor \tfrac{M}{2} \rfloor + 1, M - \lfloor \tfrac{M}{2} \rfloor).$$

A Gaussian approximation is given by

$$\mathscr{P}(y^M = y^\infty) \approx \Phi\left[\frac{M\max\{\pi_{\mathbf{x}}, 1-\pi_{\mathbf{x}}\} - M/2}{\sqrt{M\pi_{\mathbf{x}}(1-\pi_{\mathbf{x}})}}\right], \text{ where } \Phi(x) = \int_{-\infty}^{x} \mathcal{N}(u|0,1)\, du.$$

We can solve for M as a function of $\mathscr{P}(y^M = y^\infty)$:

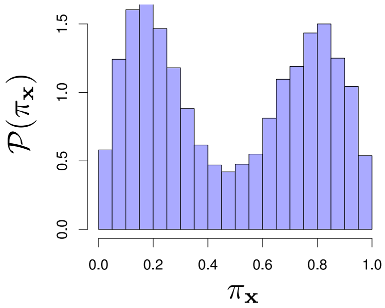
$$M_{\mathscr{P}(y^M = y^\infty)} \approx \frac{\Phi^{-1}[\mathscr{P}(y^M = y^\infty)]^2 (1-\pi_{\mathbf{x}})}{(\pi_{\mathbf{x}} - 1/2)^2},$$

For any $\mathscr{P}(y^M = y^\infty)$, if $\pi_{\mathbf{x}} \to 1/2$ then $M_{\mathscr{P}(y^M = y^\infty)} \to \infty$.

How Large Should Binary Classification Ensembles Be? V

The ensemble size depends on the number of instances $\mathbf{x}$ with $\pi_{\mathbf{x}} \approx 1/2$.

We can consider $\pi_{\mathbf{x}}$ a random variable with density $\mathscr{P}(\pi_{\mathbf{x}})$.



Histogram of 10,000 samples from $\mathscr{P}(\pi_{\mathbf{x}})$ for the *Twonorm* problem.

Estimates obtained using a random forest (RF) with 10,000 trees.

The training set has 300 labeled instances.

Note that, since $\pi_{\mathbf{x}}$ is random, $M_{\mathscr{P}(y^M = y^\infty)}$ is also a random variable.

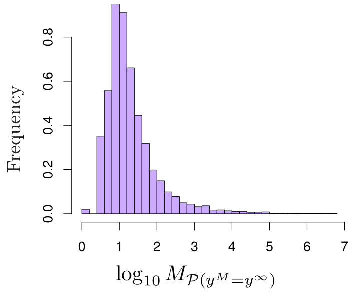
How Large Should Binary Classification Ensembles Be? VI

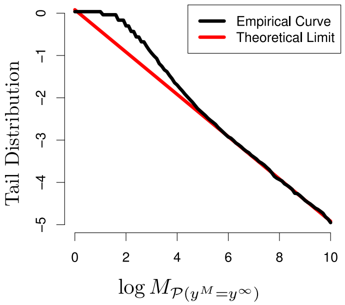
We can approximate $\mathscr{P}(M_{\mathscr{P}(y^M=y^\infty)} > z)$ when z is large by

$$\mathscr{P}(M_{\mathscr{P}(y^M=y^\infty)} > z) \approx \frac{\mathscr{P}(\pi_{\mathbf{x}} = 1/2)\Phi^{-1}[\mathscr{P}(y^M = y^\infty)]}{\sqrt{z}},$$

Universal heavy-tailed behavior!
Only depends on the classification problem through $\mathscr{P}(\pi_{\mathbf{x}} = 1/2)$.

Illustration on *twonorm* for $\mathscr{P}(y^M = y^\infty) = 0.99$:





How Large Should Binary Classification Ensembles Be? VII

We can approximate the probability that the **infinite** ensemble **agrees** with a **large finite** ensemble of size M:

$$\mathscr{P}(y^M = y^\infty) \approx 1 - \frac{\mathscr{P}(\pi_{\mathbf{x}} = 1/2)\int_{-\infty}^{0}\Phi(z)\,dz}{\sqrt{M}}\,, \quad M \to \infty\,,$$

Solving for M we obtain the **size of the ensemble** that agrees with the infinite ensemble with probability $\mathscr{P}(y^M = y^\infty)$ close to 1:

$$M^\star_{\mathscr{P}(y^M = y^\infty)} = \left[\frac{\mathscr{P}(\pi_{\mathbf{x}} = 1/2)\int_{-\infty}^{0}\Phi(z)\,dz}{1 - \mathscr{P}(y^M = y^\infty)}\right]^2\,,$$

Only depends on the classification problem through $\mathscr{P}(\pi_{\mathbf{x}} = 1/2)$.

How Large Should Binary Classification Ensembles Be? VIII

Practical implementation:

$M^\star_{\mathscr{P}(y^M=y^\infty)}$ is obtained as the minimum M such that

$$\mathscr{P}(y^M = y^\infty) \leq \frac{1}{N}\sum_{i=1}^N I_{\max\{\hat{\pi}_{\mathbf{x}}^{(i)},1-\hat{\pi}_{\mathbf{x}}^{(i)}\}}(\lfloor \frac{M}{2} \rfloor + 1, M - \lfloor \frac{M}{2} \rfloor),$$

where $\{\hat{\pi}_{\mathbf{x}}^{(i)}\}_{i=1}^N$ are estimated using OOB, validation or unlabeled data using an initial ensemble of size $M' = 100$.

If $M^\star_{\mathscr{P}(y^M=y^\infty)} > M'$ we set $M' = \min(M^\star_{\mathscr{P}(y^M=y^\infty)}, 2M')$ and repeat.

Empirical evaluation:

Ensembles of random forests and bagging with un-prunned CART trees.
18 UCI problems. Infinite ensemble approx. by finite one with 10,000 trees.
$\mathscr{P}(y^M = y^\infty)$ is selected to be 0.99.

How Large Should Binary Classification Ensembles Be? IX

Average disagreement rates:

Problem	RF-Test	RF-OOB	RF-BAN	Bag-Test	Bag-OOB	Bag-BAN
australian	1.0±0.6	1.2±0.7	2.3±1.1	1.0±0.6	1.1±0.7	2.3±1.3
breast	0.9±0.6	1.0±0.7	0.6±0.5	0.9±0.5	0.9±0.7	0.8±0.6
circle	1.0±0.4	1.1±0.5	1.3±0.6	1.0±0.4	1.0±0.5	1.3±0.7
echo	1.0±1.5	1.1±1.8	2.2±2.4	1.2±1.5	1.1±2.0	2.0±2.6
german	1.1±0.5	1.2±0.6	5.1±1.5	1.1±0.6	1.2±0.6	5.7±2.1
heart	1.2±1.1	1.3±1.2	4.7±3.1	1.3±1.0	1.2±1.1	4.9±3.4
hepatitis	1.5±1.4	1.5±1.8	4.7±3.4	1.3±1.5	1.2±1.8	5.2±3.6
horse	1.2±1.0	1.1±1.1	2.4±1.7	1.1±0.8	1.2±1.1	2.6±2.0
ionosphere	0.9±0.8	1.0±0.8	1.5±1.2	0.9±0.8	1.1±1.0	1.8±1.5
labor	1.8±2.8	1.9±2.9	3.5±4.9	1.4±2.6	1.7±3.7	3.2±4.2
liver	1.5±1.1	1.5±1.2	8.5±3.5	1.3±1.0	1.2±0.9	7.6±4.0
pima	1.1±0.7	1.0±0.7	5.2±2.1	1.3±0.6	1.2±0.7	5.2±2.2
ringnorm	1.1±0.3	1.2±0.5	2.8±0.7	1.1±0.3	1.2±0.4	3.3±1.2
sonar	1.4±1.2	1.9±1.7	8.1±3.9	1.3±1.4	1.4±1.6	7.0±4.1
spam	1.0±0.3	0.9±0.3	0.8±0.3	1.0±0.3	1.0±0.3	0.8±0.3
tic-tac-toe	0.9±0.5	0.8±0.5	1.3±0.8	0.9±0.5	0.8±0.6	0.6±0.5
twonorm	1.0±0.3	1.1±0.4	2.2±0.8	1.1±0.4	1.2±0.4	2.6±0.8
votes	0.8±0.8	0.8±0.9	0.7±0.9	1.1±0.8	1.0±1.0	1.0±0.8

How Large Should Binary Classification Ensembles Be? X

Median of the ensemble size and interquartil interval:

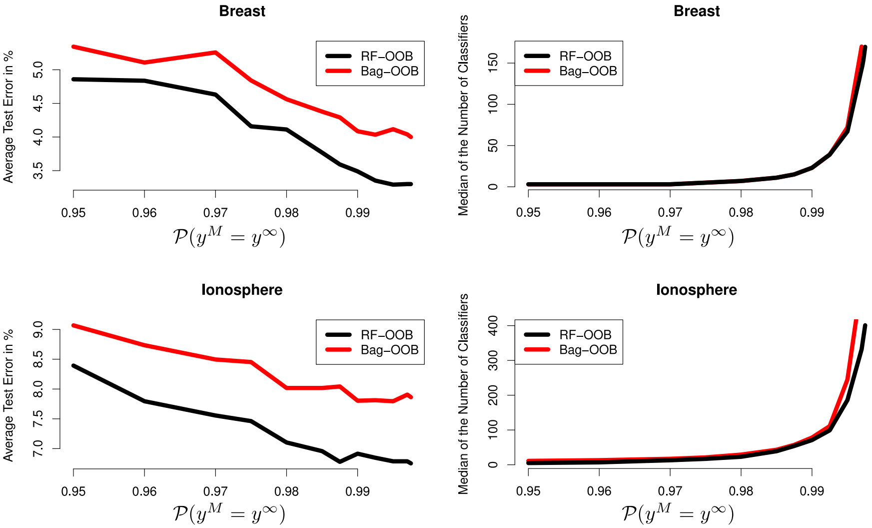
Problem	# Tree RF-Test		# Tree RF-OOB		# Tree RF-BAN	
australian	257	(192, 427)	238	(189, 318)	58	(43, 78)
breast	19	(15, 34)	23	(17, 28)	57	(36, 76)
circle	64	(46, 87)	57	(35, 87)	41	(23, 61)
echo	57	(24, 131)	88	(62, 117)	35	(18, 46)
german	1570	(1216, 2280)	1616	(1422, 2130)	78	(54, 102)
heart	529	(320, 1079)	618	(404, 1088)	47	(32, 74)
hepatitis	313	(178, 767)	532	(288, 768)	30	(20, 61)
horse	191	(126, 350)	241	(164, 368)	73	(49, 110)
ionosphere	66	(39, 100)	71	(53, 96)	41	(29, 61)
labor	64	(37, 117)	78	(53, 175)	21	(14, 37)
liver	2224	(1312, 4062)	2440	(1526, 3631)	54	(33, 81)
pima	1194	(798, 1904)	1258	(1000, 1598)	56	(36, 89)
ringnorm	563	(429, 703)	443	(346, 638)	83	(64, 111)
sonar	1975	(954, 3877)	2070	(1198, 3146)	58	(37, 85)
spam	63	(53, 72)	64	(58, 73)	90	(70, 114)
tic-tac-toe	143	(97, 195)	185	(148, 216)	116	(86, 141)
twonorm	365	(286, 428)	315	(225, 454)	96	(62, 117)
votes	20	(13, 36)	29	(19, 41)	44	(30, 61)

How Large Should Binary Classification Ensembles Be? XI

Average test error:

Problem	RF∞	RF-Test	RF-OOB	RF-BAN
australian	13.1±1.9	13.1±2.0	13.2±2.1	13.2±1.9
breast	3.2±0.9	**3.6±1.0**	**3.6±1.0**	**3.4±0.9**
circle	5.3±1.1	**5.4±1.1**	**5.4±1.2**	**5.5±1.1**
echo	9.2±3.4	**9.6±3.5**	9.2±3.5	9.5±3.5
german	24.2±1.8	24.2±1.7	24.2±1.7	**24.5±1.9**
heart	17.2±3.4	17.1±3.4	17.2±3.4	**17.9±3.6**
hepatitis	15.4±4.7	15.6±4.5	15.3±4.6	15.7±5.1
horse	14.1±2.8	14.3±2.9	14.2±2.9	**14.7±3.0**
ionosphere	6.7±2.0	6.8±1.9	**6.9±2.0**	**7.3±2.2**
labor	8.4±5.4	**9.5±5.4**	8.7±5.9	**9.9±7.4**
liver	28.2±4.0	28.2±3.9	28.4±4.0	**29.4±4.2**
pima	24.1±2.1	24.1±2.1	24.0±2.0	**24.4±2.3**
ringnorm	6.2±1.1	**6.3±1.1**	**6.3±1.2**	**6.9±1.1**
sonar	18.3±5.2	18.4±5.3	18.4±5.4	**19.4±5.0**
spam	5.0±0.6	**5.1±0.6**	5.0±0.5	**5.1±0.5**
tic-tac-toe	2.0±0.9	**2.4±0.9**	**2.2±0.9**	**2.5±1.0**
twonorm	3.8±0.7	**4.0±0.6**	**4.0±0.7**	**4.6±0.8**
votes	3.8±1.5	**4.0±1.5**	**4.0±1.5**	3.9±1.6

How Large Should Binary Classification Ensembles Be? XII



Summary

Ensemble methods...

⋆ use a **collection** of predictors whose outputs are **combined** into a response.

⋆ reduce the risk of choosing the **wrong method** or **wrong parameter values**.

⋆ often **outperform** the individual ensemble members, which must be

- ⋆ Better than random guessing.
- ⋆ Must make complementary errors.

⋆ often lead to a reduction in the **bias** and **variance** components of the error.

However, ensemble methods also

⋆ require to **store** a considerable number of predictors into **memory**.

⋆ have a **prediction time** that grows **linearly** with the **size** of the ensemble.

Finally, it is important to determine the appropriate size of an ensemble:

- ⋆ Over-estimation can result in a waste of resources.
- ⋆ Under-estimation can result in loss of prediction accuracy.

This can be done using a bound on the size M as a function of $\mathscr{P}(y^M = y^\infty)$.

References I

- Polikar, R. (2006). Ensemble based systems in decision making, Circuits and Systems Magazine, IEEE , vol.6, no.3, pp.21-45.
- Y. Koren, R. M. Bell (2011). Advances in Collaborative Filtering. Recommender Systems Handbook: 145-186
- T. Gneiting and A. E. Raftery (2005). Weather Forecasting with Ensemble Methods. Science 310 (5746), 248-249.
- J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake (2011). Real-time human pose recognition in parts from single depth images. 24th IEEE Conference on Computer Vision and Pattern Recognition, 1297-1304
- M. Ré, G. Valentini (2010). Simple ensemble methods are competitive with state-of-the-art data integration methods for gene function prediction. Journal of Machine Learning Research - Proceedings Track 8: 98-111.
- P. Viola, M. J. Jones (2004) Robust real-time face detection. International Journal of Computer Vision 57(2), 137-154.
- D. Marbach, C. Mattiussi, D. Floreano (2009). Combining Multiple Results of a Reverse-Engineering Algorithm: Application to the DREAM Five-Gene Network Challenge. The Challenges of Systems Biology: Ann. N.Y. Acad. Sci. 1158: 102-113.
- S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems. 50, 3, 602-613.
- Dietterich, T. G. (2000). Ensemble methods in machine learning. International Workshop on Multiple Classier Systems, vol. 1857 of Lecture Notes in Computer Science, pages 1-15.

References II

- Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2):123-140.
- D. Hernández-Lobato (2009). Prediction Based on Averages over Automatically Induced Learners: Ensemble Methods and Bayesian Techniques. Phd Thesis. Universidad Autónoma de Madrid.
- Breiman, L. (2001). Random forests. Machine Learning, 45(1):532.
- Díaz-Uriarte, R. and Alvarez de Andrés, S. (2006). Gene selection and classification of microarray data using random forest. BMC Bioinformatics, 7(1):3.
- Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In International Conference on Machine Learning, pages 148-156.
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5):1189-1232.
- Koren, Y. (2009). The Belkor solution to the Netflix grand prize. Online available at: `http://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf`.
- M. J. Saberian and N. Vasconcelos. In Proc. Neural Information Processing Systems (NIPS), Granada, Spain, Dec 2011.
- G. Martínez-Muñoz. Clasificación mediante conjuntos. Phd Thesis. Universidad Autónoma de Madrid, 2006.

Thank you for your attention!