# Técnicas de Aprendizaje Estadístico
# Aggregation Methods (part 2)

Mathias Bourel

IMERL - Facultad de Ingeniería, Universidad de la República, Uruguay

15 novembre 2018

## Aggregation methods

$\mathcal{L}$ the data base and $\widehat{g_1}, \ldots, \widehat{g_M}$ several predictors built over $\mathcal{L}$

$$\widehat{f} = g(\widehat{g_1}, \ldots, \widehat{g_M})$$

# Aggregation methods

## Aggregation methods

$\mathcal{L}$ the data base and $\widehat{g_1}, \ldots, \widehat{g_M}$ several predictors built over $\mathcal{L}$

$$\widehat{f} = g(\widehat{g_1}, \ldots, \widehat{g_M})$$

- Homogeneous aggregation methods (sequential and not sequential).
- Not homogeneous aggregation methods (consensus methods).

# Aggregation methods

Aggregation methods

$\mathcal{L}$ the data base and $\widehat{g_1}, \ldots, \widehat{g_M}$ several predictors built over $\mathcal{L}$

$$\widehat{f} = g(\widehat{g_1}, \ldots, \widehat{g_M})$$

- Homogeneous aggregation methods (sequential and not sequential).
- Not homogeneous aggregation methods (consensus methods).

Bias-Variance trade-off :

- Several space $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_M$ for searching predictors of different nature. However if we consider a single family $\mathcal{H}$ and predictors $\widehat{g}_1, \ldots, \widehat{g}_M \in \mathcal{H}$, the aggregated predictor $\widehat{f} = g(\widehat{g}_1, \ldots, \widehat{g}_M)$ is not necessarily a function of $\mathcal{H}$ so it could decrease the bias.
- a mean of $\widehat{g}_1, \ldots, \widehat{g}_M$ reduces the variance of the estimation.

# Plan

# Adaboost

Let $\mathcal{L} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ be a training sample where $x_i \in \mathcal{X}$ and $y_i \in \{-1, 1\}$.

1. Initialization of the weights : $w_1(i) = \frac{1}{N} \qquad i = 1, \ldots, N$.
2. For $t = 1$ to $T$ :
   - From $\mathcal{L}$ and weights $w_t(i)$, we build a predictor $h_t : \mathcal{X} \to \{-1, 1\}$ that minimizes the misclassification error :
   $$\varepsilon_t = \sum_{i=1}^{N} w_t(i) \mathbb{1}_{\{h_t(x_i) \neq y_i\}}$$
   - Calculate the classifier weight $\alpha_t = \frac{1}{2} \log\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right)$.
   - Update the weights of the observations : $w_{t+1}(i) = \frac{w_t(i) \exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}$ for all $i = 1, \ldots, N$

     where $Z_t = \sum_{i=1}^{n} w_t(i) \exp\left(-\alpha_t Y_i h_t(X_i)\right)$
3. Output. The final classifier is $f_T(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$ and the final classification rule is

$$H_T(x) = \operatorname{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) = \underset{y \in \{-1,1\}}{\operatorname{Argmax}} \left(\sum_{t=1}^{T} \alpha_t \mathbb{1}_{\{h_t(x) = y\}}\right)$$

FIGURE – *Adaboost*, Freund and Schapire, 1996

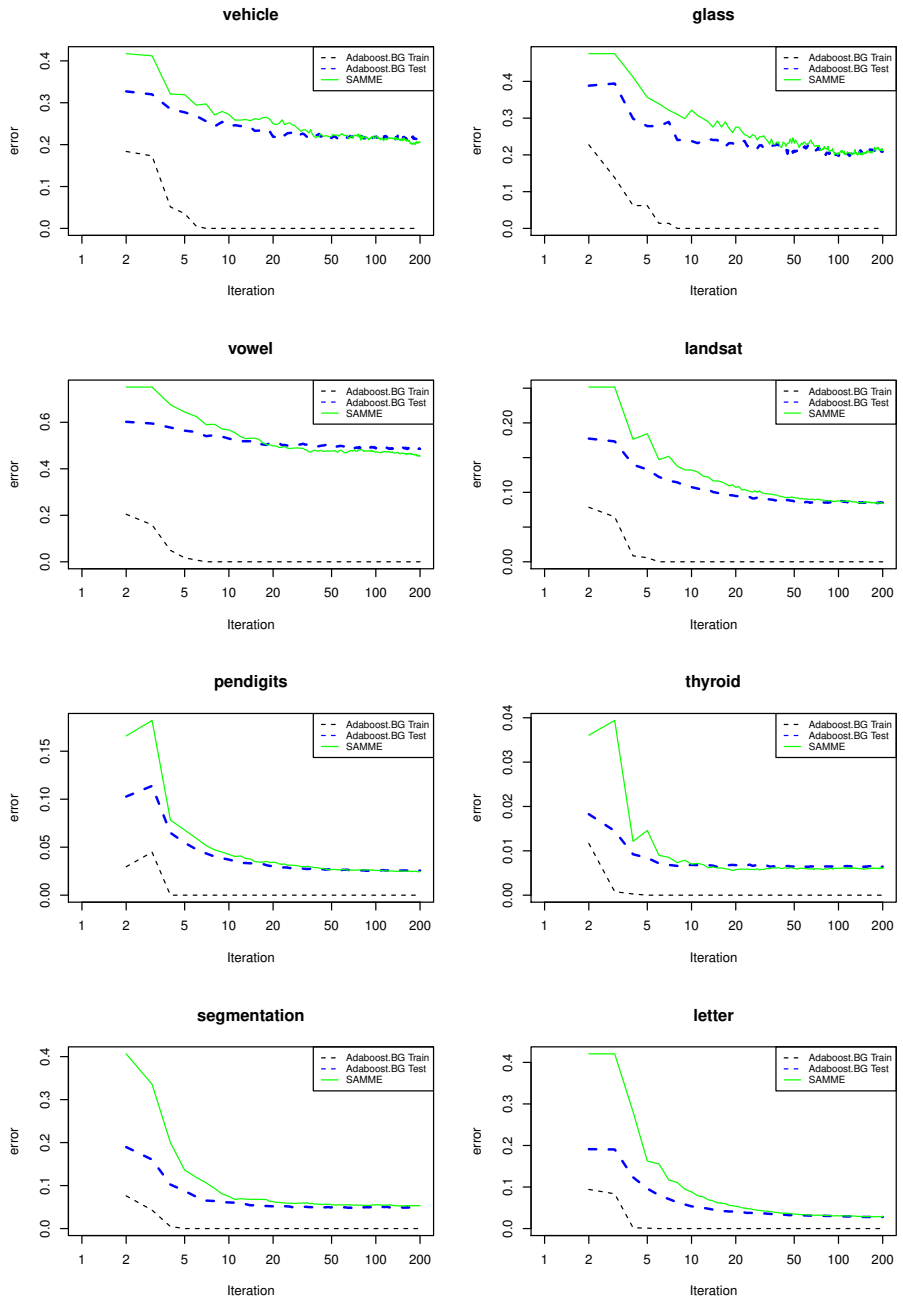
# SAMME (Adaboost for the multiclass context)

Let $\mathcal{L} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ be a training sample where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y} = \{1, \ldots, K\}$

1. Initialization of the weights : $w_1(i) = \frac{1}{N} \qquad i = 1, \ldots, N.$
2. For $t = 1$ to $T$ :
   - From $\mathcal{L}$ and weights $w_t(i)$, we build a predictor $h_t : \mathcal{X} \to \mathcal{Y}$ that minimizes misclassification the error
   $$\varepsilon_t = \sum_{i=1}^{N} w_t(i) \mathbb{1}_{\{h_t(x_i) \neq y_i\}}$$
   If $\varepsilon_t \geq 1 - \frac{1}{K}$, stop the algorithm.
   - Calculate $\alpha_t = \frac{1}{2} \log \left( \frac{1 - \varepsilon_t}{\varepsilon_t} (K - 1) \right)$.
   - Update the weights : $w_{t+1}(i) = \frac{w_t(i) \exp\left(\alpha_t \mathbb{1}_{\{h_t(x_i) \neq y_i\}}\right)}{Z_t}$ for all $i = 1, \ldots, N$ where $Z_t$ is a normalization factor.
3. Output. The final classifier is : $H_T(x) = \underset{y \in \{1, \ldots, K\}}{\text{Argmax}} \left( \sum_{t=1}^{T} \alpha_t \mathbb{1}_{\{h_t(x) = y\}} \right)$

FIGURE – **Stagewise Additive Modelling using Multiclass Exponentional loss function** (SAMME), Zhu *et al.*, 2009.

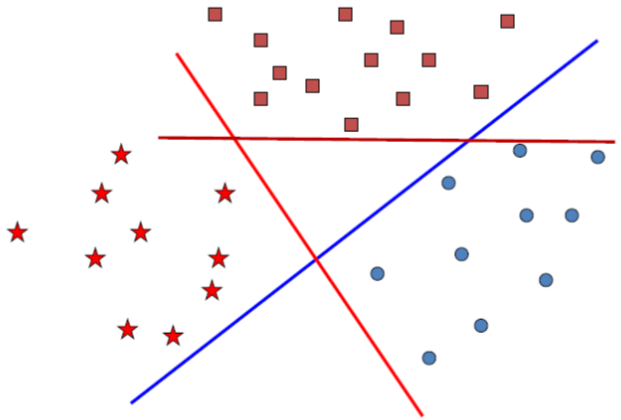
Comparison SAMME with Adaboost.BG (Bourel and Ghattas, 2017)



# Multiclass context

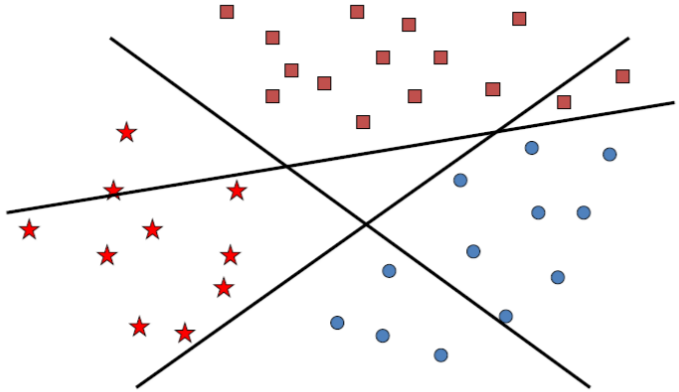
Several methods have been developed in the binary case with empirical and theory studies. But, in the case of the multiclass context, these techniques are already nowadays not sufficiently developed, and much of the methods reduced the multiclass task in several binary classifiers and combine them in different ways. We can distinguish two possibilities :

- **The one-versus-one classification** : if the problem has $K > 2$ classes, the **one-versus-one** or **all-pairs** approach consists on finding $\begin{pmatrix} K \\ 2 \end{pmatrix}$ classifiers to compare two of the $K$ classes. A test observation is classified by the $\begin{pmatrix} K \\ 2 \end{pmatrix}$ classifiers and the final assignment is done by majority vote, i.e choosing the class that most frequently appears with these classifications.



# Multiclass context

- **The one-versus-all classification** : this type of procedure construct $K$ classifiers, each of them comparing one of the $K$ classes (coded as 1) with all the others (coded as 0). If $\mathbb{P}_k(Y = k|x)$ denotes the posterior probability for $x$ to belongs to class $k$ when $k$ is coded as 1, a test observation $x$ is assigned to the class $k$ who has the highest value of $\mathbb{P}_k(Y = k|x)$.



These two methods are not very compelling, because they reduce the multiclass problem in various binary problem losing the understanding of the global problem and obviously weighing the computational calculus inefficiently.

# Plan

# Stacking for classification (Wolpert, 1992)

We fix at the beginning $M$ models $g_1, \ldots, g_M$. This is very important for the construction of the estimator.

1. Stacking for classification (Wolpert (1992)).
   We fix in advance $M$ learners $g_1, \ldots, g_M$. We split randomly the data set $\mathcal{L} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ in $J$ folds $\mathcal{L}_1, \ldots, \mathcal{L}_J$. As in cross validation for all $j = 1, \ldots, J$ we consider :
   - sets $\mathcal{L}^{(-j)} = \mathcal{L} \setminus \mathcal{L}_j$ to train the $M$ methods and obtain $\widehat{g}_1^{(-j)}, \widehat{g}_2^{(-j)}, \ldots, \widehat{g}_M^{(-j)}$ the $M$ estimators over $\mathcal{L}^{(-j)}$. We call them level-0 estimators
   - For each observation $X_i$ of $\mathcal{L}_j$ we denote by $z_{mi} = \widehat{g}_m^{(-j)}(X_i)$ the prediction of $\widehat{g}_m^{(-j)}$ over $X_i$. We obtain a vector of prediction
     $$Z_i = (z_{1i}, z_{2i}, \ldots, z_{Mi})$$
     por all $X_i \in \mathcal{L}_j$ and a new dataset $\mathcal{L}' = \{(Z_1, Y_1), \ldots, (Z_n, Y_n)\} \in \mathbb{R}^{M+1}$ of $n$ vectors. This set is called the level-1 data set.
   - With $\mathcal{L}'$ we train another model (meta model), the level-1 model and we denote it by $\mathbf{g}$.
   - With $\mathcal{L}$ we reestimate the $M$ models $g_1, \ldots, g_M$.

   The final predictor over a new observation $x$ is :
   $$f(x) = \widehat{\mathbf{g}}\big(x, \widehat{g}_1(x), \ldots, \widehat{g}_M(x)\big)$$

# Stacking for regression

A particular case of random split of $\mathcal{L}$ may be done by the leave-one-out procedure. For all $i = 1 \ldots, n$ we consider set $\mathcal{L}^{(-i)}\mathcal{L} \setminus \{(X_i, Y_i)\}$ and estimators $\widehat{g}_1^{(-i)}, \widehat{g}_2^{(-i)}, \ldots, \widehat{g}_M^{(-i)}$ trained over $\mathcal{L}^{(-i)}$. For each observation $X_i$, we obtain a new vector

$$Z_i = (\widehat{g}_1^{(-i)}(X_i), \widehat{g}_2^{(-i)}(X_i), \ldots, \widehat{g}_M^{(-i)}(X_i))$$

and the dataset of level 1 $\mathcal{L}' = \{(Z_1, Y_1), \ldots, (Z_n, Y_n)\}$.
This idea has been used by Breiman to adapt Stacking to the regression framework, where final predictor is

$$f(x) = \sum_{m=1}^{M} \alpha_m \widehat{g}_m(x)$$

where coefficients $\alpha_1, \ldots, \alpha_M$ are obtained by a **ridge regression** of $Z$ over $Y$ and $\widehat{g}_1, \ldots, \widehat{g}_M$ are estimators over initial dataset $\mathcal{L}$.

- As we mentioned before, the quantity $M$ of estimators that appear in the model built by **Stacking** is fixed in advance and this is essential in the implementation of the method. This is a significant difference from the other methods previously studied, where $M$ only affects the number of terms in the linear combination and not the components obtained.
- In Ting (1999), the authors make an interesting empirical study in the case of classification and show that Stacking is more efficient if instead of using the predictions of each estimator $\widehat{g}_1$, *dots*, $\widehat{g}_M$ of level 0, we use the posteriori probabilities of each class.

# Stacking (Smyth and Wolpert, 1999)

- $\mathcal{L} = \{x_1, \ldots, x_n\}$ and $g_1, \ldots, g_M$ fixed (KDE, different bandwiths).
- $\mathcal{L} = \mathcal{L}_1 \cup \cdots \cup \mathcal{L}_V$ and let $\mathcal{L}^{(-v)} = \mathcal{L} \setminus \mathcal{L}_v$.
- $$A = \begin{pmatrix} \widehat{g}_1^{(-1)}(\mathcal{L}_1) & \ldots & \ldots & \widehat{g}_M^{(-1)}(\mathcal{L}_1) \\ \vdots & & & \vdots \\ \widehat{g}_1^{(-V)}(\mathcal{L}_V) & \ldots & \ldots & \widehat{g}_M^{(-V)}(\mathcal{L}_V) \end{pmatrix} \in \mathcal{M}_{n \times M}$$
- We use matrix $A$ with the *EM* algorithm to estimate the coefficients of
$$\sum_{m=1}^{M} \alpha_m \widehat{g}_m(x).$$
- The output estimator is
$$\widehat{f}_{\text{stacked}}(x) = \sum_{m=1}^{M} \widehat{\alpha}_m \widehat{g}_m(x)$$
where $\widehat{g_1}, \ldots, \widehat{g_M}$ are estimate over all sample $\mathcal{L}$.

# Plan

# Random Averaged Shifted Histogram (RASH)

1. Let $\aleph$ be the original sample, $\widehat{f}_{n,0}$ be the histogram constructed over $\aleph$ and $I_{1,n}, \ldots, I_{L,n}$, where $I_{j,n} = [a_{j-1}, a_j)$, the set of the intervals of $f_{n,0}$.
2. For $m = 1$ to $M$ :
   1. Let $e_n^{(m)}$ a real random variable with density $g_n$.
   2. Set $\mathcal{I}^m = \left\{ I_{1,n}\left(e_n^{(m)}\right), I_{2,n}\left(e_n^{(m)}\right), \ldots, I_{L,n}\left(e_n^{(m)}\right) \right\}$ the modified intervals obtained by setting
   $$I_{j,n}\left(e_n^{(m)}\right) = \left[a_{j-1} + e_n^{(m)}, a_j + e_n^{(m)}\right) \quad \text{for all } j = 1, \ldots, L.$$
   3. Set $\widehat{f}_n^{(m)}$ to be the histogram constructed over $\aleph$ using the intervals in $\mathcal{I}^m$.
3. Output : $\widehat{f}_M(x) = \frac{1}{M} \sum_{m=1}^{M} \widehat{f}_n^{(m)}(x)$

FIGURE – Random Average Shifted Histogram (Bourel, Ghattas and Fraiman, 2014)

Parameter : $L$ must be given by the user.

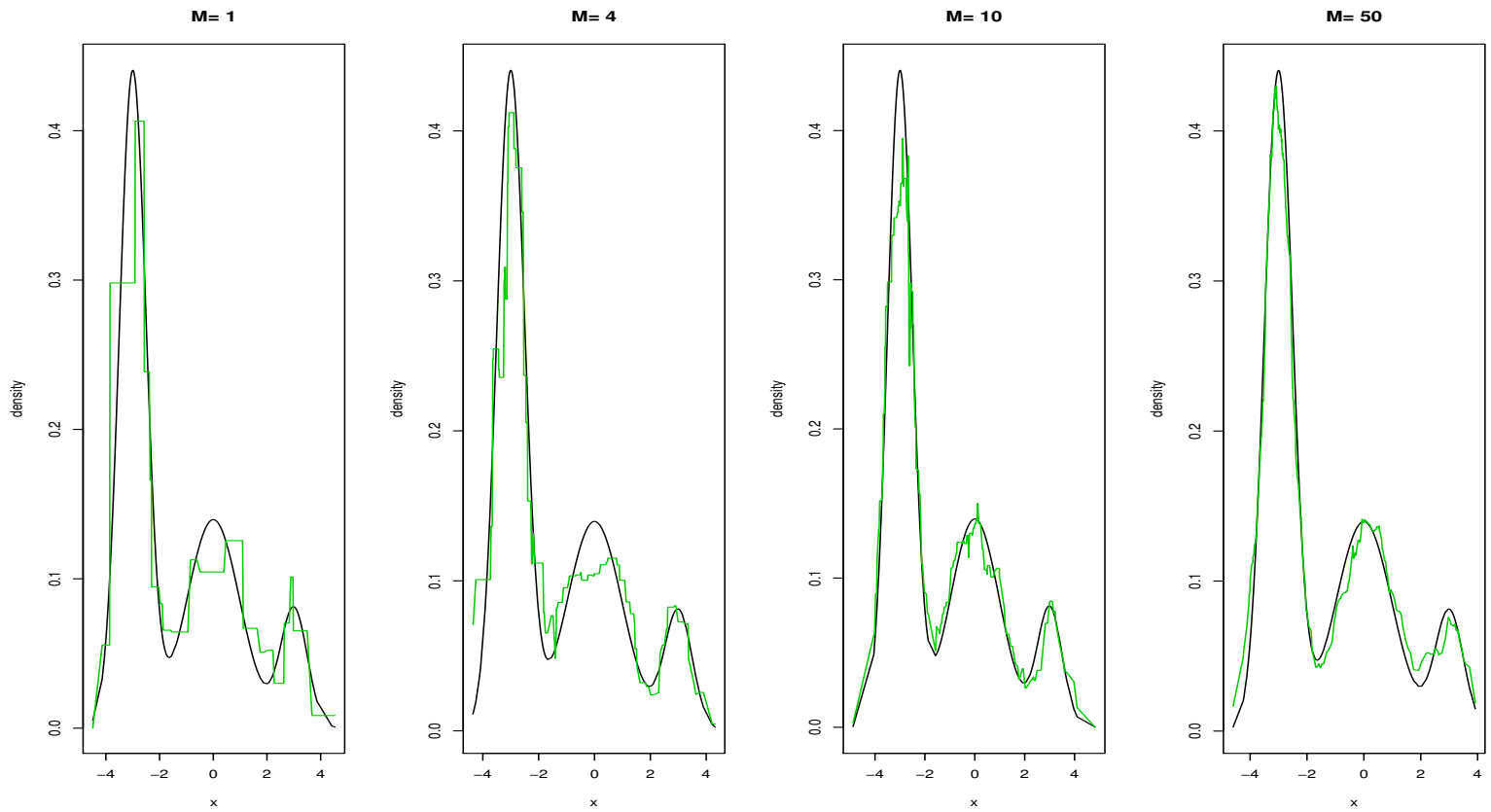



FIGURE – RASH with different values of $M$

# Bagging of Density Estimators

Inspired by Bagging of L. Breiman (1996) :

1. Sea $\mathcal{L} = \{X_1, \ldots, X_n\}$ una muestra aleatoria simple con distribución $F$ y con densidad $f$. Sea $\widehat{f}_n$ un estimador de la densidad $f$ obtenido a partir de $\mathcal{L}$.
2. Calculamos las versiones bootstrap de $\widehat{f}_n$, es decir para $b = 1, \ldots, B$ :
   1. consideramos $\mathcal{L}^* = \{X_1^*, \ldots, X_n^*\}$ una muestra bootstrap de $\mathcal{L}$ ;
   2. construimos $f_b^*$ un estimador de densidad obtenido sobre esta muestra bootstrap.
3. El estimador final es :

$$\widehat{f}^*(x) = \frac{1}{B} \sum_{b=1}^{B} f_b^*(x)$$

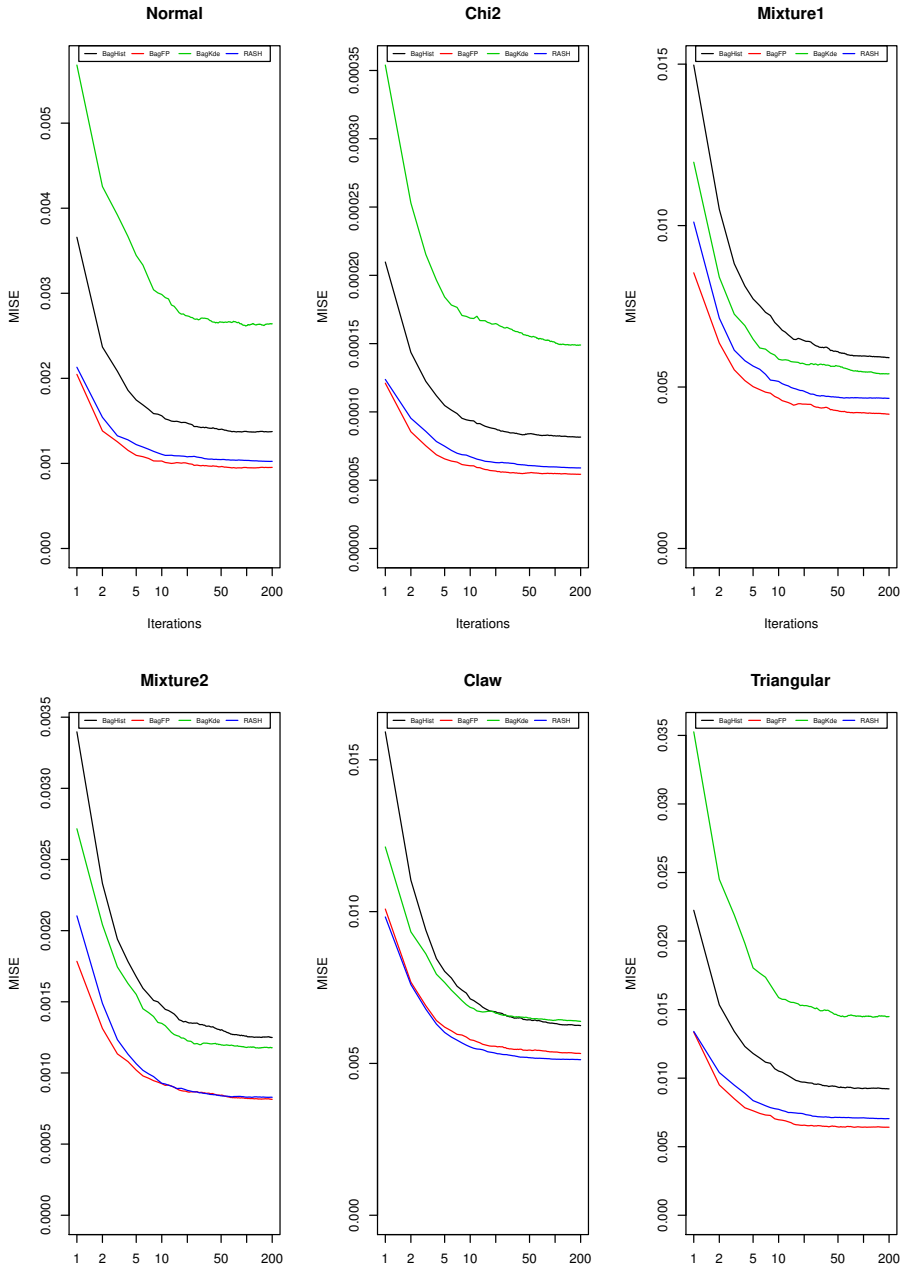
FIGURE – Bagging de estimadores de densidad (Bourel and Cugliari, 2018)

$f_b^*$ could be an histogram, a polygon frequency, a kernel density estimator.

# Bagging of Density Estimators

- Los valores óptimos de $h$ para el histograma ($h^*_{hist}$) y kde ($h^*_{kde}$) se obtienen por validación cruzada.
- Se usa el valor de $h^*_{hist}$ para FP y RASH.
- Se usa el núcleo gaussiano para kde.
- Partición de la muestra en 2/3 para entrenar, 1/3 para testear sobre $M = 100$ muestras.
- Para los métodos de agregación se utilizan $B = 200$ estimadores intermediarios.
- Criterio :

$$MISE = \int \mathbb{E}\left[\left(\hat{f}_n(x) - f(x)\right)^2\right] dx$$



# Plan

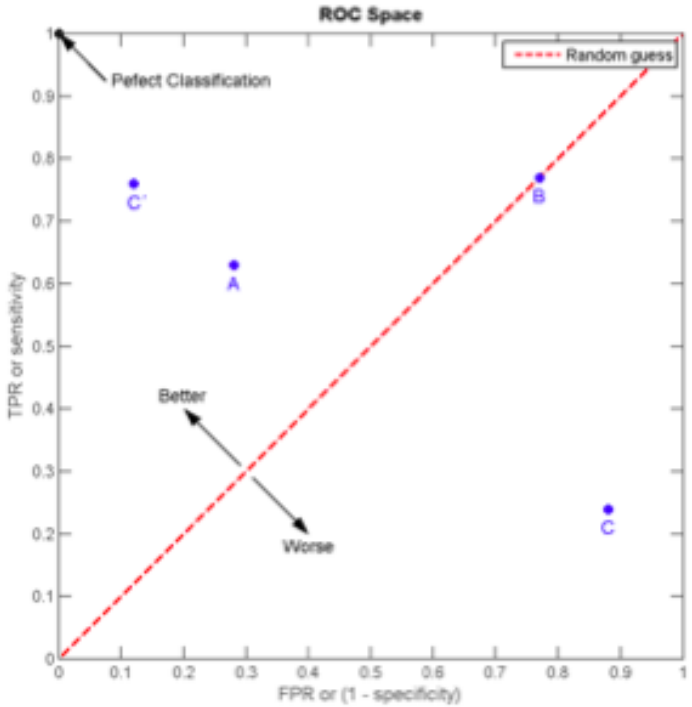
# ROC curve

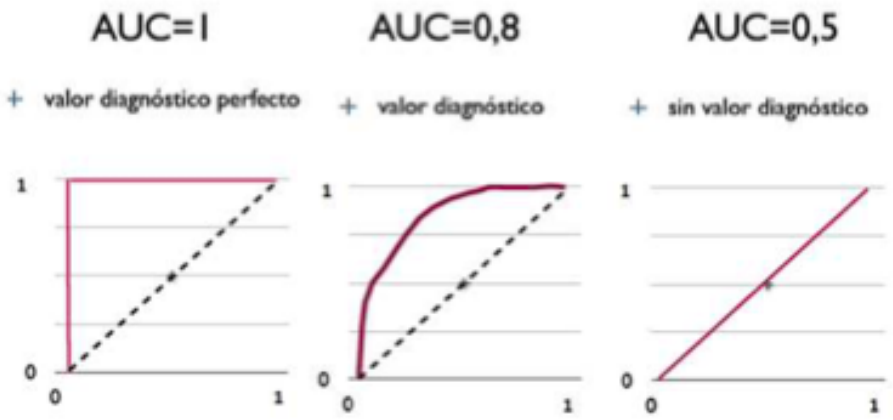
TABLE – Confusion matrix

| **Prediction \ Reality** | Positive (P) | Negative (N) |
|---|---|---|
| Positive (P') | True Positive (TP) | False Positive (FP)<br>Type 1 error |
| Negative (N') | False Negative (FN)<br>Type 2 error | True Negative (TN) |

- The sum of the elements of the first column is the number of positive instance (P) and the sum of the elements of the second column is the number of negative instance(N).
- The sum of the principal diagonal is the number of instance well classified and the sum of the other diagonal is the global error the classifier make.
- Standard performance measures associated with this matrix are the :
  1. *true positive rate (TP-rate)* or *sensitivity*, defined as $TP/P$ ;
  2. the *specificity*, defined as $TN/N$ ;
  3. the *precision*, defined as $TP/P^{'}$ ;
  4. and the $F_1$ *-measure*, defined as the harmonic mean between *sensitivity* and *precision* ; that is, $F_1 = 2 \times precision \times sensitivity/(precision + sensitivity)$.

# RocCurve


ROC Space
Random guess
Pefect Classification
C'
B
A
Better
Worse
C
TPR or sensitivity
FPR or (1 - specificity)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


AUC=1
+ valor diagnóstico perfecto
AUC=0,8
+ valor diagnóstico
AUC=0,5
+ sin valor diagnóstico
1
0
0
1

# Consensus Methods (Bourel, Crisci and Martinez, 2017)

A way of doing a mixture of experts is inspired to some extent on Bayesian Voting, and it consists in assigning a weight to each hypothesis. An hypothesis $h$ generally calculates the posterior probability that a given observation belongs to a class. To fix notation, we can think that $h$ computes a vector $\left(p_0^h(\mathbf{x}), p_1^h(\mathbf{x})\right)$ where $p_0^h(\mathbf{x})$ and $p_1^h(\mathbf{x})$ are the posterior probabilities that observation $\mathbf{x}$ belongs to class 0 or to class 1 respectively. Noting by $\mathcal{L}$ the sample, by $k \in \{0, 1\}$ a class, by $f$ the combined classifier and by $\mathcal{H}$ the class of functions from where we choose the hypothesis, we have :

$$\mathbb{P}(f(\mathbf{x}) = k|\mathbf{x}, \mathcal{L}) = \sum_{h \in \mathcal{H}} \mathbb{P}(h|\mathcal{L})h(\mathbf{x}) = \sum_{h \in \mathcal{H}} w_{h,\mathcal{L}}\, h(\mathbf{x})$$

So the probability of assigning class $k$ to $\mathbf{x}$ by the final predictor $f$ is a convex average with weights $w_{h,S}$ over all the functions of $\mathcal{H}$. This fact explicitly provides an expression between the final classifier and a weighted average of predictions of several classifiers.
For different intermediate fixed classifiers $h_1, \ldots, h_M$, we define :

$$F(\mathbf{x}) = \underset{k \in \{0,1\}}{\text{Argmax}} \left( \sum_{m=1}^{M} w_{h_m,\mathcal{L}}\, p_k^{h_m}(\mathbf{x}) \right)$$

This kind of combination is called a weighted averaging combining rule.

# Consensus Methods (Bourel, Crisci and Martinez, 2017)

- **Majority Vote (MV)**. For a given observation, the $M$ classifiers make the class prediction. The consensus consists in choosing the class that receives more votes.
- **Mean Probability (MeanProb)**. For a given observation, we will obtain two vectors of probabilities of length $M$ : one with the probabilities of belonging to class 0 ($v_0$), and the other with the probabilities of belonging to class 1 ($v_1$). This pair of vectors will be obtained for each of the single methods. Finally, we average the probabilities of $v_0$ on one side and $v_1$ on the other side along with the single models. The predicted class is the one that presents the highest mean probability.

# Consensus Methods (Bourel, Crisci and Martinez, 2017)

- **Weighted Average AUC (WA-AUC)**. The sample is divided into two parts : one for training the model (two thirds of the original learning sample) and the other (the remaining one third) for computing its accuracy. For a given observation, we consider the weighted mean of the posteriori probabilities of different single models (constructed with the two thirds of the learning sample). The weights are calculated as the area under the ROC curve (AUC) of each method (obtained with the remaining one third of the learning sample). After the normalization, the WA-AUC expressions are as follows :

$$\text{WA-AUC}_0(\mathbf{x}) = \sum_{m=1}^{M} AUC_m \times p_0^{h_m}(\mathbf{x}) \quad \text{and} \quad \text{WA-AUC}_1(\mathbf{x}) = \sum_{m=1}^{M} AUC_m \times p_1^{h_m}(\mathbf{x})$$

where $AUC_m$ is the area under the ROC curve of the hypothesis $h_m$ and $p_k^{h_m}(\mathbf{x})$ is the posterior probability of the observation $\mathbf{x}$ belonging to the class $k$ for the hypothesis $h_m$. The consensus for an observation $\mathbf{x}$ is 0 if $\text{WA-AUC}_0(\mathbf{x}) > \text{WA-AUC}_1(\mathbf{x})$ or 1 otherwise.

# Consensus Methods (Bourel, Crisci and Martinez, 2017)

- **Stacking with GLM (StackGLM)**. **Stacking** consists (generally via a cross-validation process) in fitting several base classifiers and using their predictions to compute a new learning sample to train another classifier (called **meta-classifier**). As in Ting, we used the class probabilities rather than the class predictions of the single methods and we considered a logistic regression as a meta-classifier. For each class, we compute

$$LR_0(\mathbf{x}) = \sum_{m=1}^{M} \alpha_{m0} p_0^{h_m}(\mathbf{x}) \quad \text{and} \quad LR_1(\mathbf{x}) = \sum_{m=1}^{M} \alpha_{m1} p_1^{h_m}(\mathbf{x})$$

  where the coefficients $\alpha_m$ are obtained by minimizing a mean least squared optimization problem as in linear regression. The observation $\mathbf{x}$ is assigned to class 1 if $LR_1(\mathbf{x}) > LR_0(\mathbf{x})$ or to class 0 otherwise.
- **Stacking with Random Forests (StackRF)**. Inspired by the idea of StackGLM, we apply the same method by considering RF as a meta-classifier instead of the regression considered above. This provides a stacking that is not linear on its final output.

# Consensus Methods (Bourel, Crisci and Martinez, 2017)

- **Weighted Average Prediction Error (WA-PE)**. This method gives a classifier that is a linear combination of weighted single classifiers. As in WA-AUC, the sample is divided into two parts : for each model $m$, the first part is used for training the model and the second is used for computing the accuracy $w_m$. After normalizing the weights $w_m$ of each classifier, we compute

$$\text{WA-PE}_0(\mathbf{x}) = \sum_{m=1}^{M} w_m p_0^{h_m}(\mathbf{x}) \quad \text{and,} \quad \text{WA-PE}_1(\mathbf{x}) = \sum_{m=1}^{M} w_m p_1^{h_m}(\mathbf{x})$$

  The observation $\mathbf{x}$ is assigned to class 1 if $\text{WA-PE}_1(\mathbf{x}) > \text{WA-PE}_0(\mathbf{x})$ or to class 0 otherwise.

# Consensus Methods (Bourel, Crisci and Martinez, 2017)

| | GLM | RF | Boosting | SVM | VM | MeanProb | WA-AUC | StackGLM | StackRF | WA-PE |
|---|---|---|---|---|---|---|---|---|---|---|
| *A. fraterculus* | 19.51 ± 5.4 | **15.85** ± 3.9 | 20.31 ± 4.2 | 16.15 ± 3.7 | 16.65 ± 4.0 | 17.32 ± 4.6 | 16.43 ± 4.2 | 19.88 ± 4.0 | 18.21 ± 3.9 | 16.02 ± 4.3 |
| *A. sanguinea* | 17.57 ± 10.1 | 9.32 ± 3.6 | 9.38 ± 3.6 | 10.86 ± 3.4 | 9.54 ± 3.7 | 10.37 ± 3.5 | 9.94 ± 3.7 | 8.95 ± 3.6 | **8.8** ± 3.4 | 10.31 ± 3.4 |
| *A. guyunusae* | 21.75 ± 4.6 | 22.86 ± 3.9 | 26.71 ± 4.2 | 21.94 ± 4.4 | **20.77** ± 3.9 | 20.86 ± 4.3 | 21.08 ± 4.4 | 26.71 ± 4.4 | 24.37 ± 4.2 | 21.08 ± 4.4 |
| *C. pelagica* | 35.72 ± 4.7 | 31.41 ± 5.2 | 34.09 ± 5.5 | 36.37 ± 4.6 | 32.65 ± 6 | 31.51 ± 5.4 | 31.45 ± 5.4 | 33.94 ± 5.4 | 32.89 ± 5.6 | **31.35** ± 5.3 |
| *D. acuminata* | 41.23 ± 5.7 | 38.34 ± 4.8 | 41.88 ± 5.8 | 39.69 ± 5.6 | 38.92 ± 4.5 | 38.12 ± 4.4 | 38.12 ± 4.5 | 41.81 ± 5.7 | 40.86 ± 5.3 | **38.0** ± 4.5 |
| *L. danicus* | 26.49 ± 4.9 | 27.35 ± 5.4 | 30.06 ± 5.5 | **25.97** ± 4.4 | 26.83 ± 5.4 | 27.41 ± 5.8 | 27.32 ± 5.5 | 30.12 ± 5.6 | 29.11 ± 5.7 | 27.48 ± 5.5 |
| *R. setigera* | 31.17 ± 5.2 | 30.15 ± 3.8 | 33.14 ± 4.4 | 30.4 ± 5.4 | 29.79 ± 4.8 | 30 ± 5.0 | 30.59 ± 5.2 | 32.89 ± 4.6 | 31.82 ± 4.6 | **29.61** ± 5.2 |
| *T. nitzschioides* | 34.74 ± 5.7 | 34.77 ± 5.9 | 39.85 ± 4.8 | 36.06 ± 5.1 | 34.28 ± 6.1 | 34.22 ± 6.1 | 34.09 ± 6.2 | 39.91 ± 4.8 | 37.41 ± 4.9 | **34.01** ± 6.1 |

TABLE – Mean generalization error (±*SD*) on the eight marine phytoplankton presence-absence data sets with their standard deviation over 50 test samples. For each data set, the errors with bold emphasis correspond to the model with the lowest error.

| | GLM | RF | Boosting | SVM | VM | MeanProb | WA-AUC | StackGLM | StackRF | WA-PE |
|---|---|---|---|---|---|---|---|---|---|---|
| Blood Transfusion | 22.74 ± 2.6 | 24.44 ± 2.1 | 32.48 ± 3.4 | 22.92 ± 2.6 | 24.26 ± 2.1 | 22.34 ± 2.5 | 22.28 ± 2.4 | 27.95 ± 2.7 | 26.27 ± 2.5 | **22.2** ± 2.6 |
| Credit Approval | 22.78 ± 18.3 | 13.38 ± 3.2 | 14.83 ± 3.4 | 13.61 ± 3.2 | 16.72 ± 8.7 | 13.32 ± 3.2 | 13.27 ± 3.2 | 14.81 ± 3.4 | 14.3 ± 3.4 | **13.25** ± 3.2 |
| Default | 16.86 ± 2.6 | 2.89 ± 0.2 | 4.87 ± 0.5 | 2.76 ± 0.2 | **2.75** ± 0.2 | 3.19 ± 0.2 | 3.18 ± 0.3 | 3.65 ± 0.4 | 3.73 ± 0.4 | 3.29 ± 0.3 |
| Housing | 17.21 ± 3.8 | 13.32 ± 2.4 | 12.46 ± 2.4 | 12.81 ± 2.0 | 12.44 ± 2.2 | 12.25 ± 1.9 | 12.23 ± 1.9 | 12.58 ± 2.4 | 12.51 ± 2.3 | **12.24** ± 1.9 |
| Liver Disorders | 33.65 ± 4.3 | 29.63 ± 4.2 | 31.61 ± 4.4 | 33.4 ± 5.2 | 29.75 ± 4.7 | 28.77 ± 4.0 | 28.67 ± 3.9 | 31.56 ± 4.4 | 30.67 ± 4.6 | **28.65** ± 3.8 |
| MAGIC | 18.18 ± 0.4 | **11.77** ± 0.3 | 17.59 ± 0.6 | 14.39 ± 0.4 | 12.79 ± 0.4 | 15.89 ± 0.4 | 15.45 ± 0.4 | 12.08 ± 0.4 | 11.81 ± 0.3 | 15.66 ± 0.4 |
| Orange Juice | **17.26** ± 1.8 | 19.68 ± 2 | 21.89 ± 2 | 17.74 ± 1.7 | 18.4 ± 1.8 | 18.99 ± 2.1 | 18.99 ± 2.1 | 21.62 ± 2.2 | 21.18 ± 1.9 | 18.99 ± 2.0 |
| Parkinsons | 22.74 ± 5.6 | 12.49 ± 4.6 | **10.65** ± 3.8 | 14.28 ± 4.3 | 13.23 ± 4.5 | 14.0 ± 4.0 | 13.29 ± 4.2 | 18.74 ± 8.7 | 11.66 ± 4.1 | 13.72 ± 3.9 |
| QSAR biodegradation | 30.74 ± 5.9 | 14.94 ± 2.0 | 15.05 ± 1.9 | 16.99 ± 3.1 | 14.76 ± 1.8 | 14.35 ± 1.9 | 14.36 ± 1.9 | 16.08 ± 7.4 | 14.76 ± 1.7 | **14.34** ± 1.9 |
| Spam | 31.96 ± 1.9 | 5.35 ± 0.5 | 5.32 ± 0.7 | 7.29 ± 0.6 | 5.56 ± 0.6 | 5.63 ± 0.6 | 5.56 ± 0.6 | 5.16 ± 0.7 | **5.08** ± 0.6 | 5.55 ± 0.6 |

TABLE – Mean generalization errors (±*SD*) on the 10 open access data sets with their standard deviation over 50 test samples. For each data set, the errors with bold emphasis correspond to the model with the lowest error.

# Bibliography


1. Bourel, M. and Cugliari, J., Bagging of density estimators, In Edition, https://arxiv.org/abs/1808.03447
2. Bourel, M. and Ghattas, B., Direct Multiclass boosting using base classifiers' posterior probabilities estimates, **Proceedings of International Conference on Machine Learning and Applications** (IEEE ICMLA 17), Mexico, 2017.
3. Bourel, M, Segura, A., Multiclass classification methods in Ecology, **Ecological Indicators**, Vol. 85, p. 1012-1021, 2018.
4. Bourel, M., Crisci, C., Martinez, A., Consensus methods based on machine learning techniques for marine phytoplankton presence-absence prediction, **Ecological Informatics**, Vol. 42, p. 46 - 54, 2017
5. Bourel, M., Ghattas, B., Fraiman, R., Random Averaged Shifted Histogram, **Computational Statistics and Data Analysis**, Vol. 79, p.149-164, 2014.
6. Bourel, M. Ghattas, B., Aggregating density estimators : an empirical study, **Open Journal of Statistics**, ISSN 2161-7198, Vol. 3 (5), p. 344-355, 2013.
7. Bourel, M., Agrégation de modèles en apprentissage statistique pour l'estimation de la densité et la classification multiclasse, **Tesis de doctorado**, Université Aix-Marseille, 2013.
8. Bourel, M., Métodos de Agregación de modelos y aplicaciones, **Mémorias de trabajos de difusión científica y técnica**, Vol. 10, p. 19-32, 2012.
9. Breiman, L., Stacked Regression, **Machine Learning**, 24 (1), p. 49–64, 1996.
10. Breiman, L., Bagging predictors, **Machine Learning**, 24, 123 ?140, 1996.
11. James, Witten, Hastie, Tibshirani. An introduction to Statistical Learning with application in R, Springer, 2013.
12. Freund, Y. and Schapire, E., A decision-theoretic generalization of on-line learning and application to boosting, **Journal of Computer and System Sciences**, 55(1) : p 119-13, 1997.
13. Hastie, Tibshirani, Friedman. The Elements of Statistical Learning, Springer, 2003.
14. Marmion, M., Parviainen, M., Luoto, M., Heikkinen, R. K., Thuiller, W., Evaluation of consensus methods in predictive species distribution modelling, **Diversity and Distributions**,15 (1), p. 59–69, 2009.
15. Ting, K. M. and Witten, I. H., Issues in Stacked Generalization, **Journal of Artificial Intelligence Research**, 10, p. 271–289, 1999.
16. Schapire, R.E and Freund, Y., Boosting : Foundations and Algorithms. Adaptive Computation and Machine Learning Series. Mit Press, 2012.
17. Smyth,P. y Wolpert, D.H ; Linearly combining density estimators via stacking, **Machine Learning**, 36(1-2) : p. 59 - 83, 1999.
18. Wolpert, D.H., Stacked Generalization, **Neural Networks**, 5, p. 241–259, 1992.
19. Zhu, J., Zou, H. y Rosset, S., Hastie, T ; Multi-class Adaboost, **Statistics and its Interface**, 2, p. 349-360, 2009.