Robust Self-organizing Maps*

Héctor Allende[1], Sebastián Moreno[1], Cristian Rogel[1], and Rodrigo Salas[1,2]

[1] Universidad Técnica Federico Santa María,
Dept. de Informática, Casilla 110-V, Valparaíso-Chile
{hallende,smoreno,crogel,rsalas}@inf.utfsm.cl
[2] Universidad de Valparaíso, Departamento de Computación
Rodrigo.Salas@uv.cl

Abstract. The Self Organizing Map (SOM) model is an unsupervised learning neural network that has been successfully applied as a data mining tool. The advantages of the SOMs are that they preserve the topology of the data space, they project high dimensional data to a lower dimension representation scheme, and are able to find similarities in the data.
However, the learning algorithm of the SOM is sensitive to the presence of noise and outliers as we will show in this paper. Due to the influence of the outliers in the learning process, some neurons (prototypes) of the ordered map get located far from the majority of data, and therefore, the network will not effectively represent the topological structure of the data under study.
In this paper, we propose a variant to the learning algorithm that is robust under the presence of outliers in the data by being resistant to these deviations. We call this algorithm Robust SOM (RSOM). We will illustrate our technique on synthetic and real data sets.

Keywords: Self Organizing Maps, Robust Learning Algorithm, Data Mining, Artificial Neural Networks.

1 Introduction

The Self-Organizing Map (SOM) was introduced by T. Kohonen [7] and is one of the most popular neural network models. The SOM has proven to be a valuable tool in data mining and in Knowledge Discovery Database (KDD) with various engineering applications in pattern recognition, image analysis, process monitoring and fault diagnosis.

The success of the SOM is due to its special property of effectively creating spatially organized *internal representations* of various features of input signals and their abstractions [6]. The SOM quantizes the data space formed by the training data and simultaneously performs a topology-preserving projection of the data onto a regular low-dimensional grid. The grid can be used efficiently in visualization. The SOM implements an ordered dimensionality-reducing map of the data and follows the probability density function of the data.

* This work was supported in part by Research Grant Fondecyt 1040365 and 7040051 and in part by Research Grant DGIP-UTFSM

A. Sanfeliu et al. (Eds.): CIARP 2004, LNCS 3287, pp. 179–186, 2004.

In real data there may exist outliers, data items lying very far from the main body of the data. Neural networks are not robust to the presence of outliers as we have shown in early work [1]. It is also possible that the outliers are not erroneous but that some data items really are strikingly different from the rest, for this reason it is not advisable to discard the outliers and instead special attention must be paid.

In this paper, we show that the SOM is not robust and we propose a variant to the learning algorithm that diminishes the influence of outliers, but still considers them during the training. We call this model RSOM (Robust Self Organizing Maps). The remainder of this paper is organized as follows. The next section briefly presents the Kohonen SOM algorithm. In the third section, we will give a detailed discussion on our method of generating a feature map that is robust to the presence of outliers in the data. Simulation results on synthetic and real data sets are provided in the fourth section. Conclusions and further work are given in the last section.

2 Self-organizing Maps

The SOM may be described formally as a nonlinear, ordered, smooth mapping of high-dimensional input data manifolds onto the elements of a regular, low-dimensional array.

The self-organizing maps (SOM) algorithm is an iterative procedure capable of representing the topological structure of the input space (discrete or continuous) by a discrete set of prototypes (*weight vectors*) which are associated to neurons of the network. The SOM maps the neighboring input patterns onto neighboring neurons.

The map is generated by establishing a correspondence between the input signals $\underline{x} \in \chi \subseteq \mathbb{R}^n$, $\underline{x} = [x_1, ..., x_n]^T$, and neurons located on a discrete lattice. The correspondence is obtained by a competitive learning algorithm consisting of a sequence of training steps that iteratively modifies the weight vector $\underline{m}_k \in \mathbb{R}^n$, $\underline{m}_k = (m_1^k, ..., m_n^k)$, of the neurons, where k is the location of the prototype in the lattice.

When a new signal $\underline{x}$ arrives every neuron competes to represent it. The best matching unit (bmu) is the neuron that wins the competition and with its neighbors on the lattice they are allowed to learn the signal. Neighboring neurons will gradually specialize to represent similar inputs, and the representations will become ordered on the map lattice.

The best matching unit is the reference vector c that is nearest to the input and is obtained by some metrics, $\|\underline{x} - \underline{m}_c\| = \min_i\{\|\underline{x} - \underline{m}_i\|\}$. In general, the Euclidean distance is used,

$$\|\underline{x} - \underline{m}_i\|_E = \sqrt{(\underline{x} - \underline{m}_i)^T(\underline{x} - \underline{m}_i)} = \sqrt{\sum_{j=1}^{n}(x_j - m_i^j)^2} \tag{1}$$

The winning unit and its neighbors adapt to represent the input by modifying their reference vectors towards the current input. The amount the units learn

will be governed by a neighborhood kernel $h_c(j,t)$, which is a decreasing function of the distance between the unit j and the bmu c on the map lattice at time t. The kernel is usually given by a Gaussian function:

$$h_c(j,t) = \alpha(t) \exp\left(\frac{-\left\|\underline{r}_j - \underline{r}_c\right\|^2}{2\sigma(t)^2}\right) \tag{2}$$

where $\underline{r}_j$ and $\underline{r}_c$ denote the coordinates of the neurons c and i in the lattice, $\alpha(t)$ is the learning rate parameter and $\sigma(t)$ is the neighborhood range. In practice the neighborhood kernel is chosen to be wide in the beginning of the learning process to guarantee global ordering of the map, and both its width and height decrease slowly during learning.

The learning parameter function $\alpha(t)$ is a monotonically decreasing function with respect to time, for example this function could be linear $\alpha(t) = \alpha_0 + (\alpha_f - \alpha_0)t/t_\alpha$ or exponential $\alpha(t) = \alpha_0(\alpha_f/\alpha_0)^{t/t_\alpha}$, where α_0 is the initial learning rate (< 1.0), α_f is the final rate (≈ 0.01) and t_α is the maximum number of iteration steps to arrive α_f. The final result is not greatly affected by the selection of this function [10]

During the learning process at time t the reference vectors are changed iteratively according to the following adaptation rule,

$$\underline{m}_j(t+1) = \underline{m}_j(t) + h_c(j,t)[\underline{x}(t) - \underline{m}_j(t)] \qquad j = 1..M \tag{3}$$

where M is the number of prototypes that must be adjusted. If we consider the following neighborhood:

$$h_c^*(j,t) = \frac{h_c(j,t)}{\alpha(t)} = \exp\left(-\frac{\|\underline{r}_c - \underline{r}_j\|^2}{2\sigma^2(t)}\right) \tag{4}$$

with a discrete data set and a fixed neighborhood kernel $h_c^*(j,t)$, the quantization error or the distortion measure that is stochastically minimized by the SOM [9], is

$$E = \sum_{i=1}^{N}\sum_{j=1}^{M} h_c(j,t)^* \|\underline{x}_i - \underline{m}_j\|^2 \tag{5}$$

where N is the number or training samples, and M is the number of map units. Some properties of the SOM can be found in [3].

Besides the classical SOM, there exist some variants to this algorithm, but we will not treat them here. For example, we can mention the K-means, Learning Vector Quantization and Neural Gas (see [7]).

3 Robust Self-organizing Map (RSOM)

Most data mining applications involve data that is contaminated by outliers. The identification of outliers can lead to the discovery of truly unexpected knowledge

in areas such as electronic commerce exceptions, bankruptcy, credit card fraud. One approach to identifying outliers is to assume that the outliers have a different distribution with respect to the remaining observations.

As we mention before, real data are not free of outlying observations and special care should be taken in the learning process to preserve the most important topological and metric relationships of the primary data items. In such cases it would be desirable that the outliers would not affect the result of the analysis. Each outlier affects only one map unit and its neighborhood.

First we will show that the learning algorithm given by equation (3) is not robust in the sense of Hampel criterion [4] when an outliying observation is presented. Suppose that an observation $\underline{x}$ is very distant from the majority of the data and therefore from the map. The distance from the best matching unit to the outlier has big magnitude and the learning step of this unit and its neighbors neurons moves the map towards the outlier and apart from the remaining observations. To measure this impact we used the supremum of the learning step over the whole input space

$$\sup_{\underline{x}\in\chi} \left(h_c^*(j)[\underline{x} - \underline{m}_j]\right) = \infty \tag{6}$$

i.e., the learning step is not bounded and indeed not B-robust [4], and the parameter estimation process is badly affected. To overcome this problem we propose to diminish the influence of the outliers by introducing a robust $\psi(\cdot)$ function, $\psi : \chi \times \mathcal{M} \to \mathbb{R}^n$, $\underline{m}_i \in \mathcal{M} \subseteq \mathbb{R}^n$, in the update rule as follows:

$$\underline{m}_i(t+1) = \underline{m}_i(t) + h_c(i,t)\psi\left(\frac{\underline{x}(t) - \underline{m}_i(t)}{s_i(t)}\right) \tag{7}$$

where $s_i(t)$ is a robust estimation of the variance of the data modelled by the neuron i. To estimate $s_i(t)$ we use a variant of the MEDA function given by:

$$s_i(t) = 1.483\ median\left\{|h_c^*(i,t)[\underline{x} - \underline{m}_i] - median(h_c^*(i,t)[\underline{x} - \underline{m}_i]|)\right\} \tag{8}$$

For example the Huber function could be used, which is given by $\psi_H(\underline{x}, \underline{m}_i) = sgn(\underline{r}_i)min\{|\underline{r}_i|, \delta\}$, $\delta \in \mathbb{R}^+, \underline{r}_i = (\underline{x} - \underline{m}_i)/s_i$

The quantization error that the robust learning algorithm (7) stochastically minimized is given by the following expression:

$$E = \sum_{i=1}^{N}\sum_{j=1}^{M} h_c^*(j,t)\rho\left(\frac{\underline{x}_i - \underline{m}_j}{s_j}\right) \tag{9}$$

where $\rho : \chi \times \mathcal{M} \to \mathbb{R}$ is a convex, symmetric with derivative $\psi(\underline{x}, \underline{m}) = \frac{\partial\rho(\underline{x},\underline{m})}{\partial\underline{m}}$. The conditions that the function $\rho(\underline{x}, \underline{m})$ and $\psi(\underline{x}, \underline{m})$ must fulfill can be found in [5].

4 Simulation Results

4.1 Experiment #1: Computer Generated Data

In order to see how the RSOM algorithm performs under a synthetic situation, the process was affected by an outlier generating process. Two clusters of spherical Gaussian distribution in 2 dimensions were constructed. A total of 500 training samples were drawn, where the expected size of each cluster was 250. In addition additive outliers were introduced.

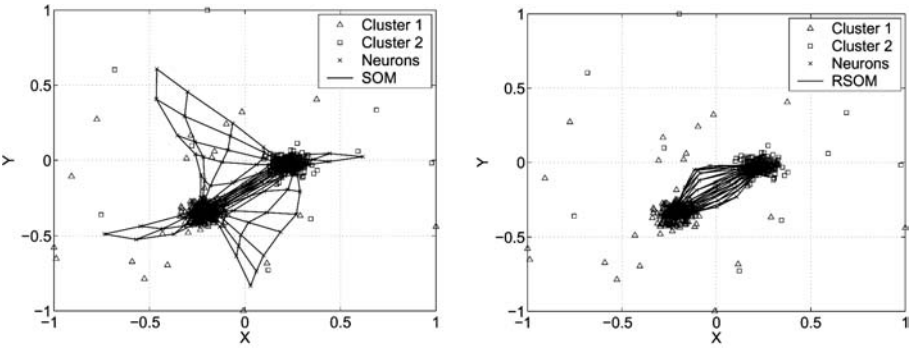

Fig. 1. Synthetic data results: Computer generated data with 5% outliers. (Left) Classical SOM modelling the data, the map is affected by the outliers as can be noted by the wings created in the map. (Right) Robust SOM modelling the data, the map is less affected and does not open towards the outliers.

Let $\chi_N = \{(x_1^i, x_2^i)\}_{i=1..N}$ be the independent sampled data from the gaussian distribution $\underline{X} = (X_1, X_2) \sim \mathcal{N}(\mu_k, \Sigma_k)$, $k = 1, 2$, where μ_k and Σ_k are the mean and the covariance matrix of the cluster k, and $\mathcal{N}(\mu_k, \Sigma_k)$ a two-dimensional gaussian distribution.

The observational process $\underline{z} = (z_1, z_2)$ is obtained by adding additive outliers: $\underline{Z} = \underline{X} + V\ U$, where V is zero-one process with $P(V \neq 0) = \gamma$, $0 < \gamma \ll 1$ and U has distribution $\mathcal{N}(\underline{0}, \Sigma_U)$ with $|\Sigma_U| \gg |\Sigma_k|, k = 1, 2$.

For the simulations we consider the following values for the data generation: $\mu_1 = [6, 4]^T$, $\Sigma_1 = 0.9 * I_2$, $\mu_2 = [-2, -3]^T$, $\Sigma_2 = 0.9 * I_2$, $\Sigma_U = 9 * I_2$ where I_2 is the identity matrix of size 2. The generating process was affected with $\gamma = 0\%, 1\%, 5\%, 10\%$ and 20% of outliers. To model this data we construct a SOM lattice with sizes 5×5 and 9×9.

In figure 1 synthetic data, the SOM model and the RSOM model are shown from left to right. The size of the map showed in the figure is 15×15. The Classical SOM is affected by outliers as can be noted by the wings created in the map, nevertheless the Robust SOM is less affected because it does not open towards the outliers. In table 1 the simulation results are shown.

In figure 2 a comparative study is shown, where in the left side the percentage of outliers in the data was fixed to 10%, and the graph of the Error v/s the

Table 1. Summary results showing the performance of the classical and robust learning methods with several sizes using the synthetic datasets. The column *Algorithm* is the type of learning algorithm, *Dim* gives the size of the map, *Neurons* gives the number of prototypes used, column $E1$ and $E2$ are the quantization error (9) of the test set consisting in 250 samples, $E1$ considers the data with outliers, $E2$ does not. Finally, columns $E3$ and $E4$ are the percentage of misclassification using the test set by considering and not considering the presence of outliers respectively.

Algorithm	Dim.	**Neurons**	% **outliers**	**E1**	**E2**	**E3%**	**E4%**
SOM	5X5	25	0	36.58	36.58	0.00	0.00
RSOM	5X5	25	0	76.77	76.77	0.00	0.00
SOM	9X9	81	0	15.58	15.58	0.00	0.00
RSOM	9X9	81	0	22.75	22.75	0.02	0.02
SOM	5X5	25	1	13.09	8.29	0.01	0.01
RSOM	5X5	25	1	22.37	16.76	0.00	0.00
SOM	9X9	81	1	8.04	4.13	0.00	0.00
RSOM	9X9	81	1	10.32	5.22	0.00	0.00
SOM	5X5	25	5	10.58	6.19	0.02	0.00
RSOM	5X5	25	5	16.05	10.32	0.02	0.00
SOM	9X9	81	5	7.28	4.33	0.02	0.01
RSOM	9X9	81	5	10.17	4.30	0.04	0.02
SOM	5X5	25	10	12.56	7.46	0.03	0.00
RSOM	5X5	25	10	15.59	9.55	0.03	0.00
SOM	9X9	81	10	9.16	9.16	0.07	0.02
RSOM	9X9	81	10	11.72	5.97	0.05	0.00
SOM	5X5	25	20	14.05	7.03	0.08	0.01
RSOM	5X5	25	20	19.28	7.20	0.05	0.00
SOM	9X9	81	20	7.66	10.20	0.10	0.02
RSOM	9X9	81	20	16.30	4.56	0.09	0.02

number of neurons for the SOM and RSOM are shown. In the right side the number of neurons was fixed to 81 and the graph of the Error v/s the percentage of outliers for the SOM and RSOM are shown and the evaluation of the test error by considering the outliers ($E1$) and without considering them ($E2$). As can be noted in the figure, the RSOM evaluated without outliers outperforms the other methods with increasing number of neurons or percentage of outliers. The SOM with bigger number of neurons tends to approximate the outliers and a poor performance is obtained if they are not considered. The quantization error $E1$ of the RSOM is worst than in the other cases with an increasing percentage of outliers implying that this is not a good performance measure when the data has outliers. When there are no outliers the classical method obtained better performance, but most real data are contaminated with outliers.

4.2 Experiment #2: Real Datasets

The second application consists of a real dataset known as the *Wisconsin Breast Cancer Database* obtained from the UCI Machine Learning repository [2] and

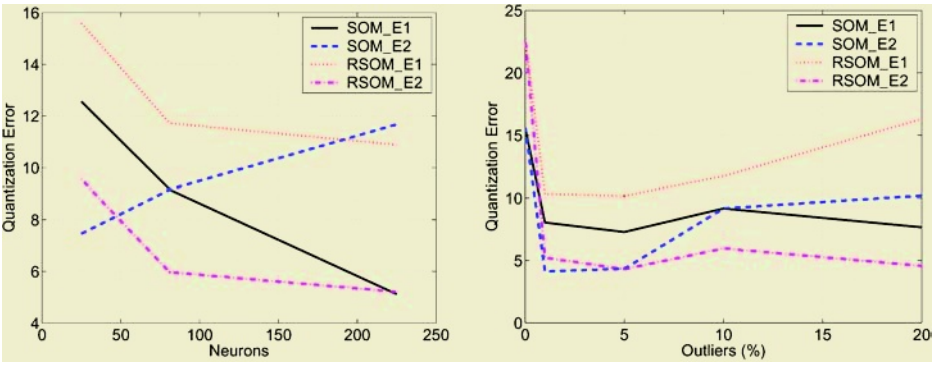

Fig. 2. Comparative Graph: (Left) The percentage of outliers in the data was fixed to 10%, and the graph of the Error v/s the number of neurons for the SOM and RSOM are shown. (Right) The number of neurons was fixed to 81 and the graph of the Error v/s the percentage of outliers for the SOM and RSOM are shown.

was collected by Dr. Wolberg, N. Street and O. Mangasarian at the University of Wisconsin [8]. The samples consist of visually assessed nuclear features of fine needle aspirates (FNAs) taken from patients' breasts. Each sample describes characteristics of the cell nuclei present in the image. It consist in 569 instances, with 30 real-valued input features and a diagnosis (M = malignant, B = benign) for each patient. Malignancy is determined by taking a sample tissue from the patient's breast and performing a biopsy on it. A benign diagnosis is confirmed either by biopsy or by periodic examination, depending on the patient's choice.

To model this data we construct a SOM lattice with sizes 3×3, 5×5, 7×7 and 9×9. In table 2 the performance of the classical and robust learning methods with several sizes are shown. The $E1$ column gives the quantization error and the $E3$ column the percentage of misclassification. The RSOM has worst performance in the quantization error $E1$, because, as shown in the previous example, the measure used considers only the global behavior. For these reason the classical SOM tends to approximate the outliers, and the quantization error is better than the RSOM. But, if the misclassification error is observed (column $E3$), the RSOM shows better results than the SOM.

5 Concluding Remarks

In this paper we introduce a Robust Self Organizing Map (RSOM) for modelling data that were affected by outliers. We apply a robust learning algorithm to the classical SOM to diminish the influence of the outlier in the learning process. We demonstrate that the classical update rule used to learn the data is not robust when there are samples that are very different (far) from the majority.

The performance of our algorithm shows better results in the simulation study in both the synthetic and real data sets. In the synthetic data set we study several degree of contamination and different networks sizes and we made

Table 2. Summary results showing the performance of the classical and robust learning methods with several sizes using the Wisconsin Breast Cancer Database.

Algorithm	Dim.	**Neurons**	Quantization Error (**E1**)	Misclassification Error (**E3** %)
SOM	3X3	9	565.14	0.0710
RSOM	3X3	9	559.74	0.1183
SOM	5X5	25	473.19	0.0888
RSOM	5X5	25	475.54	0.0296
SOM	7X7	49	417.59	0.1006
RSOM	7X7	49	448.50	0.0769
SOM	9X9	81	389.14	0.1420
RSOM	9X9	81	421.12	0.0947

a comparative analysis showing that the RSOM outperforms the SOM. In the real case, we investigate a benchmark named Wisconsin Breast Cancer Database that were studied by several researches. The RSOM shows better topology representation than the SOM obtaining better classification performance of the patients.

We also present our concern about the need of an error measure that considers the local behavior, because the quantization error given in equation (9) is global, and does not show the quality of the topology representation. Further studies are needed in order to analyze the convergence, the ordering properties together with the stationary states, metastability and convergence rate of the RSOM.

References

1. H. Allende, C. Moraga, and R. Salas, *Robust estimator for the learning process in neural networks applied in time series*, ICANN. LNCS **2415** (2002), 1080–1086.
2. C.L. Blake and C.J. Merz, *UCI repository of machine learning databases*, 1998.
3. E. Erwin, K. Obermayer, and K. Schulten, *Self-organizing maps: ordering, convergence properties and energy functions*, Biological Cybernetics **67** (1992), 47–55.
4. F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel, *Robust statistics*, Wiley Series in Probability and Mathematical Statistics, 1986.
5. Peter J. Huber, *Robust statistics*, Wiley Series in probability and mathematical statistics, 1981.
6. T. Kohonen, *The self-organizing map*, Proceedings of the IEEE, vol. 78, 1990, pp. 1464–1480.
7. ———, *Self-Organizing Maps*, vol. 30, Springer Verlag, 2001.
8. O. Mangasarian, W. Street, and W. Wolberg, *Breast cancer diagnosis and prognosis via linear programming*, Operations Research **43** (1995), no. 4, 570–577.
9. H. Ritter and K. Schulten, *Kohonen's self organizing maps: Exploring their computational capabilities*, IEEE ICNN 88 **I** (1988), 109–116.
10. M. Su and H. Chang, *Fast self-organizing feature map algorithm*, IEEE Trans. on Neural Networks **11** (2000), no. 3, 721–733.