Unsupervised Connectionist Clustering Algorithms for a better Supervised Prediction : Application to a radio communication problem

Laurent Bougrain, Frédéric Alexandre

Unsupervised Connectionist Clustering Algorithms for a better Supervised Prediction: Application to a radio communication problem

Laurent Bougrain and Frédéric Alexandre

LORIA, BP 239, 54506 Vandoeuvre Cedex, France
bougrain@loria.fr, falex@loria.fr

***Abstract*—Most** models concerned with real-world applications can be improved in structuring data and incorporating knowledge about the domain. In our problem of radio electrical wave dying down prediction for mobile communication, a geographic database can be divided in contextual subsets, each representing an homogeneous domain where a predictive model performs better. More precisely, by clustering the input space, a predictive model (here a multilayer perceptron) can be trained on each subspace. Various unsupervised algorithms for clustering were evaluated (Kohonen's maps. Desieno's algorithm, Neural gas, Growing Neural Gas, Buhmann's algorithm) to obtain classes homogeneous enough to decrease the predictive error of the radio electrical wave prediction.

I. Introduction

A modular approach is often chosen when a problem is too complex to be efficiently carried out by a single classification. The modules can be built and optimized in one step, like in mixtures of experts [4]. They can also be composed in several steps, each corresponding to a specific processing, like unsupervised and supervised classification. Such a combination of classifiers is explored here, in a real-world communication problem.

II. Data base

50,000 standardized patterns were extracted from a national geographic database, describing terrain in France, transmitting and receiving with 32 attributes. The data are split up into 40,000 patterns for the learning set and 10,000 patterns for the test set.

III. Categorization

In this approach, non-homogeneities in the distribution of input data are exploited to obtain a compression from the current pattern to an index determining the subspace on which a specific forecasting model will be applied to predict the attenuation of radio electrical waves. The index is the result of an unsupervised learning, called vector quantization, where each domain is characterized by a reference vector with the same dimensions as inputs. Reference vectors are some kind of centers of gravity. The input space is clustered. One input patterns belongs to a single cluster. The process is a competition between the reference vectors to determine which one is the best representation of the input to be categorized. Categorization is a way to better understand the world by reducing the information to the pertinent features. A wide range of methods exits and shows that there is no single algorithm available for all problems and producing consistently good results. In the next sections, some of the most useful algorithms are presented. Then each codebook (i.e. each reference vector set) is projected onto a 2-dimensional space to better visualize the effects of the particularity of each method on our database.

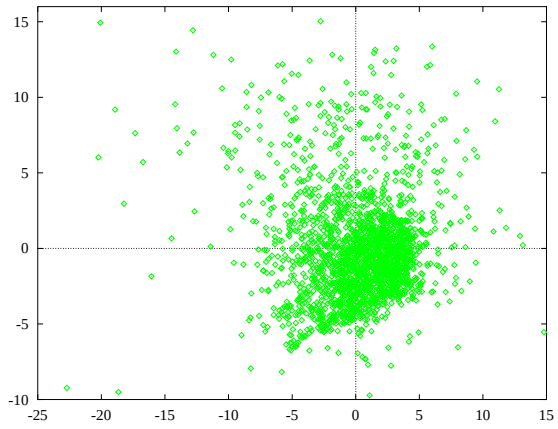

Fig. 1. Samples.

A. The self-organization map

Self-organization maps (SOMs) have been made popular by Kohonen. This algorithm is characterized by the following addition : A topological relation is fixed between the reference vectors to express topological relations in the inputs on the reference vectors [6]. During the training stage of the reference vectors, for each pattern, the winner (i.e. the closest vector) moves toward the pattern. The reference vectors are updated depend-

ing on a neighborhood function centered upon the winner and often chosen as a Mexican hat. Neighborhood and learning rate decrease with time to provide convergence of the reference vectors, seen as prototypes representing classes.

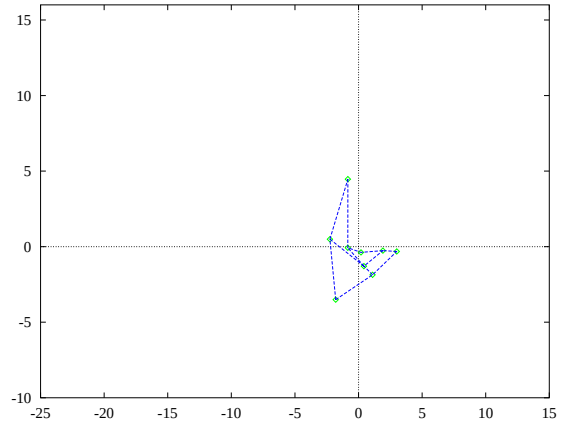

Fig. 2. Kohonen

B. A fair-play competition

DeSieno proposed an alternative solution to winner-take-all algorithms where the closest reference vector always moves toward the input vector [2]. So if some prototypes are initialized in a region with very low density or if several prototypes are very close, some of them may never win. To prevent this situation, it is possible to move all prototypes, depending of their distance to the input. The competition is not henceforth a winner-take-all one. The processing time is longer due to the multiplication of the updates. To solve this problem with a winner-take-all algorithm, DeSieno's algorithm penalizes the similarity metric with a conscience factor, based on frequency of victory. So, a prototype set further than a frequent winner can win and move. Forcing the prototype distribution to be non parametric is not good if the input distribution is not regular but by this way it is possible to prevent very unequal probabilities of selection. The prototypes are more representative of the input distribution.

C. Neural Gas

As explained in the previous subsection, updating all prototypes prevents some of them from being rather useless because they sometimes represent a small number of patterns. Neural gas algorithm (NG) is based upon an extension of the k-means clustering algorithm. Here, after ordering the reference vectors as a function of their distance to the current pattern, the updates of

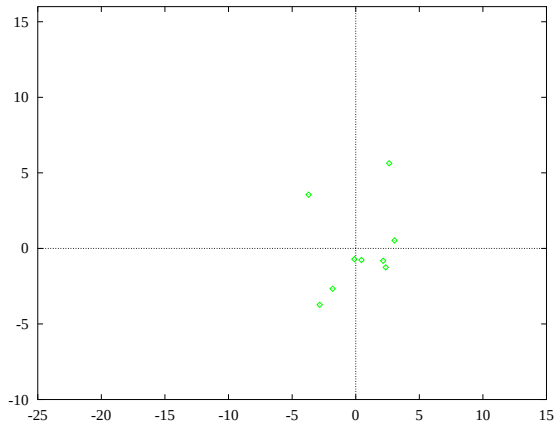

Fig. 3. DeSieno

the reference vectors are a function of their neighborhood ranking [8]. At the same time, a simple Hebbian learning rule determines the topology of the reference vectors [7]. When a pattern is presented, a connection is created, between the nearest and second-nearest output units, which is maintained over a period governed by a temporal parameter. The number of connections depends on this parameter. The topology is constructed and not fixed.

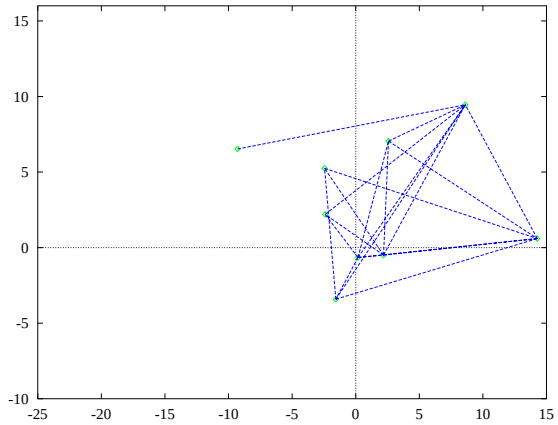

Fig. 4. NG

D. Growing Neural Gas

This model (GNG), presented by Fritzke [3], is an evolving cell structure version of the neural gas network. The topology is still adaptivly constructed adaptively through the same Hebbian learning. Only the winner and its neighbors are updated. Starting from two prototypes, new ones are progressively inserted near prototypes having the most accumulated error and its worst neighbor. The number of prototypes gradually increases up to a maximum number of classes. Old

connections are removed, the same for no longer making anymore connections.

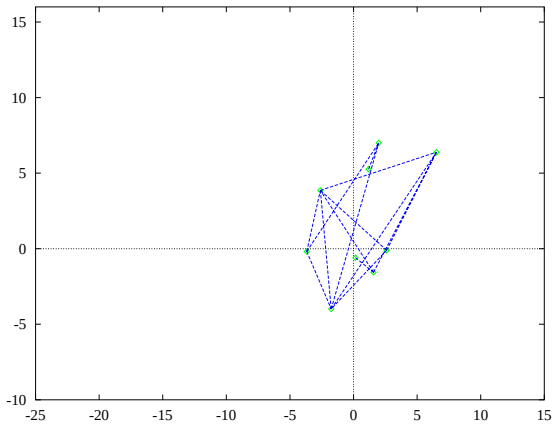

Fig. 5. GNG

E. Buhmann's model

This vector quantization by Buhmann and Kühnel (VQB) [1] adds a complexity cost to the distortion criterion. A maximum entropy estimation of the clustering cost function provides an optimal number of classes, the assignment frequency and the position of the reference vectors. Starting with one prototype corresponding to the center of gravity of the data, the existing prototypes are split, one by one, to test after a new the learning stage if the free energy of the configuration is smaller than before [5]. So a new prototype is created only if a high number of patterns are far away from their closest prototype, or if a few patterns are far away from every prototypes. The algorithm stops when all new configurations by splitting the reference vectors do not decrease the free energy.

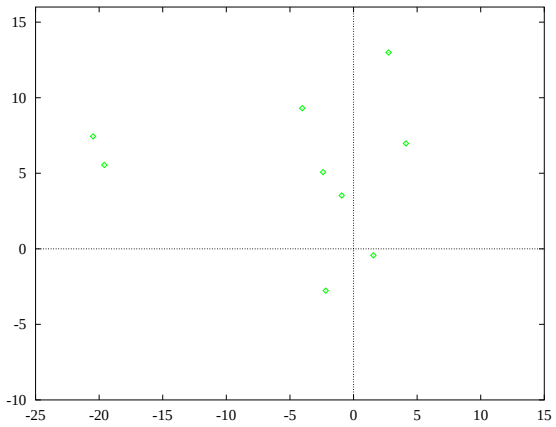

Fig. 6. VQB

F. 2-dimensional visualization

To have a visual information about how the input patterns are spread (figure 1) and how the prototypes of each algorithm cluster them (figures 2-6), a projection's algorithm by Sammon [9] was used. This algorithm tries to determine a mapping that preserves inter-point distance ratio after projection. To prevent a blow up computation, only 2000 patterns from the learning corpus and the prototypes from every method are projected together into a 2-dimensional space. Principal component analysis gives a linear projection where the information lost is minimal. Our goal is different. The input patterns are standardized. For all the attributes, the mean is equal to zero. So the center of gravity after projection should be kept near to the point (0,0). SOM and DeSieno's algorithm give prototypes gathered around the mean values (Fig 2 and 3). So the class densities are quite similar. NG and GNG (Fig 4 and 5) could have some prototypes isolated in a low density region by a particular initialization. VQB has its prototypes very spread in the input space (Fig 6).

IV. Assignment

A. Assignment rule

To know which predictive model to apply to the current pattern, a pattern assignment rule, only based upon the pattern values and information extracted from the learning pattern set, must be introduced. Discriminant analysis propose various predictive methods. The geometrical methods use a similarity measure or a distance measure in their assignment rule.

B. Similarity measure

For example, the centers of gravity $\overline{x}_k$ of the learning pattern set of class k and a dot product matrix M defining an Euclidean structure of R^q can constitute such a distance measure $d^2_M(x, \overline{x}_k)$. By taking one matrix M_k per class k, such as the inverse of the within class covariance W, the distance measure takes the cluster distribution of each class into account and ensures that the cohesion measure of the cluster is minimal. To compare the distances without a bias, M_k should be normalized. So the distance measure used is $d^2_{M_k}(x, \overline{x}_k)$ with $M_k = (detW_k)^{1/q} W_k^{-1}$ and W_k is the within class covariance matrix of class k. This distance is called Mahalanobis' distance. Taking the global covariance matrix makes the discriminant functions given by the assignment rule simpler and experience has shown that the results are not worse because patterns are noisy.
The assignment rule, based upon the closest Mahalanobis' distance measure between the current pattern

and each reference vector, is used to assign the pattern to a class for all clustering methods except vector quantization by Buhmann. In fact, Buhmann's algorithm uses the probability of each class to assign a pattern. It is a probabilistic method of assignment.
The number of patterns assigned to each class is presented in figure 7 examining the clustering methods. SOM and DeSieno's algorithm create classes with quite

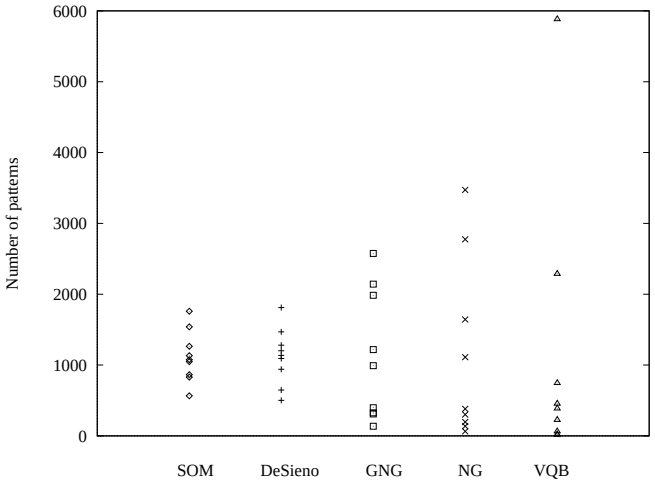

Fig. 7. Assignment

the same number of patterns, because of the neighborhood constraints when learning for SOM and because the equal probability constraints on class selection is balanced in DeSieno's algorithm. GNG and NG have a neighborhood function less large so the numbers of class can be different. The disparity of the number of patterns for VQB shows that this algorithm can create a new class just for some far away pattern or affect a lot of patterns to the same class if these patterns are close. Examining the number of patterns contained in each class can explain in practice the particularity of one method with regard to the others.

C. Within class inertia

A criterion to obtain a categorization with in average a good homogeneity within classes is to define a clustering such that the within class inertia $I_w = \sum_k P_k I_k$ is minimal (or such that the between class inertia I_b is maximal seeing that the total inertia is constant for all clusterings). Actually, the total inertia I is computed from patterns only : $I = \frac{1}{2} \sum_k \sum_{i,j \in C_k} p_i p_j d_M^2(x_i, x_j)$ where p_i and p_j are the weights of pattern i and j). This criterion can be useful only if the number of classes is fixed, because if there is as many classes as patterns, I_w is minimal and equal to zero. It is better to maximize the inertia between the classes because the metric does not intervene in the formula.

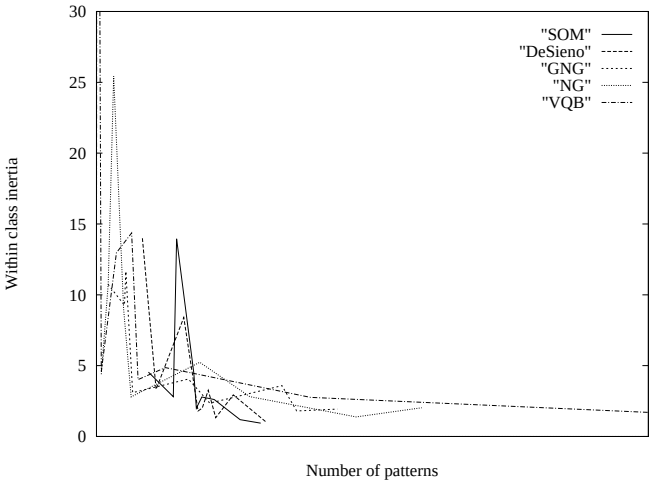

Fig. 8. Relation between the number of patterns and I_w

Referring to figure 8, the number of patterns and I_w seems to be correlated. This observation is confirmed by the correlation coefficient of -0.35 given with a significant level of 0.02 (which indicates that the null hypothesis of zero correlation is disproved when the value is high). This is true in particular with DeSieno's classes and GNG's classes (correlation coefficient $r < -0.7$). For all methods, the within class inertia goes down when the number of pattern grows up. So, for this application, it seems better to have no large differences

V. Forecasting stage

The aim of this study is to observe the interest of clustering complex data set to make the task of forecasting easier. The conclusion of the study will be valid whatever the prediction model. Here, a standard multilayer perceptron is used to predict the attenuation of radio electrical waves from the current pattern and a supervised value. One MLP is trained on one class. To evaluate the global performance of each clustering algorithm, all classes merged, some measures are introduced. Then the results are reported in a comparison table. Finally, an analysis is presented.

A. Model

A multilayer perceptron is a good model to approximate a function where the data of the problem are real attributes and where the desired value is known. All multilayer perceptrons have a similar architecture with 32 input neurons corresponding to the attributes extracted from the current situation, 10 neurons in the single hidden layer and 1 output neuron, the attenuation of radio electrical waves to be predicted. The supervised learning is realized by the backpropagation algorithm.

B. Measures

Three measures are used to evaluate the global performance starting from the local measure of each class.

1) Mean square error:

$$\mu = E(X) = \sum_k P_k E(X_k)$$

where P_k and $E(X_k)$ are the weight and the mean square error of class k.

2) Standard deviation:

$$\sigma = \sqrt{\sum_k P_k(var(X_k) + E^2(X_k)) - E^2(X)}$$

where P_k, $E(X_k)$ and $var(X_k)$ are the weight, the mean square error and the variance of class k.

3) Confidence interval:

$$I(\alpha, N) = \frac{T + \frac{Z_\alpha^2}{2N} \pm Z_\alpha \sqrt{\frac{T(1-T)}{N} + \frac{Z_\alpha^2}{4N^2}}}{1 + \frac{Z_\alpha^2}{N}}$$

where N is the corpus size, T the performance, $Z_\alpha = 1.96$ if $\alpha = 95\%$

C. Results

Firstly, the following table (table I) expresses the influence of the number of classes upon the performance.

Results	Learning		Test	
	μ	σ	μ	σ
SOM with 6 classes	3.58	4.61	3.59	4.69
SOM with 9 classes	3.36	4.45	3.48	4.57
SOM with 16 classes	3.27	4.23	3.51	4.65

TABLE I

Influence of the number of classes

Prediction obtained, after clustering data in 6 classes, has worse results than with a higher number of classes. The data seems to be too complex to be gathered in a low number of classes. 16 classes clustering does not improve the performance. When the number of classes is high, the number of patterns contained in each class is smaller. The learning data set is not big enough to have a good generalization of the model and the test data set is too small for having a good estimation of the error.

A 9 classes clustering is chosen for all algorithms. Their performances are reported (tables II et III).

Results	Learning		Test	
	μ	σ	μ	σ
Kohonen's maps	3.36	4.45	3.48	4.57
DeSieno's algorithm	3.41	4.41	3.52	4.58
Buhmann's algorithm	3.60	4.70	3.67	4.78
Neural Gas	3.58	4.63	3.59	4.69
Growing Neural Gas	3.49	4.53	3.58	4.73
without classification	4.10	5.24	4.03	5.18

TABLE II

Mean Square Error (dB)

D. Analysis

Desired values have a mean equal to 130.53 dB and a standard deviation equal to 16.07 dB. Process without classification gives reasonably good forecastings but large standard deviations.

Firstly, all processes applied to an input data clustering improve both the performances (table II). The prediction error can be reduced of 0.5 dB and the standard deviation decreases of 0.6 dB. The predictions become more precise and constant. The performance improvement is significant examining the confidence interval (table III) and the statistical F-test (which disproves the null hypothesis of equal variance).

Results	Learning		Test	
	perf(%)	+/-	perf(%)	+/-
Kohonen's maps	97.43	0.155	97.33	0.315
DeSieno's algorithm	97.38	0.157	97.30	0.317
Buhmann's algorithm	97.24	0.161	97.19	0.323
Neural Gas	97.26	0.160	97.25	0.320
Growing Neural Gas	97.33	0.158	97.26	0.319
without classification	96.86	0.171	96.91	0.338

TABLE III

Confidence Intervals

Secondly, the clustering method performances are not significantly different. So the best clustering method will not be chosen because it has the best performance but because their particularities help us to apply some additive algorithms according to their characteristics (for examples, a special subdomain density distribution or a topology between the domains). We have to consider that the Buhmann's algorithm had to be parametred to obtain as many classes as the other algorithms. Its performance, lower than the other ones, could be explained by this reason. But the unequal distribution of their subdomains seems to be a better reason. Figure 9 shows the relation between the number of patterns and the forecasting error. Subdomains which contain a little number of patterns can have very high or very low performance. But the global error is firstly determined

by the performance of subdomains which contain a lot of patterns and the performance of the largest class of VQB has a higher error. We have seen in section IV.C that the correlation between the number of patterns and the within class inertia is negative, but we can not conclude that the performance are correlated with the within class inertia. Another consideration can explain the good performance of clustering methods with quasi-equal distribution probability: The learning stage must have enough patterns to have a good generalization and the test stage must have enough patterns to have significant measures.

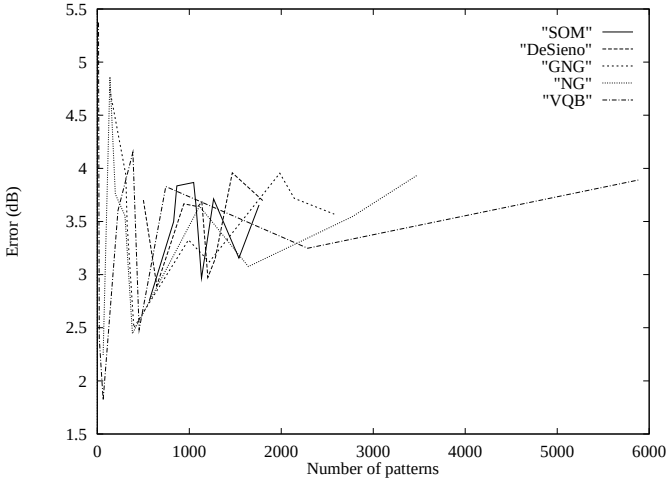

Fig. 9. relation between the number of patterns and the forecasting error

VI. Conclusion

Our work evaluates some of the most useful unsupervised neural networks on a predictive real-word problem of radio communication. Data structure mapping representation and statistical analysis are used to put in a prominent the global properties of each algorithm. A specific training improves the prediction whatever the method. The best unsupervised algorithm can be chosen knowing the results and the properties of topological conservation and number of classes variability. Various predictive models such as recurrent networks, where the prediction of the neighbors is used to correct the prediction, were evaluated with some better performances than a MLP. Then, training one recurrent network by cluster should be an attractive development because additional information based upon the neighbors will be more significant. Hybrid systems, combining classifiers or using Mixture of Experts might improve the performances by mixing the prediction of each class for a pattern.

References

[1] J. Buhmann and H. Kühnel. Complexity optimized vector quantization: A neural network approach. *Proceedings of data compression conference*, 1992.

[2] D. Desieno. Adding a conscience to competitive learning. *IEEE International Conference on Neural Networks*, 1:117–124, 1988.

[3] B. Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press, Cambridge MA, 1995.

[4] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214, 1994.

[5] S. Kirkpatrick, C. G. Jr., and M. Vecchi. Optimization by simulated annealing. *Science*, 220, 1983.

[6] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, 3 edition, 1989.

[7] T. M. Martinetz. Competitive hebbian learning rule froms perfectly topology preserving maps. In *ICANN'93: International Conference on Artificial Neural Networks*, pages 427–434, Amsterdam, 1993. Springer.

[8] T. M. Martinetz and K. J. Berkovich, S. G.and K. J Schulten. Neural-gas network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4(4):558–569, 1993.

[9] J. W. J. Sammon. A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.*, 5:401–409, 1969.