Selection and Statistical Validation of Features and Prototypes

M. Sebban[1,2], D.A. Zighed[2] & S. Di Palma[2]

[1] : **R.A.P.I.D.** laboratory - West Indies and Guiana University, France.
Marc.Sebban@univ-ag.fr
[2] : **E.R.I.C.** laboratory - Lyon 2 University, France.
{zighed,sebban,sdipalma}@univ-lyon2.fr

Abstract. Features and protypes selection are two major problems in data mining, especially for machine learning algorithms. The goal of both selections is to reduce storage complexity, and thus computational costs, without sacrificing accuracy. In this article, we present two incremental algorithms using geometrical neighborhood graphs and a new statistical test to select, step by step, relevant features and prototypes for supervised learning problems. The feature selection procedure we present could be applied before any machine learning algorithm is used.

1 Introduction

We deal in this paper with learning from examples ω described by pairs $[X(\omega), Y(\omega)]$, where $X(\omega)$ is a vector of p feature values and $Y(\omega)$ is the corresponding class label. The goal of a learning algorithm is to build a *classification function* φ from a sample Ω_a of n examples $\omega_{j,(j=1...n)}$.

From a theoretical standpoint, the selection of a good **subset of features** X is of little interest : a Bayesian classifier (based on the true distributions) is monotonic, *i.e.*, adding features can not decrease the model's performance [10]. This task has however received plenty of attention from statisticians and reseachers in Machine Learning since the monotonicity assumption rarely holds in practical situations where the true distributions are unkown. Irrelevant or weakly relevant features may thus reduce the accuracy of the model. Thrun et al. [18] showed that the C4.5 algorithm generates deeper decision trees with lower performances when weakly relevant features are not deleted. Aha [1] also showed that the storage of the IB3 algorithm increases exponentially with the number of irrelevant features.

Selection of **relevant prototype subsets** has also been much studied in Machine Learning. This technique is of particular interest when using non parametric classification methods such as *k-nearest-neighbors* [8], Parzen's windows [12] or more generally methods based on geometrical models that have a reputation for having high computational and storage costs. In fact, the classification of a new example often requires distance calculations with all points stored in

J.M. Żytkow and J. Rauch (Eds.): PKDD'99, LNAI 1704, pp. 184–192, 1999.

memory. This led researchers to build strategies to reduce the size of the learning sample (selecting only the "best" examples which will be called *prototypes*), keeping and perhaps increasing classification performances [8], [7] and [17].

We present in this article two *hill climbing* algorithms to select relevant features and prototypes, using models from computational geometry. The first algorithm step by step selects relevant features independently of a given learning algorithm (the classification accuracy is not used to identify the "best" features but only to stop the selection algorithm). This feature selection technique is based on the idea that performances of a learning algorithm, whatever the algorithm may be, necessarily depend on the geometrical structures of classes to learn. We propose characterizing these structures in $I\!R^p$ using models inspired from computational geometry. At each stage, we statistically measure the separability of these structures in the current representation space, and verify if the kept features allow to build a model more efficient than the previous one.

Unlike the first, the second algorithm uses the classification function to select prototypes in the learning sample. It tests the "quality" of selected examples, verifying on the one hand that they allow to obtain on a validation sample a success rate significantly close to the one obtained with the full sample, and on the other hand that they constitute one of the best learning subsets with this size.

2 Definitions in Computational Geometry

The approach we propose in this article uses neighborhood graphs. Interested readers will find many models of neighborhood graphs in [13], such as Delaunay's Triangulation, Relative Neighborhood Graph, and the Minimum Spanning Tree (Fig. 1).

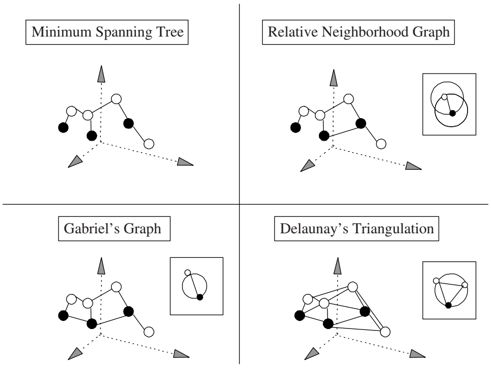

Fig. 1. Neighborhood Structures

Definition 1. : *A graph G (Σ,A) is composed of a set of vertices noted Σ linked by a set of edges noted A.*

NB : In the case of an oriented graph, A will be the set of arcs. In our paper, we only consider non-oriented graphs, i.e. a link between two points defines an edge. This choice makes every neighborhood relation symmetrical.

3 Selection of Relevant Features

3.1 Introduction

Given a representation space $\mathbf{X}$ constituted by p features $X_1, X_2, ..., X_p$, and a sample of n examples noted $\omega_1, \omega_2, .., \omega_n$. ,a learning method allows to build a classification function φ_1 to predict the state of Y.

Consider now a subset $\mathbf{X}' = \{X_1, X_2, ..., X_{p'}\}$ of all features, with $p^{'} < p$, and note φ_2 the classification function built in this new representation space. If classification performances of φ_1 and φ_2 are equivalent, we will always prefer the model using fewer features for the construction of φ . Two reasons justify this choice:

1. The choice of $\mathbf{X}'$ reduces overfitting risks.
2. The choice of $\mathbf{X}'$ reduces computational and storage costs.

Generalization performances of φ_2 may sometimes be better than those obtained with φ_1, because some features can be "noised" in the original space. Nevertheless, we can not test all combinations of features, *i.e.* build and test $2^p - 1$ classification functions.

Constructive methods (decision trees, fuzzy trees, induction graphs, etc.) select features step by step when they improve performances of a given criterion (classification success rate, homogeneity criterion). In these methods, the construction of the φ function is done simultaneously with features choice. Among works using the estimation of the classification success rate, we can cite the cross-validation procedure [10], and Bootstrap procedure [5]. Nevertheless, even if these methods allow to obtain an unbiased estimation of this rate, calculation costs seem prohibitive to justify these procedures at each stage of the feature selection process.

Methods using homogeneity criterion often propose simple indicators fast to compute, such as entropy measures, uncertainty measures, separability measures like the Λ of Wilks [14] or Mahalanobis's distance. But results also depend on the current φ function.

We propose in the next section a new features selection approach, applied before the construction of the φ classification function, independently of the learning method used. To estimate quality of a feature, we propose to estimate quality of the representation space with this feature.

3.2 How to Evaluate the Quality of a Representation Space?

We consider that m different classes are well represented by p features, if the representation space (characterized by p dimensions) shows wide geometrical structures of points belonging to these classes. In fact, when we build a model, we always search for the representation space farthest from the situation where each point of each class constitutes one structure. Thus, the quality of a representation space can be estimated by the distance to the worst situation characterised by the *equality of density functions of classes*. To solve this problem, we can use one of the numerous statistical tests of population homogeneity. Unfortunately, none of these tests is both nonparametric and applicable in $I\!R^p$. In Sebban [15], we built a new statistical test (called *test of edges*), which does not suffer from these constraints. Under the null hypothesis H_0 :

$$H_0 : F_1(x) = F_2(x) = ... = F_m(x) = F(x)$$

where $F_i(x)$ corresponds to the repartition function of the class i

The construction of this test uses some contributions of computational geometry. Our approach is based on the search for geometrical structures, called *homogeneous subsets,* joining points that belong to the same class. To obtain these *homogeneous subsets* and evaluate the quality of the representation space, we propose the following procedure :

```
1. Construct a related geometrical graph, such as the Delaunay
   Triangulation, the Gabriel's Graph, etc. [13].
2. Construct homogeneous subsets, deleting edges connecting points
   which belong to different classes.
3. Compare the proportion of deleted edges with the probability
   obtained under the null hypothesis.
```

The critical threshold of this test is used to search for the representation space which is the farthest from the H_0 hypothesis. Actually, the smaller this risk is, the further from the H_0 hypothesis we are. Two stategies are possible to find a "good" representation space :

1. Search for the representation space which *minimizes the critical threshold* of the test, *i.e.* which is the farthest from the H_0 hypothesis. Later on, we will use this approach to tackle this problem.
2. Search for a way to *minimize the size* of the representation space (with the advantage of reducing storage and computing costs), without reducing the quality of the initial space.

3.3 Algorithm

Let $\mathbf{X} = \{X_1, X_2, ..., X_p\}$ be the representation of a given Ω_a learning sample. Among these p features, we search for the p^* most discriminant ones ($p^* < p$) using the following algorithm:

```
1. Compute the $\alpha_0$ critical threshold of the test of edges in the
   initial representation space $\mathbf{X}$
2. Compute for each combination of $p-1$ features taken among the $p$
   current, the $\alpha_c$ critical threshold
3. Select the feature which minimizes the $\alpha_c^*$ critical threshold
4. If $\alpha_c^* < \alpha_0$ then delete the selected feature, $p \leftarrow p-1$, return
   to step 1[1] else $p^* = p$ and stop.
```

This algorithm is a *hill climbing* method. It does not search for an optimal classification function, in accordance with a criterion based on an uncertainty measure, but rather aims at finding a representation space that allows to build a better model.

3.4 Simulated Example

To illustrate our approach, we apply in this section our algorithm to a simulated example.

Let Ω_a be a learning sample composed of 100 examples belonging to two classes. Each example is represented in $I\!R^3$ by 3 features (noted $X_{1,}$ $X_{2,}$ X_3). The two classes are statistically different, *i.e.* characterised by two different probability densities. For instance,

∗ Normal law $N(\mu_1, \sigma_1)$ for examples of y_1 class
∗ Normal law $N(\mu_2, \sigma_2)$, where $\mu_2 > \mu_1$ for examples of y_2 class

To estimate the capacity of our algorithm to find the best representation space, we generate 3 new noised features (noted $X_{4,}$ $X_{5,}$ X_6). Each feature is generated identically for the whole sample. The first α_0 risk in $I\!R^6$ is about 1.10^{-8}. Applying our algorithm, we obtain the following results (table 1).

Table 1. Application of the feature selection algorithm

step i	X_1	X_2	X_3	X_4	X_5	X_6	α_c^*	α_0	Decision
1	5.10^{-5}	5.10^{-5}	2.10^{-8}	2.10^{-13}	3.10^{-12}	8.10^{-12}	2.10^{-13}	1.10^{-8}	Continue
2	7.10^{-6}	3.10^{-6}	1.10^{-12}	*	1.10^{-13}	4.10^{-14}	4.10^{-14}	2.10^{-13}	Continue
3	4.10^{-6}	4.10^{-4}	6.10^{-12}	*	1.10^{-15}	*	1.10^{-15}	4.10^{-14}	Continue
4	3.10^{-5}	1.10^{-4}	9.10^{-14}	*	*	*	9.10^{-14}	1.10^{-15}	Stop

During step 1, deletion of X_4 feature allows to reduce critical threshold (from 1.10^{-8} to 2.10^{-13}). Steps 2 and 3 lead to the supression of X_6 and X_5. At the fourth step, the value 9.10^{-14} (without the X_3 feature) does not allow to reduce the α_0 risk and thus the process stops. Unlike numerous other methods that would also select (X_1, X_2,X_3), our approach does not use the φ classification function.

[1] If we search for minimizing the size of the space, we will return to **2**.

4 Prototype Selection

4.1 Presentation

Intuitively, we think that a small number of prototypes can lead to comparable and perhaps higher performances than those obtained with a whole sample. We justify this idea as follows :

1. Some noise or repetitions in data could be deleted,
2. Each prototype can be viewed as a supplementary degree of freedom. Reducing the number of prototypes can thus sometimes avoid overfitting situations.

To reduce storage costs, some approaches use an algorithm selecting misclassified examples such as *condensed nearest neighbors* [8] which allows to find a *consistent subset*, *i.e.* which correctly classifies all the remaining points in the sample set. In [7], the author proposes the *reduced nearest neighbor rule* which improves the previous algorithm by finding the *minimal consistent subset* if it belongs to the Hart's consistent subset.

Skalak [17] proposes two different prototype selection algorithms : the first is a Monte Carlo sampling algorithm ; the second applies random mutation hill climbing, where the *fitness function* is the classification success rate on the learning sample. Yet, this approach is limited to simple problems where classes of patterns are easily separable, since the author *a priori* defines the number of prototypes as the number of classes. We can easily imagine some problems when classes are mixed. In our mind, we could improve this algorithm using as the number of prototypes the number of homogeneous subsets described in the previous section.

Other works about prototype selection can be found in [9] or [11].

In this section, we present a new decision rule, the *Probabilistic Vote*, that uses the information contained in a connected neighborhood graph. We then present the principle of its use in a variant of the prototype selection method proposed in [8] and [7].

4.2 The Probabilistic Vote

Our approach uses a weighted vote of neighbors (in a connected neighborhood graph) to label a new example. The weight of the a neighbor relationship is measured by the probability of the two examples being neighbors even if the size of the set increases.

We present in this section the theoretical frame of the Probabilistic Vote, with Gabriel's Graph but this approach can be extended to other neighborhood structures.

Definition 2. : *Weight* $\alpha(\omega_j, \omega)$

Let $\alpha(\omega_j, \omega)$, *the weight of the* ω_j *voter, neighbor of* ω*, be defined as :*

$$\begin{aligned}\alpha : \Omega_a * \Omega \rightarrow \quad & [0,1] \\ (\omega_j,\omega) \longmapsto & \alpha(\omega_j,\omega) = \Pr(\omega' \notin S_{\omega_j,\omega}),\ \forall \omega' \in \Omega\end{aligned}$$

where $S_{\omega_j,\omega}$ is the hypersphere with the diameter (ω_j,ω).

Definition 3. **:** *Covering space*

We define the covering space D containing all possible membership of the Ω set as the hypercube covering the union of hyperspheres of neighbors in the learning sample.

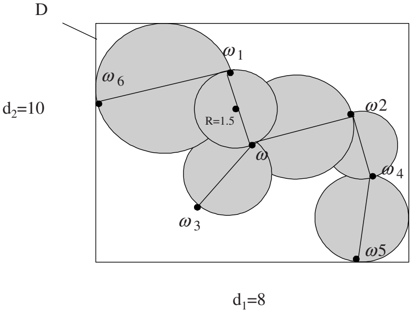

Fig. 2. Example of covering space.

From D, we calculate the probability

$$Pr(\omega' \notin S_{\omega_j,\omega}) = \frac{V_D - V_{S_{\omega_j,\omega}}}{V_D}$$

where V_D is the volume of D and $V_{S_{\omega_j,\omega}}$ is the volume of the hypersphere with diameter (ω_j,ω).

Definition 4. : *We define $V_{S_{\omega_j,\omega}}$, the volume of a given hypersphere in $I\!R^p$ with diameter (ω_j,ω)* as :

$$V_{S_{\omega_j,\omega}} = \frac{2}{p} r^p_{\omega_j,\omega} \frac{\sqrt{\pi^p}}{\Gamma(\frac{p}{2})}$$

where $r_{\omega_j,\omega}$ is the radius of the hypersphere with diameter (ω_j,ω) and $\Gamma(x)$ is the Gamma function.

V_D is obtained by multiplication of the lengths of the hypercube's sides.

Example :

Given a Gabriel's Graph built from a learning sample $\Omega_a = \{\omega_1,\omega_2,\omega_3,\omega_4,\omega_5,\omega_6\}$ (Fig. 2), and ω a new example to label, we can calculate the weight $\alpha(\omega_1,\omega)$ of ω_1,

$$\alpha(\omega_1,\omega) = \frac{V_D - V_{S_{\omega_1,\omega}}}{V_D} = \frac{d_1 * d_2 - \pi r^2_{\omega_1,\omega}}{d_1 * d_2} = 0.911$$

4.3 Prototype Selection Algorithm

Two types of algorithms exist for the building of geometrical graphs [3]:

1. *Total algorithms : in this case, neighborhood structures (Gabriel, Relative neighbors or Delaunay's Triangles) are applied on the whole sample. To build a new edge, some conditions must be imposed on the whole set. Thus, when a neighborhood is built, it is never suppressed.*
2. *Constructive algorithms : in this case, the graph is built point by point, step by step. Each point is inserted, generating some neighborhoods, deleting others. Thus, only a local update of the graph is necessary [4].*

For these two types of algorithms, the label of points to insert is not used. The prototypes selection algorithm presented in this section belongs to the second category but takes into account the label of points already inserted in the graph. It may thus only be used with supervised learning. Its principle is summuarized by the following pseudo-code.

```
– Let $\Omega_a$ be the original training sample and $\Omega^*$ be the set of
  selected proptotypes
– Initially, $\Omega^*$ contains one randomly selected example
– Repeat
    • Classify $\Omega_a$ with the Probabilistic Vote using the examples
      in $\Omega^*$.
    • Move misclassified examples into $\Omega^*$.
– until all examples remaining in $\Omega_a$ are well classified.
```

Thus, the pertinence of an example is defined as following : "*a point is pertinent if it brings information about its class*".

Interested readers may find the results of an application of our prototype selection technique on the well-known Breiman wave forms problem [2] in [16]. This results show that the selection technique allows ,on this problem, to cut by more than half the size of learning sample without lowering the generalisation accuracy of the built classification function.

5 Conclusion

The growing size of modern databases makes feature selection and prototype selection crucial issues. We have proposed in this article two algorithms to reduce the dimensionality of the representation space and to reduce the number of examples of a learning sample. Our approach is currently limited in that it supposes that examples are only described by numerical features. We are now working on new neighborhood structures to take into account symbolic data, without using euclidean distances.

References

1. Aha, D.W., Kibler, D., & Albert, M.K. Instance-based learning algorithms. Machine Learning 6(1):37-66, 1991.
2. Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, C.J. Classification And Regression Trees. Chapman & Hall, 1984.
3. Chassery, J.M., & Montanvert, A. Géométrie Discrète en Analyse d'Images. Hermès, 1991.
4. Devijver, P.A., & Dekesel, M. Computing Multidimensional Delaunay tesseletions. Research Report,1982.
5. Efron, B., & Tibishirani, R. An Introduction to the Bootstrap. Chapman & Hall,1993.
6. Gabriel, K.R., & Sokal, R.R. A new Statistical Approach to Geographic Variation Analysis, Systematic Zoology, 259-278. 1969.
7. Gates, G.W. The Reduced Nearest Neighbor Rule. IEEE Trans. Inform. Theory, 431-433, 1972.
8. Hart, P.E. The Condensed Nearest Neighbor Rule. IEEE Trans. Inform. Theory, 515-516, 1968.
9. Ichino, M., & Sklansy, J. The Relative Neighborhood Graph for Mixed Feature Variables, Pattern recognition ; ISSN 0031-3203 ; USA ; DA, (18):161-167, 1985.
10. Kohavi, K. Feature Fubset Selection as Search with Probabilistic Estimates, AAAI Fall Symposium on Relevance,1994.
11. LeBourgeois, F. & Emptoz, H. Pretopological Approach For Supervised Learning , 13th ICPR 96, Vienna Austria August 25-29, 256-260, 1996.
12. Parzen, E. On Estimation of a Probability Density Function and Mode. Ann. Math. Stat, (33):1065-1076, 1962.
13. Preparata, F.P., & Shamos, M.I. Pattern Recognition and Scene Analysis. Springer-Verlag.1985.
14. Rao, C. Linear Statistical Inference and its Applications. Wiley New York. 1965.
15. Sebban, M. Modèles théoriques en Reconnaissance de Formes et Architecture Hybride pour Machine Perceptive. Thèse de doctorat de l'Université Lyon1.1996.
16. Zighed, D.A. and Sebban, M. Sélection et Validation Statistique de Variables et de Prototypes. Revue Electronique sur l'Apprentissage par les Données, (2), 1998.
17. Skalak, D.B. Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms. Proceedings of 11th International Conference on MAchine Learning, Morgan Kaufmann, 293-301.1994.
18. Thrun etal. The Monk's Problem: a Performance Comparison of Different Learning Algorithms. Technical Report CMU-CS-91-197, Carnegie Mellon University, 1991.