Quality measure based on Kohonen maps for supervised learning of large high dimensional data

Elie Prudhomme and Stéphane Lallich

Laboratoire E.R.I.C, Université Lumière Lyon 2
5, avenue Pierre Mendès-France,
69676 BRON Cedex, France
(e-mail: stephane.lallich@univ-lyon2.fr,
elie.prudhomme@etu.univ-lyon2.fr)

Abstract. In supervised learning, the prediction of the class is the ultimate goal. On a broader basis, a good learning methodology is expected to (1) enable a representation of the data in order to facilitate user's navigation within the data set and (2) contribute to the choice of examples and attributes, while ensuring a structured, understandable prediction. Various studies have shown how the so-called neighbourhood graph, from the predictors, gives ground to such a methodology (e.g.: the relative neighbourhood graph of Toussaint). However, the construction of such a graph (O(n^3)) remains complex. Moreover, when the number of dimensions increases, distance becomes hard to compute and lose their selectivity.

In the case of large high dimensional dataset, we propose to substitute a self-organized map built on the predictors to the neighbourhood graph. After a short reminder on the principles of the SOM for unsupervised learning, we analyse how it can found an optimized strategy of learning. Then we propose to use original statistics (narrowly correlated with the error in generalization) in order to assess the level of quality of this strategy. Diverse experiments highlight the feasibility of this approach, therefore reliable criterion are available for us to select relevant examples and attributes.

Keywords: supervised learning, Kohonen maps, statistical validation.

1 Motivation

Supervised learning methods of a categorical variable aim at predicting the class of a new instance from a sample of labelled examples. Indeed, prediction is only a step in the learning process, which is enriched through the exploratory analysis of the data. This allows to clean and transform the data, to select features and subsets of records, and to detect outliers, while integrating possible contextual information.

In such a perspective, resorting to neighbourhood graphs brings an effective solution. One builds the neighbourhood graph based on the predictors, for example the Relative neighbourhood Graph of Toussaint [1980] (*RNG*). The vertices of the graph are then colored according to the class they belong to. To find the class of a new instance, it is first inserted in the neighbourhood graph and then it is attributed the majority class among its neighbours on the graph. Various studies proposed a statistic: the cut edge weight statistic. This statistic evaluates the predictive capacity of a neighbourhood graph. It also allows for the selection of relevant variables or for the detection of outliers by spotting the impact of an example or of a variable on the predictive capacity of the graph ([Sebban, 1996], [Zighed *et al.*, 2001], [Lallich, 2002], [Muhlenbach *et al.*, 2003], [Zighed *et al.*, 2004]).

In comparison with the k-Nearest Neighbour method (kNN), neighbourhood graphs adapt the number of nearest neighbours to the local topology. On those graphs, the cut edge weight statistic that evaluates their predictive capacity is strongly correlated to their error rate in generalization. Their results in generalization are at least as good and they have the advantage of establishing an effective procedure of navigation within the basis of examples. Furthermore, the neighbourhood graph allows to navigate efficiently in the database, making the exploratory analysis of the data easier.

Neighbourhood graphs present a double difficulty when confronted to large high-dimensional datasets. Firstly, their great complexity - $O(n^3)$ for Relative Neighbourhood Graphs of Toussaint - makes them poorly adapted to very large datasets. The second issue is linked to the curse of dimensionality which triggers a loss of selectiveness of euclidean distance.

Faced with this double difficulty, we propose to replace the *RNG* issued from the predictors with a Self Organized Map (*SOM*). We thus get a representation of the information given by the predictors. That method has the advantage of preserving the local topology in case of high dimensional data while using a complexity which varies linearly with the number of examples. The advantages of neighbourhood graphs are also maintained in the *SOMs*: especially the spatialisation of the information obtained from the predictors and the efficient navigation in the database.

In this article, we show that it is possible to construct a cross-product statistic which is closely linked to the predictive ability of the map in generalization. This statistic has the advantage of helping us in data preparation, especially to select relevant variables or detect outliers. After presenting the notations we used (see section below), we introduce the SOM algorithm and its use in supervised learning (section 3). Then we present our cross-product statistics estimates in *SOM* (section 4). Their validation on different datasets is presented in section 5.

2 Notations

- m: number of examples, d: number of predictors, p: number of classes, n: number of neurons.
- X: (m, d) matrix of data; line i corresponds to example i and column j to predictor j.
- y: vector with m components indicating the class of each example.

- W: (n, d) matrix of general term w_{ij}, designating the weight of neuron i for predictor j.
- c: vector with n components indicating the class of each neuron; $c_i = 0$ if neuron i is ambiguous, $c_i = -1$ if neuron i is empty.
- $bmu_i = \arg\min_r \|w_r - x_i\|$, index of *best matching unit*, the nearest neuron to example i.
- $dist_c(r, q)$: distance between neurons r and q, according to the map.
- $dist_p(r, q)$: Euclidean distance between the weights of neurons r and q.
- PPV: (n, n) symmetrical matrix of general term ppv_{ij}, worth 1 if $dist_c(i, j) \leq \max\left(dist_c(i, k), dist_c(j, k)\right), \forall k, k \neq i, k \neq j; c_i, c_j, c_k \neq -1$ $(i, j$ connected), 0 otherwise; ppv_{r+} represents the number of neurons connected to neuron r.

3 *SOM* and supervised learning

The Self Organized Map allows i) a fast unsupervised learning of input examples and ii) their representation. The map is built on a uniform distribution of neurons in 2 or 3 dimensions. Each neuron is associated to a vector in the space of the example. Originally, that association was called a model. During the learning, the input examples are successively presented to the map. Assuming a general distance measure between inputs and models (usually euclidian distance), the neuron the nearest to the input (called the Best Matching Unit) is modified with its neighbourhood so that all of them get closer to the input example.

The iterative algorithm for the input example i at time t is summarized by the following formula updating the weights W of the neuron r:

$$w_r^{t+1} = w_r^t + h_r^t \times (x_i - w_r^t)$$

where $h_r^t = \alpha^t \times v_r^t$, with α^t the learning-rate factor and v_r^t the neighbourhood function which represents the size of the modified neighbourhood. Both α^t and v_r^t are monotonically decreasing as a function of time.

This algorithm ensures a local preservation of the topology through a non linear projection. Thus, after learning, two close input examples will have close models on the *SOM*. Nevertheless this non linear projection is particular in the sense that it does not preserve the distances from the input space.

Because of those properties (fast algorithm and topology preservation) some authors have adapted them to a supervised learning. The most popular of those algorithms is the *LVQ* proposed by Kohonen [1988]. Here, the classes of the input examples are used to control the modification of the models. Another idea is used by Midenet [1994] in the LASSO model. In that case, the classes are used during the learning phase in the same way as other input variables. Two phenomena result in the use of classes during learning. First, the prediction is more robust: more information is used. But at the same time the local topology preservation is changed. It is not simply a function of the input variables (as in the original *SOM* algorithm) but also a function of the classes.

To avoid that problem some authors have proposed a different approach. On that account, the class of the input example is only used after a classical learning of the *SOM* on the input variables. During that second step, the neurons take the class of the inputs they represent. The reverse happens during prediction: the class of a new input example is determined by the class of the best matching unit of that example. Three methods use that principle: Kohonen-KNN [Zupan *et al.*, 1994], Kohonen-WI [Song and Hopke, 1996] and Kohonen-Opt [Prudhomme and Lallich, 2005]. There are at least two cases which show the difference between those three approaches. First empty neurons: after the learning phase some neurons do not match any input example. Secondly ambiguous neurons: after the learning phase some neurons match the same proportion of examples from different classes. So each method proposes a way of predicting a new example which matches one of those two type of neurons. Prudhomme and Lallich [2005] have shown that Kohonen-Opt generally gives better results in generalisation than the others. Moreover, the results obtained with Kohonen-Opt on different datasets are almost equivalent to those obtained by the ID3 method of classification.

Consequently, *SOMs* could be used in supervised learning. In that case there is a double advantage. First the non linear projection is particulary adapted to high dimensional spaces. It allows a dimension reduction based on the most significant feature. Secondly the examples are synthetically represented by the models. Thus the *SOM* representation is well adapted to large datasets. In the rest of the document we propose a statistic which takes advantage of those two points in order to assess the predictive capacity of the *SOM*. Because this statistic is based on the neighbourhood, distance preservation is not mandatory.

4 Quality measures for SOM under supervised learning

We therefore suggest a learning strategy that relies on the construction of the *SOM*. The reliability of the *SOM*, reagrdless of any consideration of class, can be assessed through various statistical tools proposed notably by [Bodt *et al.*, 2002]. We suggest here an assessment of the predictive ability of the *SOM* through different statistics. We will experimentally show the strong correlation of those statistics with the precision in generalization. Similarly to the cut edge weight statistic worked out for neighbourhood graphs [Lallich, 2002], those different statistics are based on the notion of cross-product statistic [Mantel, 1967]. Thus they are constructed as the scalar product of two proximity measures, the first one depending on the predictors and the other one depending from the class.

4.1 Definition of J type statistics

To assess the strength of the link between proximity in the sense of the map and proximity in the sense of the classes, one can reason about examples or neurons. Reasoning about neurons helps to deal with a large amount of examples.

When reasoning on examples, the proximity between examples based on the map is assessed by the matrix T' of general term t'_{ij}, which is worth 1 if the examples i and j are represented by the same neuron, and 0 otherwise. In order to take into account the topological properties of the map, one also can resort to the matrix T'' of general term t''_{ij}, which is worth 1 if the examples i and j are represented by the same neuron, $norm(dist_p(w_{bmu_i}, w_{bmu_j}))$ if i and j are represented by adjacent neurones ($i.e.\ dist_c(bmu_i, bmu_j) = 1$), and 0 otherwise. The proximity between examples based on the class is assessed by the matrix U of general term U_ij, which is worth 1 if the examples i and j do not have the same class (i.e $c_i \neq c_j$), and 0 otherwise.

When reasoning on neurons, the proximity between neurons based on the map is assessed by the matrix T''', of general term t'''_{ij}, which is worth $norm(dist_p(w_i, w_j))$ if $ppv_{ij} = 1$, and 0 otherwise. The proximity between neurons based on the class is assessed by the matrix R, of general term r_{ij}, which is worth 1 if the neurons i and j do not have the same class, 0 otherwise.

As a result, one will obtain three different statistics, J', J'' and J''' which are defined below.

J'	**J"**	**J"'**
$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} T'_{ij}U_{ij}$	$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} T''_{ij}U_{ij}$	$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} T'''_{ij}R_{ij}$

The following simplifying notations are used, where T can take the value of T',T'' or T''' and the sums finishing respectively in m for the two former cases and in n for the latter one:

S_0	S_1	S_2
$\sum_{i=1}\sum_{j=1} t_{ij}$	$\frac{1}{2}\sum_{i=1}\sum_{j=1}(t_{ij} + t_{ji})^2$	$\sum_{i=1}(t_{i+} + t_{+i})^2$

J Type statistics vary between 0 and $\frac{1}{2}S_0$. They are the weakest when the link between proximity according to the class and proximity according to the map is strongly positive. They may be standardized by forming $2J/S_0$ which varies between 0 and 1.

Dataset	**Variables**	**Classes**	**Example**	**Dimensions**	**Times**
(1) Abalone	8	29	4177	25 × 25	90000
(2) Balance Scale	4	3	625	15 × 15	60000
(3) Breast Cancer	9	2	699	20 × 20	90000
(4) Glass Indent	9	6	214	10 × 10	10000
(5) Haberman	3	2	306	10 × 10	20000
(6) Ionosphère	34	2	351	10 × 10	20000
(7) Iris	4	3	150	10 × 10	2000
(8) Italian Olive Oil	9	9	572	15 × 15	45000
(9) Liver	6	2	345	10 × 10	35000
(10) Yeast	8	10	1484	25 × 25	90000

Table 1. dataset and associate parameters

4.2 Meaning of J type statistics

In order to know to which extent the evaluation given by J is not due to chance, a random multinomial outline was defined. The null hypothesis (H_0) was that the examples (the neurons) are labelled independently from each other, with the same probability distribution $(\pi_r)_r$ where π_r denotes the frequency of the class $y_r, r = 1, 2, \ldots, p$.

The significance of the observed value of J is appraised with its left unilateral p-value. This is the probability of getting a value of J as extreme as or more extreme than the observed one if H_0 is true. That calculation can be done either by simulation or more quickly by normal approximation [Cliff and Ord, 1981]. In the last case, we have to calculate $\mu = E(J/H_0)$ and $\sigma^2 = Var(J/H_0)$. It is easy to calculate $\mu = S_0 \sum_{r=1}^{p-1} \sum_{s=r+1}^{p} \pi_r \pi_s$. One can find in Lallich [2002], following Cliff and Ord [1981], the calculation of the variance σ^2, which depends on S_0, S_1 and S_2.

5 Experiment

Those different statistics were tested on 10 datasets, coming from the repository of the University of Irvine [Blake and Merz, 1998] (except for one *Italian Olive Oil* which is from [Hopke and Massart, 1993]). Table 1 details those datasets in terms of the number of input variables for each example, the number of classes and the number of input examples in each dataset. This table also summarizes some parameters of the *SOM* used for learning: the total number of input examples presented (called time) and the size of the *SOM*. The algorithm used for learning is the classical one presented in section 3. Table 2 shows the value of each statistic, their associated *p-value* and the error rate in generalization with the *Kohonen-Opt* method.

The *p-values* are significant ($p < 0,05$) for J'', and for J' (except for *Haberman* where $p = 0.08$). Thus they are sufficiently robust to assess the

Base	$2J'/S_0$	**p-value**	$2J''/S_0$	**p-value**	$2J'''/S_0$	**p-value**	**Error**
(1)	79,96	0	79,88	0	81,97	1	73,86
(2)	21,66	0	23,68	0	22,28	0	17,3
(3)	0,40	0	1,10	0	8,30	0	3,21
(4)	43,29	0	44,64	0	61,65	0,02	34,21
(5)	34,57	0,078	32,51	0,005	28,20	0,022	24,06
(6)	10,92	0	14,76	0	36,11	0,022	11,6
(7)	3,60	0	4,70	0	8,50	0	4,67
(8)	41,40	0	8,05	0	20,06	0	7,69
(9)	60,58	0,0031	0,3750	0	58,28	0,9989	37,53
(10)	50,00	0	52,40	0	64,52	0	47,53
Means	31,64	0,0081	29,92	0,0005	27,99	0,2045	

Table 2. Statistic, their associated *p-value* and error rate in test with Kohonen-Opt

quality of the representation built by the *SOM*. In the case of J''', two *p-values* are almost equal to 1. For this statistic, the link between two ambiguous neurons is a cut edge one. In the two cases, the graph extracted from the *SOM* has a many ambiguous neurons. So, in the statistic sense, the class of the neuron is independent from the topology of the map. For that reason, the *p-value* is high. In fact, this happened only when the error rate was high too.

A more interesting property is the correlation between that statistic and the error rate in generalization. r^2 of this correlation is respectively 0.78, 0.98 and 0.88 for J', J'' and J'''. J' just takes into account the input example of different classes matching the same neurons. So this statistic does not use the information contained in the local topology of the *SOM*. That information is used by J''. For that reason that statistic has a better correlation with the error rate. The correlation between J''' and the error rate is intermediate. That statistic takes into account the local topology of the *SOM* thanks to the the neighbourhood graph which was built on the map. On the other hand, the input examples are not used. Therefore some information is lost during the projection of the input space on the map. However the estimation of that statistic has a low complexity as only the neurons are used. In the case of datasets composed by a high number of examples, it is an interesting property. On the contrary, J'' needs the examples.

Moreover, we have tested the capacity of that approach to be applied on large datasets. Therefore, we used Wave [Blake and Merz, 1998], which allows to randomly generate a user fixed number of input examples. For each generated dataset, the error rate in generalization is know and constant. We applied Kohonen-Opt and our statistics on different datasets containing 5 000 to 1 280 000 examples. The learning time was reported on table 3. The *SOM* used for each dataset is the same and the test was made on the same dataset of 100 000 examples, never used in learning.

The table 3 shows the results. First, the error rate in generalization is stable regardless of the number of input examples (approximatively 15%). Secondly the time needed for learning increases linearly from a factor 2 (like the number of examples). Finally, the statistics ($2J/S0$) are stable too with a little decrease when the number of examples increases. Their *p-values* are always equal to 0.

This experiment shows that the quality of the learning by the *SOM* does not decrease when the number of examples increases. Thus they could be used in the case of large datasets. This is also the case for the proposed statistics which are relatively stable.

Size	$2J'/S_0$	$2J''/S_0$	$2J'''/S_0$	**Error**	**Time (s)**
1250	26,67	26,27	17,80	16,03	2
2500	26,08	26,09	13,39	15,70	5
5000	24,57	24,92	9,87	15,46	10
10000	24,45	24,36	9,44	15,57	20
20000	23,25	23,68	7,42	14,92	41
40000	22,65	23,04	7,78	14,84	78
80000	22,64	23,22	7,40	15,25	127
160000	22,53	23,03	7,45	14,93	245
320000	22,94	23,37	7,92	15,04	500
640000	22,54	23,09	7,18	15,17	1073
1280000	22,32	22,94	7,98	15,22	2014

Table 3. Statistic, their associate *p-value* and error rate in test with Kohonen-Opt on different Waves dataset

Finally, we have tested the capacity of that approach on high dimensional datasets. Here we use the Forest CoverType dataset [Blake and Merz, 1998]. That dataset presents 54 input variables for 8 classes. Moreover the classification performance on that dataset is known. It was obtained by Blackard [1998] for neural networks and linear discriminant analysis.

Table 4 shows those results and those obtained with Kohonen-Opt. A direct application of Kohonen-Opt on this dataset gives poor results. To avoid that problem, a normalization of the attributes was carried out i) with the Milligan and Cooper (*MC*) procedure [1988] and ii) with a standardization by removing the mean and dividing by the standard deviation (s). Since attributes are both boolean and continuous, the MC procedure gives better results. In that case, the error rate is in the same order as the one obtained by the neural network. That result tends to show that the learning based on the *SOM* is robust when the number of input variables increases.

	Kohonen-Opt			Other	
Method	None	s	MC	ANN	linear discriminant
Error Rate	45,7	43,4	32,2	30	42

Table 4. Result in classification task on Forest CoverType dataset

6 Conclusion

SOMs are popular algorithm in unsupervised learning. Their complexity is linear with the number of example and they allow for a data exploration [Lechevallier, 2002]. In that paper we suggested that they can be used in supervised learning. In that case *SOMs* synthesize the information of the predictors through a non linear projection and enable a navigation through the dataset. Even if that non linear projection does not maintain the distance, it is nevertheless a way to assess our statistic $(2J/S_0)$ which is correlated to the error rate.

In further work we want to use that statistic for outliers detection and feature selection from large high dimensional datasets. In addition, we want to test the effect of the choice of the distance on the learning process. We hope to show that fractional distance metrics are more useful than euclidian distance to learn high dimensional datasets with *SOMs*, as it is the case for k-means [Aggarwal *et al.*, 2001].

References

[Aggarwal *et al.*, 2001]Charu C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. *Lecture Notes in Computer Science*, 1973:420–434, 2001.

[Blackard, 1998]Jock A. Blackard. *Comparison of Neural Networks and Discriminant Analysis in Predicting Forest Cover Types.* Ph.d. dissertation, Department of Forest Sciences. Colorado State University., Fort Collins, Colorado, 1998.

[Blake and Merz, 1998]C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[Bodt *et al.*, 2002]E. Bodt, M. Cottrell, and M. Verleysen. Statistical tools to access the reliability of the self organizing maps. *Neural Network*, 15:967–978, 2002.

[Cliff and Ord, 1981]A. D. Cliff and J. K. Ord. *Spatial processes, models & applications.* London, 1981.

[Hopke and Massart, 1993]P. K. Hopke and D. L. Massart. Reference data sets for chemometrical methods testing. *Chemometrics and Intelligent Laboratory Systems*, 19:35–41, 1993.

[Kohonen, 1988]T. Kohonen. Learning vector quantization. *Neural Network*, 1:303, 1988.

[Lallich, 2002]S. Lallich. *Mesure et validation en extraction des connaissances à partir des données.* Habilitation à diriger les recherches, Université Lumière Lyon 2, Lyon: France, 2002.

[Lechevallier, 2002]Y. Lechevallier. Construction de super-classes à partir de la carte de kohonen et indicateurs de qualité de cette carte, séminaire laboratoire eric, http ://www-sop.inria.fr/axis/talks/∼eric/, 2002.

[Mantel, 1967]N. Mantel. The detection of disease clustering and a general regression approach. *Cancer Res.*, 27:209–220, 1967.

[Midenet and Grumbach, 1994]S. Midenet and A. Grumbach. Learning associations by self-organisation : the lasso model. *Neurocomputing*, 6:343–361, 1994.

[Milligan and Cooper, 1988]G. W. Milligan and M. C. Cooper. A study of standardization of variables in cluster analysis. *Journal of Classification*, 5:181–204, 1988.

[Muhlenbach *et al.*, 2003]F. Muhlenbach, S. Lallich, and D.A. Zighed. Identifying and handling mislabelled instances. *Journal of Information Intelligent Systems*, 22:89–109, 2003.

[Prudhomme and Lallich, 2005]E. Prudhomme and S. Lallich. Validation statistique des cartes de kohonen en apprentissage supervisé. In *5èmes Journées Francophones d'Extraction et de Gestion des Connaissances (EGC 05), Paris*, Revue des Nouvelles Technologies de l'Information, Janvier 2005.

[Sebban, 1996]M. Sebban. *Modèles théoriques en reconnaissance des formes et architecture hybride pour machine perceptive.* Thèse de doctorat en informatique, Université Lumière Lyon 2, Lyon: France, 1996.

[Song and Hopke, 1996]X-H. Song and P. K. Hopke. Kohonen neural network as a pattern recognition method based on the weight interpretation. *Analytica Chimica Acta*, 334:57–66, 1996.

[Toussaint and Menard, 1980]G. T. Toussaint and R. Menard. Fast algorithms for computing the planar relative neighborhood graph. In *Methods of Operations Research, Proceedings of the Fifth Symposium on Operations Research*, pages 425–428, 1980.

[Zighed *et al.*, 2001]D. A. Zighed, S. Lallich, and F. Muhlenbach. Séparabilité des classes dans r^p. In *Actes du VIIIème Congrès de la Société Francophone de Classification (SFC'01)*, pages 356–363, 2001.

[Zighed *et al.*, 2004]D. A. Zighed, S. Lallich, and F. Muhlenbach. A statistical approach of classes separability. In H. Mannila et H. Toivonen T. Elomaa, editor, *Revue Applied Stochastic Models in Business and Industry*, pages 475–487. Springer-Verlag, 2004.

[Zupan *et al.*, 1994]J. Zupan, M. Novic, X. Li, and J. Gasteiger. Classification of multicomponent analytical data of olive oils using different neural networks. *Analytica Chimica Acta*, 292:219–234, 1994.