Tailored Aggregation for Classification

Tristan Mary-Huard and Stéphane Robin

Abstract—Compression and variable selection are two classical strategies to deal with large-dimension data sets in classification. We propose an alternative strategy, called aggregation, which consists of a clustering step of redundant variables and a compression step within each group. We develop a statistical framework to define tailored aggregation methods that can be combined with selection methods to build reliable classifiers that benefit from the information contained in redundant variables. Two algorithms are proposed for ordered and nonordered variables, respectively. Applications to the kNN and CART algorithms are presented.

Index Terms—Classification, aggregation, selection, large-dimension data, ordered variables.

1 Introduction

Classification is an important field of investigation that has received a lot of attention from computational scientists and statisticians in the last decades. With the recent development of high-throughput technologies, for example, in molecular biology, many strategies have been developed to deal with large-dimension data, where the number of observations n is lower than the number of variables p. In particular, variable selection and compression have become popular strategies to reduce the dimension of the data.

In large-dimension data problems, selection methods allow the discarding of most of the variables to obtain a classifier based on a few variables only, but the selection process may be unstable, i.e., the list of selected variables may depend on the sampling of the original data set into a train sample and a test sample. Compression methods build combinations of the initial variables according to the label that can be more stable according to the sampling, but the new variables are combinations of all the old variables. Consequently, neither of the two strategies leads to interpretable results.

1.1 Limits of Variable Selection

In the classification context, it has been pointed out that space dimensionality reduction by variable selection (or feature selection) could lead to more efficient classifiers. The literature on the topic is abundant; the reader may consult Fukumizu et al. [15], Hastie et al. [20], Krishnapuram et al. [22], and Xiong et al. [33] for some theoretical and practical considerations. In most of the articles on variable selection, three purposes are frequently mentioned to motivate variable selection.

1. Variable selection allows one to get rid of variables that are uninformative about the label.
2. Variable selection reduces the data dimension and, consequently, the computational burden.
3. Reducing the number of variables results in an interpretable classification rule.

In practice, purposes 1 and 2 are achieved using variable selection (Guyon and Elisseeff [16], Dudoit et al. [12]), while the interpretation purpose is not, especially for large data sets where redundancy is high. Indeed, variable selection provides a limited list of variables that may seem to be easy to interpret. However, this ease of interpretation is fallacious, as shown by Michiels et al. [27] on a microarray example. This paper shows that the list of variables could drastically change according to the way the data set is split into training and test sets, which makes the results actually uninterpretable.

1.2 Toward a Tailored Aggregation

What can be done to deal with redundancy? In the selection perspective, removing informative predictor that is correlated to selected ones makes sense since it removes redundancy. However, this introduces some randomness in the process, while combining correlated (i.e., redundant) predictors could be more effective, especially when predictors are noisy. Combining correlated predictor will keep all informative variables in the final model without increasing the number of parameters since only one parameter will be estimated for each group of correlated predictors.

This reasoning motivates considering a variable aggregation step before the variable selection step. In the aggregation step, variables that share common information would be classified into the same cluster and summed up by a summarized variable and, in the selection step, informative summarized variables would be selected. In the following, methods that cluster variables into groups according to their information about the label and produce a summarized variable per group will be called *aggregation* methods. We emphasize that interpretability and classification performance are two different goals. In this paper, we are concerned about interpretability. Therefore, the classification error rate will not be the gold standard to evaluate the strategy we propose.

Variable clustering is a classical problem in statistics (Anderberg [2], Harman [17]), but the aggregation problem is much more recent. The GeneShaving method proposed in Hastie et al. [19] is perhaps the best known aggregation method. The principle of the algorithm is to find a linear combination of the variables that is closely correlated to the label. The linear combination is then "shaved" to discard variables that are not or are poorly correlated with the initial combination. Then, a second linear combination, orthogonal to the first one, is built and shaved, and so on. Dettling and Bühlmann [10] and Dettling [9] present two algorithms, Wilma (*Wil*coxon and *Ma*rgin criteria) and Pelora (*Pe*nalized *Lo*gistic *R*egression *A*nalysis) that aim to build small groups of variables with highly predictive profiles. The predictive potential of a group of variables is estimated by the Wilcoxon test or the penalized logistic regression. The most informative variable groups are then combined with different classification algorithms to produce a classifier. Diaz-Uriarte [11] proposed a method that identifies signature components that are then combined with a classification method.

According to the Wrapper/Filter typology used in the learning community (see Kohavi and John [21]), all of these methods are "Filter" methods: The variable aggregation criterion is independent from the classification algorithm that will be used in the following step to build the classifier. While wrapper methods are not preferable to filter methods a priori, we would like to guarantee the coherence between the aggregation and selection steps, in order to build an efficient classifier. The Wrapper terminology is not well established: it is sometimes dedicated to strategies that are tailored for the classification algorithm and sometimes to strategies that use both the classification algorithm and the labels. Although we consider the later definition too restrictive, we will use the term "tailored" in place of "wrapper" throughout this paper. Here, we propose a statistical framework for tailored aggregation, defined according to the following guidelines:

- The method should aggregate variables according to their information, whatever the pertinence of the information.

- *The authors are with UMR AgroParisTech/INRA, MIA 518, 16 rue Claude Bernard, 75231 Paris Cedex 05, France. E-mail: {maryhuar, robin}@agroparistech.fr.*

Manuscript received 13 Mar. 2008; revised 10 July 2008; accepted 18 Feb. 2009; published online 4 Mar. 2009.
Recommended for acceptance by M. Figueiredo.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-03-0148.
Digital Object Identifier no. 10.1109/TPAMI.2009.55.

The selection of the informative summarized variables will be performed during the selection step.
- The aggregation step should explicitly take into account the classification algorithm that will be used to build the classification rule.

The paper is organized as follows: Section 2 presents the principle of the tailored aggregation strategy, along with some algorithms to apply this strategy. Section 3 presents four applications of the method on real data and Section 4 discusses different ways to extend the strategy to other classification algorithms.

2 METHODS

We briefly introduce the classification framework. The aim of classification is to predict the unknown class label Y of an observation, according to some information X collected on this same observation. In the following, we suppose that the class label is either 0 or 1 and X is a vector of p variables $(X^1, \ldots, X^p)$. We have to build a classifier on the basis of some training data (X_i, Y_i), $i = 1 \ldots n$, for which both the information X_i and the true label Y_i are known.

We first present the tailored aggregation objective as a universal optimization problem. We then present two applications of this general formulation to the kNN and CART algorithms.

2.1 The Optimization Program

In order to formally describe the aggregation objective, we introduce some important definitions for the following.

Definition 1. *An aggregation A consists in splitting up the p initial variables into N_C clusters $\mathcal{C}_1, \ldots, \mathcal{C}_{N_C}$, and to sum up the information in each cluster with a linear combination of the cluster variables:*

$$Z^{\mathcal{C}_i} = \sum_{X^j \in \mathcal{C}_i} \alpha_j X^j.$$

We suppose that the classifier will be built using a classification algorithm alg. Furthermore, we suppose that the information we lose during the aggregation step by replacing the p initial variables with the N_C variables $Z^{\mathcal{C}_i}$ can be quantified by a loss function:

$$\begin{aligned} &L_{alg} : \mathcal{A} \to \mathbb{R} \\ &\quad A \mapsto L_{alg}(A) = L_{alg}\big(\{X^1, \ldots, X^p\}, \quad \{Z^{\mathcal{C}_1}, \ldots, Z^{\mathcal{C}_{N_C}}\}\big), \end{aligned}$$

where $\mathcal{A}$ is the set of all possible aggregations. The loss L_{alg} is defined according to the classification algorithm, ensuring the tailored property of the aggregation method. Given this loss, we can define our aggregation objective as an optimization problem as given below.

Proposition 1. *Let $(X_1, Y_1), \ldots, (X_n, Y_n)$ be n observations where X_i is a vector of p variables $X^1, \ldots, X^p$. Let N_C be a given integer such that $N_C < p$. The optimal aggregation A^* of size N_C for the chosen classification algorithm is the aggregation that satisfies*

$$A^* = \underset{A \in \mathcal{A}}{\operatorname{Argmin}}\ L_{alg}\big(\{X^1, \ldots, X^p\}, \quad \{Z^{\mathcal{C}_1}, \ldots, Z^{\mathcal{C}_{N_C}}\}\big).$$

In the two following applications, we describe how to define the loss function for the kNN and the CART algorithm.

2.2 Aggregation for kNN

The principle of the kNN algorithm (Fix and Hodges [13], Fix and Hodges [14]) is the following: To predict the label of individual x_0, the k nearest neighbors of x_0 in the training set are consulted and x_0 is classified according to the majority label between the neighbors. Importantly, the algorithm has two successive steps: the tessellation step where the neighbors are found and the labeling step. We can observe that variables $X^1, \ldots, X^p$ are only used in the tessellation step, while the labels are only used in the labeling step. Therefore, we can define the redundancy between variables by only considering the tessellation step: Two variables that produce the same tessellations will provide identical prediction for any point x_0. We can define the redundancy for the kNN as given below.

Definition 2. *Two variables are redundant for the kNN algorithm if their contributions to distances between points are the same.*

From this definition, we can define the loss function for the kNN as a loss of information about distances. Distance conservation can be measured via inertia:

$$I(X^1, \ldots, X^p) = \frac{1}{2n} \sum_{i=1}^{n} \sum_{j=1}^{n} d^2(X_i, X_j),$$

where d is the distance measure chosen for the kNN algorithm. The inertia criterion is used in many popular algorithms, such as Principal Component Analysis (PCA), and its properties have been largely investigated. In particular, inertia is known to decrease when variables are aggregated (Saporta [30]). We can use this property to formally define the loss function for the kNN.

Definition 3. *The kNN loss associated to aggregation A is quantified by the decrease in inertia due to this aggregation:*

$$\begin{aligned} &L_{kNN}(\{X^1, \ldots, X^p\}, \quad \{Z^{\mathcal{C}_1}, \ldots, Z^{\mathcal{C}_{N_C}}\}) \\ &\quad = I(X^1, \ldots, X^p) - I(Z^{\mathcal{C}_1}, \ldots, Z^{\mathcal{C}_{N_C}}). \end{aligned}$$

Now if we suppose that d is the classical euclidian distance, then inertia is additive:

$$\begin{aligned} I(X^1, \ldots, X^p) &= \frac{1}{2n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{p} \left(X_i^k - X_j^k\right)^2 \\ &= \frac{1}{2n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{\ell=1}^{N_C} \sum_{X^{\ell_1}, \ldots, X^{\ell_q} \in \mathcal{C}_\ell} \left(X_i^{\ell_1} - X_j^{\ell_1}\right)^2 \\ &= \sum_{\ell=1}^{N_C} \sum_{X^{\ell_1}, \ldots, X^{\ell_q} \in \mathcal{C}_\ell} I(X^{\ell_1}, \ldots, X^{\ell_q}). \end{aligned}$$

This additive property can be extended to the loss function L_{kNN} and the optimization program of Proposition 1 becomes as given below.

Proposition 2. *The optimal aggregation A^* of size N_C for the kNN algorithm satisfies*

$$A^* = \underset{A \in \mathcal{A}}{\operatorname{Argmin}} \sum_{\ell=1}^{N_C} \sum_{X^{\ell_1}, \ldots, X^{\ell_q} \in \mathcal{C}_\ell} L_{kNN}(\{X^{\ell_1}, \ldots, X^{\ell_q}\},\ Z^{\mathcal{C}_\ell}).$$

According to Definition 5, $Z^{\mathcal{C}_\ell}$ is defined for each cluster $\mathcal{C}_\ell$ as the linear combination of variables $X^{\ell_1}, \ldots, X^{\ell_q}$ that minimizes the loss of inertia. The optimal linear combination is the first principal component of the PCA performed on variables $X^{\ell_1}, \ldots, X^{\ell_q}$. The optimization program is completely defined for the kNN algorithm by choosing $Z^{\mathcal{C}_\ell}$ as the first component of the cluster variables.

2.3 Aggregation for CART

The Classification and Regression Tree algorithm (CART) has become very popular since the seminal work of Breiman et al. [7]. We present here the basics of the CART algorithm; the reader may consult Hastie et al. [20] for a complete description. Using CART, an observation is classified according to its answers to successive questions, where all the questions are of the form "is the value of variable X^j higher than a threshold s_j?" Such a classification rule

can be represented by a binary tree. Each intermediate node represents a question and each terminal node (or leaf) is the predicted label.

The goal of the training step is to construct an optimal tree, for which we need:

- to build a complete (large) tree, i.e., to define for each node an optimal combination of a variable X^j and a threshold s_j,
- to prune the complete tree. In practice, the pruning step is performed using a model selection criterion.

The pruning steps occur after the choice of the variables, so we only need to focus on the first of the two steps to define redundancy for CART. To formulate the first question, the procedure is the following: each combination of a variable X^j and a threshold s^j splits the training sample into two subsamples E_{yes} and E_{no}, according to the answer of each training observation to the question. We look for the question that optimizes the purity of two subsamples:

$$I(E) - \frac{n_{yes}}{n} I(E_{yes}) - \frac{n_{no}}{n} I(E_{no}),$$

where n_{yes} and n_{no} are the sizes of the two subsamples. $I(E)$ is usually one of the following criteria:

$$\begin{array}{ll} I(E) = \hat{\pi}_0 \ln(\hat{\pi}_0) + \hat{\pi}_1 \ln(\hat{\pi}_1), & \text{deviance or entropy criterion,} \\ I(E) = \hat{\pi}_0 \hat{\pi}_1, & \text{Gini index,} \\ I(E) = \min(\hat{\pi}_0, \hat{\pi}_1), & \text{classification error rate,} \end{array}$$

with $\hat{\pi}_0$ and $\hat{\pi}_1$ the proportions of 0 and 1 in the sample, respectively. Once the optimal question is found, the same procedure can be iteratively applied to E_{yes} and E_{no}, until the subsamples are perfectly pure.

According to the CART algorithm, we may observe that two variables X^j and X^k that induce the same ranking of the observations on the training sample will lead to equivalent questions. More precisely, any question formulated with X^j can be translated into a question formulated with X^k. This holds whatever the purity criterion, assuming that purity only depends on $\hat{\pi}_0$ and $\hat{\pi}_1$. Hence, we can define the redundancy between two variables for CART as given below.

Definition 4. *Two variables are redundant for the CART algorithm if they induce the same ranking on observations.*

From this definition, we can define the loss function for CART as a loss of information about ranks. We introduce the rank variables $R^1, \ldots, R^p$ associated to the initial variables $X^1, \ldots, X^p$ and measure the conservation of rank information via inertia:

$$I(R^1, \ldots, R^p) = \frac{1}{2n} \sum_{i=1}^{n} \sum_{j=1}^{n} \|R_i - R_j\|^2.$$

Inertia is commonly used to measure rank conservation (Kendall coefficient) or rank correlation (Spearman correlation). Using this criterion, we can define the loss function for CART:

Definition 5. *The CART loss associated to aggregation A is quantified by the decrease in inertia on associated rank variables due to this aggregation:*

$$\begin{aligned} &L_{CART}(\{X^1, \ldots, X^p\}, \quad \{Z^{\mathcal{C}_1}, \ldots, Z^{\mathcal{C}_{N_C}}\}) \\ &\quad = I(R^1, \ldots, R^p) - I(Z^{\mathcal{C}_1}, \ldots, Z^{\mathcal{C}_{N_C}}). \end{aligned}$$

From this, we obtain the characterization of the optimal aggregation for CART:

Proposition 3. *The optimal aggregation A^* of size N_C for the CART algorithm satisfies*

$$A^* = \underset{A \in \mathcal{A}}{\text{Argmin}} \sum_{\ell=1}^{N_C} \sum_{X^{\ell_1}, \ldots, X^{\ell_q} \in \mathcal{C}_\ell} L_{CART}(\{X^{\ell_1}, \ldots, X^{\ell_q}\},\ Z^{\mathcal{C}_\ell}),$$

where $Z^{\mathcal{C}_\ell}$ is the first component of the PCA on variables $R^{\ell_1}, \ldots, R^{\ell_q}$.

2.4 Optimization Algorithm

The resolution of the optimization problem given in Proposition 1 requires an exhaustive search among all the possible clusterings of the p variables into N_C groups. In practice, the computational burden associated with this search is too high, and one needs a heuristic method to explore a reduced (but well chosen) number of clusterings. We propose to use the hierarchical clustering algorithm (HCA, Anderberg [2]). HCA is an iterative procedure: it starts with p clusters containing one of the p initial variables, and at each step, the two closest clusters are joined. The procedure stops when the number of clusters is N_C.

To run the algorithm, a distance between clusters must be defined. If the aggregation loss is additive (as for the kNN and the CART algorithms), we can directly deduce the distance from the optimization program:

Definition 6. *Let $\mathcal{C}_\ell$ and $\mathcal{C}_{\ell'}$ be two clusters containing variables $X^{\ell_1}, \ldots, X^{\ell_q}$ and $X^{\ell'_1}, \ldots, X^{\ell'_{q'}}$, respectively. The distance between clusters $\mathcal{C}_\ell$ and $\mathcal{C}_{\ell'}$ is $D(\mathcal{C}_\ell,\ \mathcal{C}_{\ell'})$*

$$\begin{aligned} &= L_{alg}(\{X^{\ell_1}, \ldots, X^{\ell_q}, X^{\ell'_1}, \ldots, X^{\ell'_q}\}, Z^{\mathcal{C}_{\ell\ell'}}) \\ &\quad - L_{alg}(\{X^{\ell_1}, \ldots, X^{\ell_q}\}, Z^{\mathcal{C}_\ell}) \\ &\quad - L_{alg}(\{X^{\ell'_1}, \ldots, X^{\ell'_{q'}}\}, Z^{\mathcal{C}_{\ell'}}), \end{aligned} \tag{1}$$

where $\mathcal{C}_{\ell\ell'}$ is the cluster containing all the variables of clusters $\mathcal{C}_\ell$ and $\mathcal{C}_{\ell'}$.

The previous distance for clusters of variables may be understood as an equivalent of the popular Ward distance for clusters of observations: At each step, we look for the fusion of two clusters that achieves the minimum loss of inertia.

In many applications, there exists an ordering of the variables. In the speech recognition problem for instance (see Section 3.1), the goal is to distinguish between two words on the basis of the Fourier decomposition of the vocal signal. In this case, the variables are the different frequencies of the spectrum, and we can assume that there exist ranges of adjacent frequencies that hold common information. In this case, we want the aggregation to take into account the ordering of the variables. To deal with the case of ordered variables, we propose a Constrained version of the Hierarchical Clustering Algorithm (CHCA), where only adjacent variables (or clusters) may be aggregated. Interestingly, the complexity of the CHCA is drastically reduced ($\mathcal{O}(p)$) compared with the classical HCA algorithm ($\mathcal{O}(p^2)$), ensuring the CHCA to be very efficient to analyze large-dimension data. Note that dynamic programming would provide the exact optimal aggregation in the ordered case, but with the same computational cost ($\mathcal{O}(p^2)$) as the HCA algorithm.

2.5 Choice of the Number of Clusters

In the previous section, we assumed that the optimal number of variable clusters was known. However, in practice, this is a strong assumption and, in many cases, N_C is unknown and has to be chosen. Since we want to apply an aggregation-selection strategy, we also need to choose N_S, the number of variables (or variable clusters) to be selected. Since we want to use a maximum of information, these two parameters have to be tuned according to the data.

The choice of N_S is a classical problem and many authors have proposed to choose N_S according to the prediction performance of the classifier. To do this, different values of N_S are proposed and, for each of these values, the prediction performance of the

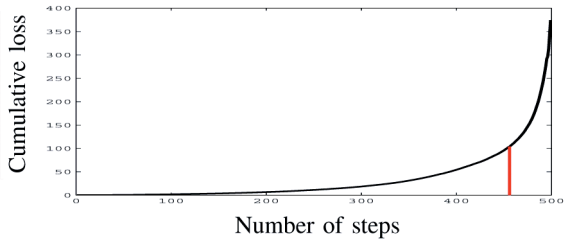

Fig. 1. Aggregation curve on the speech recognition data. The number of steps is given in abscissaes. At step s, there are $500 - s$ clusters of variables. The cumulative loss at each step is given in ordinates. The thick vertical lines show the breakpoint s^*, and the corresponding value of $N_{min} = 500 - s^*$ is 47.

classifier is evaluated by cross-validation/hold-on (Hastie et al. [19]) or upper bounded with a penalized criterion (Mary-Huard et al. [26]). The optimal value of N_S is the one that optimizes the estimated prediction performance.

While this strategy is relevant to tune N_S, since the main goal of the selection step is to improve the classification performance, it may not be as relevant to use this same strategy for the choice of N_C. Indeed, we have already argued that the goal of the aggregation step is to improve the interpretability of the results rather than the performance. The choice of N_C should reflect this objective. We propose to use the aggregation curve to fix a minimal value N_{min} of aggregation. The aggregation curve, as presented in Fig. 1, represents the cumulative loss at each step of the aggregation process. If there is a clear underlying structure of the variables, then it should be reflected in the aggregation curve: the first steps should be relevant (similar variables are clustered, with small associated losses), while the last ones should be irrelevant (groups of nonsimilar variables are clustered, with high associated losses). The path from relevant to irrelevant steps should be marked by an identifiable breakpoint on the aggregation curve (see Fig. 1): We assume that, after a given step s^*, the loss increases significantly so that we should not keep aggregating and we note $N_{min} = p - s^*$, the corresponding number of clusters. The step at which we should stop can be identified in the aggregation curve using a breakpoint detection strategy (Lavielle [24], Lavielle [25], and Birgé and Massart [5]). Here, we use the breakpoint detection strategy described in Lavielle [24] to determine s^* that is defined as the greatest value of s such that the second derivative of the aggregation curve is greater than a given threshold. Details can be found in the paper of Lavielle.

Since we want the results of the aggregation procedure to be interpretable, we will choose N_C by hold-on using a validation data set (as is done for N_S), but only among values that are higher than N_{min}.

3 Applications

In this section, we present the application of the aggregation strategy on four sets of real data. We compared different classifiers:

- a classifier built with the selection strategy (S),
- a classifier built with the aggregation/selection strategy (HCA+S),
- a classifier built with the ordered aggregation/selection strategy (CHCA+S).

Here, a classifier is defined by the choice of a classification algorithm (for instance, CART), the number of clusters N_C after the aggregation step that can vary from p (no aggregation) to 1 (all variables clustered into a unique group), and the number N_S of selected clusters (or variables). In the following examples, when the kNN algorithm is applied, we performed forward variable selection.

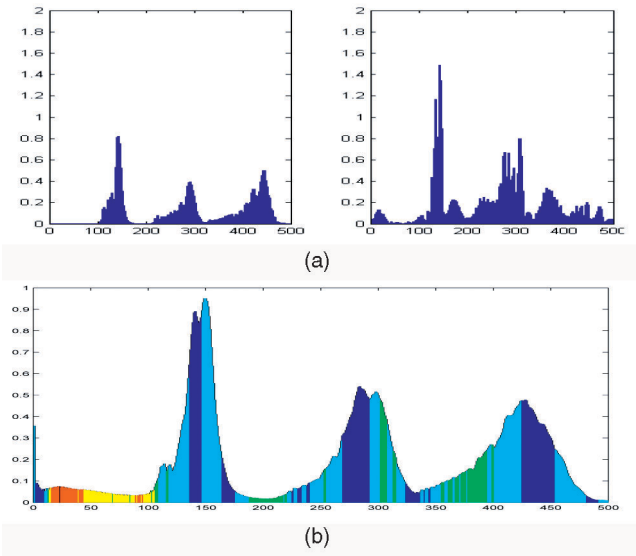

Fig. 2. (a) Examples of spectra for words *boat* (left) and *goat* (right). Picture obtained from Biau et al. [4]. (b) Average spectral density over the 100 signals. Colors represent the level of the T-test statistic of the comparison of means between the two populations.

To evaluate the different strategies, we used the following sampling of the data: For a given data set with n observations, we note n_ℓ, n_v, and n_t the sizes of the training, validation, and test sets, respectively, with $n_\ell + n_v + n_t = n$. In the following, we adopt the notation $[n_\ell, n_v, n_t]$ to designate this evaluation sampling strategy. For the speech recognition data set, for instance, n_t is fixed at 1. In this case, we performed n samplings of the data set. For sampling i, the ith observation was removed, and the remaining observations were randomly split into n_ℓ training data and n_v validation data. The training set is used to estimate the parameters of the classifier (for instance, in CART to select variables and thresholds), N_C and N_S being fixed. Then, the n_v validation data are used to compare all the classifiers and select the optimal combination $\{N_C^*, N_S^*\}$. Once these two parameters are chosen, all $n_\ell + n_v$ observations are used to build an optimal classifier, whose performance is estimated on the test set (i.e., by leave-one-out since $n_t = 1$).

3.1 Speech Recognition: First Example

We consider the speech recognition data set presented in Biau et al. [4], where 55 recordings of the word "Boat" and 45 of the word "Goat" were collected. The data arise from the discretization of the 100 analog signals and consist of time series of length 8,192. Each data go through a Fourier transform process on a $P = 500$ period basis:

$$f(t) = \sum_{j=1}^{P} A_j e^{\frac{2\pi j}{P} t},$$

so that the processed data are of the form (X, Y), where $X^j = |A_j|$ is the energy associated with angular period $\frac{2\pi j}{p}$ and $Y = 1$ if the signal corresponds to the word "Boat." Fig. 2 shows the average spectral density over the 100 signals. The 500 angular periods are represented in abscissaes and the ordinates correspond to the average energy associated with each period. Colors represent the level of the T-test statistic of the comparison of mean between the two populations (Boat and Goat). It clearly appears that there exist ranges of periods that share common information, and the relevant information to classify is roughly held by periods between 20 and 100.

We performed the classification using the kNN algorithm, with parameter k fixed at 3 (the analysis performed with $k = 5$ gives

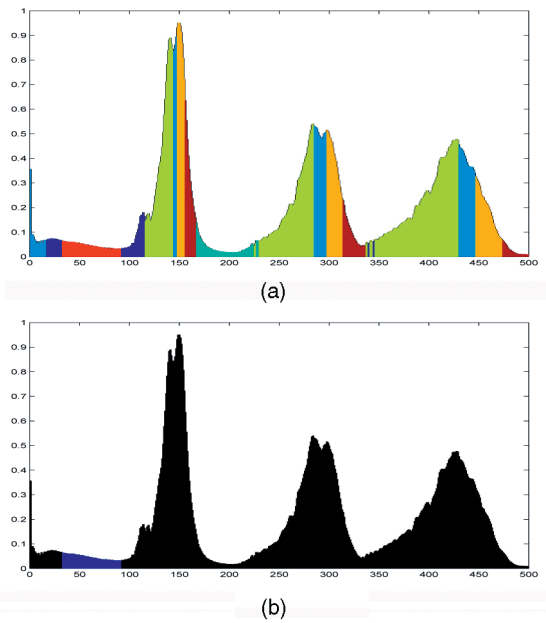

Fig. 3. (a) Periods with the same color are clustered together. While the ordering of the data is not taken into account, periods are clustered into large redundant regions. (b) During the selection step, only one region is selected to build the final classifier. The selected region is the one identified during the descriptive analysis in Fig. 2.

TABLE 1
Classification Error Rate for the Three Strategies

Method	N_C^*	N_S^*	Error Rate
kNN (S)	-	3.7 (2.5)	7%
kNN (HCA+S)	19.5 (23.5)	5.2 (4)	0%
kNN (CHCA+S)	38.8 (29.5)	4.7 (2.7)	0%
Cart (S)	-	1.4 (0.5)	3%
Cart (CHCA+S)	21.9 (9.2)	1.7 (0.4)	2%

Column N_C^* (respectively, N_S^*) indicates the average number of clusters after the aggregation step (respectively, number of selected variables/clusters) over the 100 samplings. Standard deviations are given in parentheses.

similar results) and using a $[50, 49, 1]$ evaluation sampling as in Biau et al. [4]. Fig. 3 illustrates the process of the (HCA+S) strategy. First, the aggregation step creates clusters of periods. Notice that while the ordering is not taken into account here, the aggregation algorithm clusters periods into ordered regions (left). This demonstrates that the nonordered aggregation is able to find the variable ordering when it exists. Noncontiguous are then clustered, showing that the nonordered aggregation also captures long-range correlations. In the selection step, a single region is selected, corresponding to the 20-100 range that was observed in Fig. 2.

We now consider the classification performance of the different strategies. As a comparison, Biau et al. [4] achieved a classification error rate of 21 percent by applying the kNN algorithm to the first d principal components of the Fourier transform, where d is chosen with a cross-validation strategy. Tuleau [32] improved the parameter selection strategy proposed by Biau and lowered the classification error rate to 15 percent. Last, Rossi and Villa [29] used an adapted SVM algorithm for functional data and obtained an error rate of 8 percent. Both Biau and Rossi used the same sampling strategy presented here. Table 1 shows the results obtained with the (S), (HCA+S), and (CHCA+S) strategies. In the case of the speech recognition data set, the aggregation strategy contributes to improve the performance of the classifier.

Fig. 4 shows the frequencies of selection for the 500 periods, over the 100 samplings, for strategies (S), (HCA+S), and (CHCA+S), respectively. The selection strategy does not allow

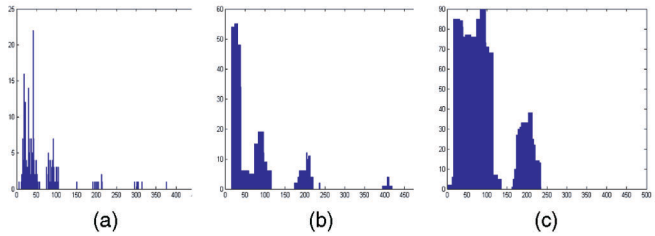

Fig. 4. Selection frequency of the 500 periods with strategies (a) S, (b) HCA+S, and (c) CHCA+S.

TABLE 2
Classification Error Rate for the Different Strategies on the Phoneme Data Set, over 100 Samplings

Method	N_C^*	N_S^*	Error Rate
Funct. Lin. SVM ([Rossi and Villa (2006)])			19.4%
Cart (S)	–	5.15 (3.3)	24.1% (1.2)
Cart (CHCA+S)	47.6 (27.2)	3.02 (2.5)	22.1% (1.2)
kNN (CHCA+S)	61.5 (22.4)	6.19 (2.5)	24.8% (1.6)

Standard deviations are given in parentheses.

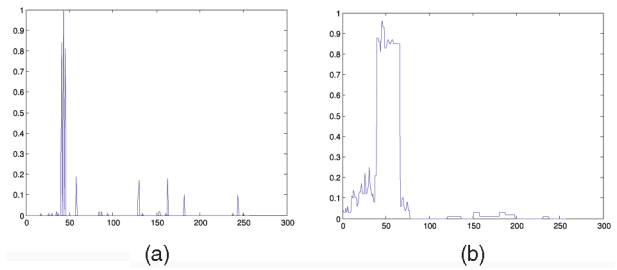

Fig. 5. Selection frequency of the 256 variables with the (a) (S) strategy and (b) (CHCA+S) strategy.

the identification of a clear region that holds the information about the status, while the (HCA+S) and (CHCA+S) strategies do. With Cart, the (S) and (CHCA+S) strategies give similar results (Table 1) and the same region is found with the aggregation strategy, and not with selection only (results not shown). These results demonstrate that the aggregation strategy results in a gain of insight about the data.

3.2 Speech Recognition: Second Example

We consider the TIMIT data set (Hastie et al. [18]) that consists of 4,509 speech frames. Each frame is recorded at a 16 kHz sampling rate and we retain only the first 256 frequencies. Following Rossi and Villa [29], we only consider the 695 "aa" and the 1,022 "ao" speech frames, because these two frame classes are the most difficult to distinguish. Thus, the data set consists of 1,717 series of length 256 with class membership 0 ("aa") or 1 ("ao"). Furthermore, we used the same splitting into 439 test data and 1,278 train/validation data that were used in previous publication, and we considered 100 different splits of the 1,278 train/validation data set into 638 train and 638 validation data.

Table 2 summarizes the results. Again, aggregation improves the classification performance of the classifier. Fig. 5 shows the selection frequency of the 256 variables. There is a clear zone that is highly informative for the classification goal, which is found by the two (CART and kNN) aggregation algorithms. This zone contains 31 frequencies (from frequency 33 to 63). On average, 22.5 of these 31 frequencies are selected with the (CHCA+S) procedure, whereas only three of them are selected with the (S) procedure, so that, on a single run with (S), one can fail to detect that the entire 33-63 zone is relevant for classification.

TABLE 3
Classification Error Rate for Different Strategies

Method	N_C^*	N_S^*	Error Rate
PLS+LDA ([Preda *et al.* (2007)])	–	–	11.2%
Cart (S)	–	1.4 (0.5)	8.5% (5.1)
Cart (CHCA+S)	36.6 (27.0)	1.4 (0.49)	8.6 (4.3)
kNN (S)	–	2.0 (2.0)	6.2% (3.3)
kNN (CHCA+S)	22 (7.0)	2.5 (2.5)	6.6% (3.5)

Column N_C^* (respectively, N_S^*) indicates the average number of clusters after the aggregation step (respectively, number of selected variables/clusters) over the 100 samplings. Standard deviations are given in parentheses.

3.3 Kneading Data

We consider the kneading data set used in Costanzo et al. [8] and Preda et al. [28], where the goal is to predict the quality of cookies according to kneading measurements. The data set consists of 90 kneading curves, each representing the density of dough observed during the kneading process. For a given cookie, we have 241 measurements (one measurement every 2 seconds, between 0 and 480 seconds), and a binary label that scores 1 if the quality is good, 0 otherwise. Details about the preprocessing of the data can be found in Costanzo et al. [8]. We performed 100 samplings using a $[40, 20, 30]$ evaluation sampling. Since the data are ordered, we used the CHCA algorithm for the aggregation step.

The performance of (S) and (CHCA+S) for kNN and Cart along with the performance related in Preda et al. [28] are given in Table 3. Results are comparable for the different methods, and in this example, selection improves the classification performance, while aggregation does not.

While the classification performance is important here, we also would like to detect as soon as possible whether the cookie quality will be good or not. We expect variables to be more and more informative about the quality, since it should be easier to predict at the end of the process than in the beginning. Fig. 6 shows, for each measurement (each time), the number of times it was used to predict the label for the Cart analysis.

With only a selection step, a single variable (t_{344}) seems to hold most of the information about the cookie quality. If an aggregation step is combined with the selection step, there is a range of variables (from t_{344} to t_{380}) sharing the information, variable t_{344} being the first of the range (Fig. 6b). This means that either a unique variable holds the information, and the aggregation step corrupts the interpretation by clustering variable t_{344} with other variables, or there is a cluster of equivalent variables, and the first one is selected most of the time during the selection step because it appears first on the data set. To investigate this point, we permuted the order of variables t_{344} and t_{352} and applied strategy (S) to the modified data set. The most selected variable is now variable t_{352} (Fig. 6c). This clearly shows that even in simple applications, the selection step can lead to a misleading interpretation of the data. With the aggregation step, we can conclude that the more relevant information about the quality is contained in the measurements after t_{150}, where there are two ranges of time with useful information for classification.

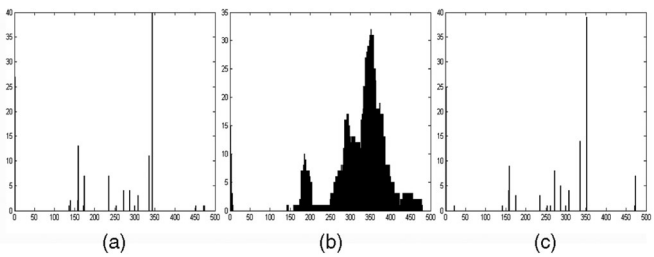

Fig. 6. Number of selections per variable, for the (a) (S) strategy and (b) (CHCA+S) strategy. (c) The result of the (S) strategy after permutation of variables t_{344} and t_{352}.

TABLE 4
Classification Error Rate and Selection Frequency of Genes CD33 Antigen and Macmarcks as a Function of the Aggregation Level, for the Golub Data Set

N_C	N_S	Error rate	CD33 Antigen	Macmarcks
3571 (No Aggreg)	3	10.4%	10 %	19 %
600	3	7.4%	50 %	58 %
100	3	7.3%	62 %	65 %

3.4 Microarray Data

The Golub data set consists of 78 individuals, described by 3,571 genes, and labeled according to their type of leucemia (AML or ALL). A complete description of the data can be found elsewhere (Ben-Dor et al. [3]). We first analyzed the Golub data set using a $[38, 34]$ evaluation sampling (38 for training, 34 for test, no validation since S is fixed). We fixed the number of selected variables at $N_S = 3$ and considered the performance obtained for different levels of aggregation N_C. The results are provided in Table 4. One can see that the classification error rate improves when aggregation is performed. Furthermore, the selection frequency of genes CD33 Antigen and Macmarcks, which are known to be both discriminative and redundant in this classification problem, increases with the aggregation level.

We then analyzed the Golub data using a $[45, 26, 1]$ evaluation sampling. Results are provided in Table 5, along with some results reported in previous publications.

In this microarray context, no obvious prior structure in the data, such as an order on the variables, exists. Selected clusters can only be interpreted in the light of literature. We compared the list of the variables selected with the kNN algorithm and either the (S) strategy or the (HCA+S) strategy, with the list of 100 variables described in Su et al. [31]. The Su list contains variables that were reported in several publications. With the (HCA+S) strategy, there were 40 genes that were selected more than 60 times on 100 samplings. Most of these genes (27) belong to the Su list. As a comparison, with the (S) strategy, only nine genes are selected at least 10 times and only four of them belong to the Su list.

4 Discussion

Little work has been done to exploit the great amount of information that can be extracted from sets of redundant variables in the classification framework. In many articles, redundancy has been described as a problem that may alter the performance of the classification algorithm, rather than a strength. In fact, some authors have argued that there may be a trade-off between the ease of interpretation and the performance of a classifier. In [6], for instance,

TABLE 5
Classification Error Rates for Different Algorithms, for the Golub Data Set

Method	N_C^*	N_S^*	Error Rate
kNN	–	–	2.8%
kNN (S)	–	8.6	7%
kNN (S+HCA)	22.5	3.8	2.8%
Adaboost ([Ben-Dor *et al.* (2000)])	–	–	4.2%
SVM (quadratic kernel, [Ben-Dor *et al.* (2000)])	–	–	4.2%
Logistic regression ([Krishnapuram *et al.* (2004b)])	–	–	2.8%
JCFO (linear kernel) ([Krishnapuram *et al.* (2004b)])	–	–	0%

the author compares CART with the random forest algorithm. On one hand, the CART algorithm is an easy-to-interpret classification rule with limited prediction performances: *While trees rate an A+ on interpretability, they are good, but not great, predictors. Give them, say, a B on prediction.* On the other hand, random forests significantly improve the classification performance of CART, but at the price of a increase in complexity of the classification rule: *So forests are A+ predictors. But their mechanism for producing a prediction is difficult to understand. Trying to delve into the tangled Web that generated a plurality vote from 100 trees is a Herculean task. So, on interpretability, they rate an F.* This exemplifies the trade-off: An improvement in the performance results in a decrease in interpretability.

The strategy proposed by Breiman and the one developed here are both based on aggregation. The difference lies in the fact that Breiman proposes to aggregate models and not variables. Model aggregation is an efficient way to deal with the problem of multiplicity of good models, where several different models lead to the same classification performance. While Breiman makes no assumption about the origin of the problem of multiplicity of good models (the word "redundancy" does not appear in his paper), we assume that, in many cases, the multiplicity problem comes from redundancy. If this assumption is satisfied, then variable aggregation may be an attractive alternative to model aggregation, and both the prediction performance and the interpretability of the classification rule may be improved. The results presented here prove that there is no systematic conflict between the two goals.

In this paper, we proposed a generic framework for the development of tailored aggregation methods. This strategy has been successfully applied to the CART and kNN algorithms. While we illustrated the performance of the aggregation strategy on classification problems, both CART and kNN are also used in the regression framework. The aggregation method can be straightforwardly extended to this framework: Similarly to the classification context, the regression kNN algorithm breaks down into a tessellation step and a prediction step. The tessellation remains the same as in Section 2.2, and conditionally to the tessellation, the prediction step does not take into account variables $X^1, \ldots, X^p$. Hence, the aggregation developed for classification may be directly applied to the regression case. The same reasoning also holds for CART.

The aggregation algorithm may also be applied to other classification algorithms. According to Proposition 1, the aggregation method we propose requests the definition of the loss function L_{alg} that depends on algorithm alg. To define this loss may be a hard task, especially when considering "black box" classification algorithms, where the effect of a given variable on the classification rule is very difficult to quantify. Considering the two applications presented in this paper, algorithms where information held by the variables is summed up into an individual-to-individual table appear to be a favorable case to apply the tailored aggregation strategy. In the kNN example, the variable information only appears in the $n \times n$ table of distance between individuals, so the aggregation must alter this information as little as possible. Further applications of our strategy should include linear SVM, where the variable information also appears as contained in an individual-to-individual table (i.e., the Gram matrix), so that the framework presented here could be easily extended to this algorithm.

It is important to propose efficient algorithms to deal with large data sets. In many applications, new technologies give access to thousands (sometimes millions) of measurements in a single experiment. The constrained hierarchical aggregation we proposed in this paper takes advantage of the ordering of the variables and performs aggregation with a linear computational complexity cost. We applied the (CHCA+S) strategy using a personal computer and Matlab: Aggregation for data sets with 2,000 variables only takes a few seconds, meaning that the aggregation strategy proposed here is able to tackle very high-dimensional data sets with ordered variables, such as those provided by high-density microarray technologies.

ACKNOWLEDGMENTS

The authors want to thank the reviewers for their rich and constructive comments that helped to improve the manuscript.

REFERENCES

[1] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine, "Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays," *Proc. Nat'l Academy of Sciences USA,* vol. 96, no. 12, pp. 6745-6750, 1999.
[2] M. Anderberg, *Cluster Analysis for Applications.* Academic Press, Inc., 1973.
[3] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, "Tissue Classification with Gene Expression Profiles," *J. Computational Biology,* vol. 7, pp. 559-583, 2000.
[4] G. Biau, F. Bunea, and M. Wegkamp, "Functional Classification in Hilbert Spaces," *IEEE Trans. Information Theory,* vol. 51, no. 6, pp. 2163-2172, June 2005.
[5] L. Birgé and P. Massart, "Minimal Penalties for Gaussian Model Selection," *Probability Theory and Related Fields,* vol. 138, pp. 33-73, 2007.
[6] L. Breiman, "Statistical Modeling: The Two Cultures," *Statistical Science,* vol. 16, no. 3, pp. 199-231, 2001.
[7] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees.* Wadsworth Int'l, 1984.
[8] D. Costanzo, C. Preda, and G. Saporta, "Anticipated Prediction in Discriminant Analysis on Functional Data for Binary Response," *Proc. 17th Symp. Computational Statistics,* pp. 821-828, 2006.
[9] M. Dettling, "Revealing Predictive Gene Clusters with Supervised Algorithms," *Proc. Conf. in Distributed Statistical Computing,* 2003.
[10] M. Dettling and P. Bühlmann, "Supervised Clustering of Genes," *Genome Biology,* vol. 3, no. 12, pp. 1-15, 2002.
[11] R. Diaz-Uriarte, "Molecular Signatures from Gene Expression Data," to be published, 2004.
[12] S. Dudoit, J. Fridlyand, and T. Speed, "Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data," *J. Am. Statistical Assoc.,* vol. 97, pp. 77-87, 2002.
[13] E. Fix and J. Hodges, "Discriminatory Analysis—Nonparametric Discrimination: Consistency Principles," *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques,* B.V. Dasarath, ed., IEEE CS Press, 1991.
[14] E. Fix and J. Hodges, "Nonparametric Discrimination: Small Sample Performance," *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques,* B.V. Dasarath, ed., IEEE CS Press, 1991.
[15] K. Fukumizu, F. Bach, and M. Jordan, "Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces," *J. Machine Learning Research,* vol. 5, pp. 73-99, 2004.
[16] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *J. Machine Learning Research,* vol. 3, pp. 1157-1182, 2003.
[17] H. Harman, *Modern Factor Analysis.* Univ. of Chicago Press, 1973.
[18] T. Hastie, A. Buja, and R. Tibshirani, "Penalized Discriminant Analysis," *Annals of Statistics,* vol. 23, pp. 73-102, 1995.
[19] T. Hastie, R. Tibshirani, M. Eisen, A. Alizadeh, R. Levy, L. Staudt, W.C. Chan, D. Botstein, and P. Brown, "'Gene Shaving' as a Method for Identifying Distinct Sets of Genes with Similar Expression Patterns," *Genome Biology,* vol. 1, no. 2, 2000.
[20] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2001.
[21] R. Kohavi and G. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence,* vol. 97, pp. 273-324, 1997.
[22] B. Krishnapuram, A. Hartemink, L. Carin, and M. Figueiredo, "A Bayesian Approach to Joint Feature Selection and Classifier Design," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1105-1111, Sept. 2004.
[23] B. Krishnapuram, L. Carin, and A. Hartemink, "Gene Expression Analysis: Joint Feature Selection and Classifier Design," *Kernel Methods in Computational Biology,* MIT Press, 2004.
[24] M. Lavielle, "Detection of Multiple Changes in a Sequence of Dependent Variables," *Stochastic Processes and Their Applications,* vol. 83, pp. 79-102, 1999.
[25] M. Lavielle, "Using Penalised Contrasts for the Change-Point Problem," *Signal Process,* vol. 85, no. 8, pp. 1501-1510, 2005.
[26] T. Mary-Huard, S. Robin, and J.-J. Daudin, "A Penalized Criterion for Variable Selection in Classification," *J. Multiple Analysis,* vol. 98, no. 4, pp. 695-705, 2007.
[27] S. Michiels, S. Koscielny, and C. Hill, "Prediction of Cancer Outcome with Microarrays: A Multiple Random Validation Strategy," *Lancet,* vol. 365, pp. 488-492, 2005.
[28] C. Preda, G. Saporta, and C. Lévéder, "PLS Classification of Functional Data," *Computational Statistics,* vol. 22, no. 2, pp. 223-235, 2007.

[29] F. Rossi and N. Villa, "Support Vector Machine for Functional Data Classification," *Neural Computing*, vol. 69, nos. 7-9, pp. 223-239, 2006.
[30] G. Saporta, *Probabilités, Analyse des Données et Statistique.* Editions Technip, 1990.
[31] Y. Su, T. Murali, V. Pavlovic, M. Schaffer, and S. Kasif, "Rankgene: Identification of Diagnostic Genes Based on Expression Data," *Bioinformatics*, vol. 19, no. 12, pp. 1578-1579, 2003.
[32] C. Tuleau, "Sélection de Variables Pour la Discrimination en Grande Dimension et Classification de Données Fonctionnelles," PhD thesis, Univ. Paris-Sud XI, 2005.
[33] M. Xiong, W. Li, J. Zhao, L. Jin, and E. Boerwinkle, "Feature (Gene) Selection in Gene Expression-Based Tumor Classification," *Molecular Genetics and Metabolism*, vol. 73, no. 3, pp. 239-247, 2001.
[34] L. Yu and H. Liu, "Efficient Feature Selection via Analysis of Relevance and Redundancy," *J. Machine Learning Research*, vol. 5, pp. 1205-1224, 2004.