An Extension of Self-Organizing Maps to Categorical Data

Ning Chen[1] and Nuno C. Marques[2]

[1] Institute of Mechanics, Chinese Academy of Sciences,
P. R. China
ningchen74@yahoo.com
[2] CENTRIA/Departamento de Informtica,
Faculdade de Ciencias e Tecnologia, Universidade Nova de Lisboa,
Quinta da Torre,
2829-516 Caparica,
Portugal
nmm@di.fct.unl.pt

Abstract. Self-organizing maps (SOM) have been recognized as a powerful tool in data exploratoration, especially for the tasks of clustering on high dimensional data. However, clustering on categorical data is still a challenge for SOM. This paper aims to extend standard SOM to handle feature values of categorical type. A batch SOM algorithm (NCSOM) is presented concerning the dissimilarity measure and update method of map evolution for both numeric and categorical features simultaneously.

1 Introduction

Clustering is an unsupervised process to partition a set of data into homogeneous clusters. Without the supervision of classes, data segmentation in clustering is performed based on intrinsic similarity of data.

Data can be described by categorical features and numeric features [1]. Nominal features, e.g. post code, gender, transportation mode, residence choice, are typically categorical taking on values from a limited and predetermined set of categories without natural ordering. Another type of categorical feature is ordinal, e.g. education level, social status, which has particular order but unknown distance. Numeric features have numerical distance between values. Numeric features can be further categorized to discrete type with relatively few values, e.g. age, number of cars, and continuous type with a large number of values, e.g. price, salary, temperature.

Self-organizing maps (SOMs) have broad applications in pattern recognition, engineering system, medical diagnosis and image segmentation [15]. The appeal of SOM as a model exploration method in clustering is its unique advantage on data visualization and summarization. From the visualizations, the models of phenomena could be generalized and the patterns could be recognized interactively.

Generally, standard SOMs are applied to feature values of numeric type. Usually, an Euclidean function is used to calculate the distances between input vectors and reference vectors. During the learning, the update of reference vectors is performed by

incremental or arithmetic operations. Unfortunately, these calculations are not practical on categorical values. Although categorical data has been discussed in some clustering algorithms (please see [4], [5] or [6]), it is not directly addressed in SOMs due to the limitation of learning laws. A traditional approach is to translate categories to numeric numbers in data preprocess and then perform standard SOMs on the transformed data [19]. Despite its feasibility on ordinal features by converting the categories into integers and preserving the natural order, an extra order is posed on nominal values. Also, this approach is not adapted to binary data as reported in [11]. In [12], an overview is made on several methods to encode categorical data for SOM, and their implications are analyzed in terms of the influence on the calculus of the best-matching unit(BMU). In this paper we work on the same direction, however the categorical mapping of our method is now done directly inside the SOM.

In order to operate categorical data, two issues should be considered: the dissimilarity measure of categorical features and the update method of map neurons. In this paper, a batch SOM algorithm is proposed based on new distance measurement and update rules in order to extend the usage of standard SOMs to categorical data. Different from the prior work [12] which mainly talks about the usage of binary-based similarity measures in SOMs, the proposed work focuses on the update method of neurons for both numeric and categorical features simultaneously.

The remaining of this paper is organized as follows. Sect. 2 presents a NCSOM learning algorithm for numeric and categorical data. Some experiments and results are shown in Sect. 3. Lastly, the contributions and future improvements are given in Sect. 4.

2 NCSOM: a batch SOM algorithm for numeric and categorical data

Self-organizing maps (SOMs) are artificial neural networks (ANN) used to visualize and interpret high-dimensional data in a low-dimensional space. SOMs are able to reduce the amount of data and simultaneously project data nonlinearly onto a lower dimensional array. The neurons (units) are organized on a regular grid of usually one or two dimensions. Each neuron is associated with a reference vector, reflecting the strength of association to input vectors. The topological relation of neurons is described by a neighborhood kernel function. The reference vectors are initialized at the beginning and updated iteratively in the training process. As a result, the neurons become topologically ordered on the grid, where neurons close to each other in the grid space have similar features in the input space.

Batch SOM algorithms [9] update the reference vectors at the end of each iteration of whole data set. In each epoch, the data vectors are input one by one and listed under the BMUs. Then the reference vectors are calculated as the weighted mean of input vectors that are similar either to themselves or to their topological neighbors. Batch SOMs are order-insensitive, facilitate the development of parallel processing, and eliminate the influence of learning rate as a coefficient [2]. In this section, a batch SOM algorithm for numeric and categorical data will be studied based on the distance measure introduced in [6].

2.1 Dissimilarity measure

Data projection is based on the distance or dissimilarity between input vectors and reference vectors. Due to the unknown distance between values of categorical features, a simple mismatch measurement [7] is used here. The dissimilarity between two values of single categorical feature is zero if and only if they belong to the same category, otherwise the dissimilarity is one. For a data set with mixed type features, the dissimilarity of two instances is measured on numeric and categorical features separately. Let n be the number of input vectors, m the number of map units, and d the number of features. Suppose the input vectors consist of p numeric features and $d-p$ categorical features, $\{\alpha_k^1, \alpha_k^2 \ldots \alpha_k^{n_k}\}$ is the set of variant values of the k^{th} categorical feature. We denote $x_i = [x_{i1}, \ldots, x_{id}]$ as the i^{th} input vector and $m_j = [m_{j1}, \ldots, m_{jd}]$ as the reference vector of the j^{th} neuron. The dissimilarity between x_i and m_j is defined as the combination of squared Euclidean distance on numeric features and number of mismatches on categorical features [6]. To ensure all features have equal influence on distance, numeric features are usually normalized before distance calculation.

$$d(x_i, m_j) = \sum_{l=1}^{p} (x_{il}, m_{jl})^2 + \sum_{l=p+1}^{d} \delta(x_{il}, m_{jl}), \quad \delta(x_{il}, m_{jl}) = \begin{cases} 0 & x_{il} = m_{jl} \\ 1 & x_{il} \neq m_{jl} \end{cases} \tag{1}$$

2.2 Update rules

In the training process, an input vector is mapped to the best-matching unit, namely, the winner with the closest reference vector. Then a Voronoi set can be generated for each unit: $V_i = \{x_k \mid d(x_k, m_i) \leq d(x_k, m_j), 1 \leq k \leq n, 1 \leq j \leq m, i \neq j\}$. As a result, the input space is divided into a number of Voronoi sets: $\{V_i, 1 \leq i \leq m\}$. At the end of each epoch, the map is updated by different strategies depending on the type of features.

The update rule of reference vectors on numeric features is same to that of standard batch SOMs [9]. Assume $m_{pk}(t)$ is the value of the p^{th} unit on the k^{th} numeric feature at time t. The incremental value on m_{pk} is $\Delta m_{pk}(t) = \sum_{i=1}^{n} h_{c_i p}(x_{ik} - m_{pk}(t))$, where $c_i = arg\ \min_j d(x_i, m_j(t))$ is the BMU of x_i and $h_{c_i j}$ is a non-increasing neighborhood function centered at the best-matching unit. At time $t+1$,$m_{pk}(t+1) = m_{pk}(t) + \frac{1}{\sum_{i=1}^{n} h_{c_i p}} \Delta m_{pk}(t)$. If $\sum_{i=1}^{n} h_{c_i p} = 0$ for some p, that means m_p is neither the winner of any input vector nor the neighbor of other winners, then $m_{pk}(t+1) = m_{pk}(t)$.

Update rule on numeric features:

$$m_{pk}(t+1) = \frac{\sum_{i=1}^{n} h_{c_i p} x_{ik}}{\sum_{i=1}^{n} h_{c_i p}} \tag{2}$$

Due to the unknown distance between categorical values, they can not be updated incrementally as numeric values. Intuitively, the category occurring most frequently in the Voronoi sets of a neuron and its neighbors should be chosen as the new value for the

next epoch. To determine the new category of a neuron, the frequency of each category is calculated as the average weight of all input vectors having the same value on this feature. For this purpose, a set of counters is used to store the frequencies of variant values for each categorical feature.

$$F(\alpha_k^r, m_{pk}(t)) = \frac{\sum_{i=1}^{n}(h_{c_i p} \mid x_{ik} = \alpha_k^r)}{\sum_{i=1}^{n} h_{c_i p}}, r = 1, 2, \ldots, n_k \tag{3}$$

For nominal features, the best category c, i.e., the value having maximal frequency, is accepted at once if its frequency is more than the total frequency of other categories or accepted randomly with a threshold θ. (The smaller value of θ implies the higher possibility to accept c. If θ = 0, c is always accepted. In the experiments of section 3, θ is set as 50%.) This random acceptance strategy works profitable to avoid local minima of optimization.

Update rule on nominal features:

$$m_{pk}(t+1) = \begin{cases} c & \text{if } F(c, m_{pk}(t)) > \sum_{r=1, r\neq c}^{n_k} F(\alpha_k^r, m_{pk}(t)) \\ c & elseif\ random(0,1) > \theta \\ m_{pk}(t) & \text{otherwise} \end{cases} \tag{4}$$

where

$$c = arg\ \max_r F(\alpha_k^r, m_{pk}(t))$$

For ordinal features, the category closest to the weighted sum of frequencies on all possible categories is chosen as the new value concerning about the natural ordering of values.

Update rule on ordinal features:

$$m_{pk}(t+1) = round(\sum_{r=1}^{n_k} r * F(\alpha_k^r, m_{pk}(t))) \tag{5}$$

Some neighborhood kernel functions are used for describing the topological structure of SOMs. The bubble function, $h_{r_i r_j} = \begin{cases} 1 \text{ if } \parallel r_i - r_j \parallel^2 \leq \delta(t) \\ 0 \text{ otherwise} \end{cases}$, defines a neighbor set within a neighborhood region of radius $\delta(t)$, where $\delta(t)$ monotonically decreases with regression steps in order to stabilize the effect of the input vectors on the maps. In this case, the frequency could be determined by the percent of the category occurring in the union of Voronoi sets. Gaussian function $h_{r_i r_j} = exp\left(-\frac{\|r_i - r_j\|^2}{2\delta^2(t)}\right)$ is another popular neighborhood function. Compared to bubble function, it is more effective but computationally heavier [9].

2.3 Algorithm description

In summary, NCSOM algorithm can be described as follows.

Step 1: Initialize the reference vectors of map units.

Step 2: Input the instances one at a time. Calculate the distances between the input vector and reference vectors using Equation(1). Project the input to the best-matching unit.

Step 3: Update the reference vectors on each feature separately at the end of each epoch over the training process. The values on numeric features are the average values of all input vectors weighted by the neighborhood function values according to Equation(2). The values on nominal features and ordinal features are updated according to Equation(4) and Equation(5) respectively. Replace old reference vectors with new ones.

Step 4: Repeat from Step 2 a few times until the solution can be regarded as steady.

3 Experiments and Discussion

The NCSOM algorithm has been implemented in an adapted version of SOM software [10], [16]. Also, the initial center selection, partitive clustering algorithms, and cluster assignment are developed. The experiments are performed on a few data sets in a machine with 256M memory and intel celeron 1.03 GHz processor running windows XP professional operating system.

3.1 Experimental results

Empirical studies have been conducted on three pure categorical data sets: soybean, mushroom, tic-tac-toe and two mixed numeric/categorical data sets: credit approval, cleveland heart disease in UCI Machine Learning Repository [13]. All features are made to contribute to distance calculation equally, by normalizing the numeric features to unity range. Figure 1 represents the results of NCSOM on five data sets. For easy visualization, these data are shown in a 2-dimensional space through principal component projection (PCA), that is a linear transformation of high dimensional data to a low dimensional space [3].

The first well-known soybean data set consists of 47 instances with 35 nominal features. The instances are divided into four classes of 10,10,10,17 members respectively. This data set is used to classify soybean plants according to the diseases. In the bottom right of Figure 1(a), soybean data is visualized in a 2-dimensional space spanned by the eigenvectors of two maximum eigenvalues of data through PCA. Each dot represents one instance, showing in different color according to class labels. The neurons of trained map are also displayed in the same space, and adjacent neurons are connected by lines presenting the neighborhood relations between units. As it was shown, the instances of 'D0' and 'D1' form two clusters individually. It seems that the other cluster is composed of instances in 'D2' and 'D3'. On the top left graph, neurons are covered by hexagons of size proportional to hit values (the absolute number of instance histogram matching to map neurons) and marked by the hit values. Intuitively, neurons in clusters get more hits than those between clusters [21]. In fact, the four clusters are separated from each other by the zero-hit neurons. Each Voronoi set forms a subcluster of data.

[3] All components are handled as numeric in data transformation.

By looking at the top right graph, the dominating classes of subclusters are known immediately. If the members of a subcluster belong to more than one class, we can detect the constitution of subclusters from the hit values of diverse classes. In the bottom left graph, a pie chart is displayed in the place of each neuron with nonzero hit, showing the percent of classes contained in the corresponding subcluster. It can be observed that NCSOM performs on soybean data perfectly, generating a number of subclusters of individual class.

The second data set under consideration is mushroom data. Although it has 8124 instances, only 500 random samples are selected as experimental data. The goal is to label the instances as 'edible' or 'poisonous' according to 21 nominally valued features. Figure 1(c) visualizes the results on mushroom data, with labels of map units on the left and pie chart of hit values on the right. Although mushroom does not present clear cluster structure (on the visualization in Figure 1(b), each of the two clusters consists of mixed instances of two classes), it still reaches exceptionally high accuracy on SOM clustering. It can be stated that mushroom is composed of a number of small and compact subclusters of instances almost coming from individual class.

Tic-tac-toe is the third data set of interest. It concerns the board configuration of games with 958 instances and 2 classes. It is described by 9 nominal features, each corresponding to one tic-tac-toe square. Also, a sample of 500 instances is randomly generated for analysis. Figure 1(d) shows the labels and hits for tic-tac-toe. As reported by other clustering algorithms [14], NCSOM also performs poorly on this data. We speculate that the poor performance could be explained by the weak cluster models in the data.

Next, we turn to mixed type data sets. Credit approval data set concerns credit card applications, consisting of 9 nominal-valued and 6 numeric-valued features. The 690 samples are classified into two classes with 307 and 483 respectively. It contains 67 missing values on both numeric and categorical features, which are ignored in distance calculation and neuron update. As given in Figure 1(e), the instances of class '+' are projected mainly to the neurons on top of map and those of class '-' to neurons on the bottom. The cluster structure can be detected from the histogram visualization. The neurons labeled by single class usually locate in the inner of clusters, while neurons labeled by multiple classes on the cluster boundary. For this data, it was observed that the neurons of pure class are surrounded by those of mixed classes.

Finally, heart data set contains the records of heart disease diagnosis for 303 patients. The data is described by 5 numeric features: age, cholesterol, max heart rate, resting blood pressure, ST depression relative to rest, and 8 categorical features: sex (male, female), chest pain type (typical angina, atypical angina, non-angina pain, asymptomatic), fasting blood sugar (< 120 or ≥ 120), resting electrocardiographic results (normal, abnormality, hypertrophy), exercise induced angina (yes or no), slope of peak exercise ST segment (up, flat, down), number of vessels colored (0,1,2,3), thalium scan (normal, fixed, reversable). Due to the natural ordering of values, these features are handled as ordinal except sex and exercise induced angina. The instances are classified to 2 classes as 'healthy' or 'sick'. The latter class can be further divided into 4 subspecies (S1, S2, S3, S4). Figure 1(f) gives the composition of subclusters on the same map labeled by 2

classes and 5 classes respectively. In comparison with the former, the pie of 'sick' class in most neurons is divided into several parts of diverse diseases in 5-classes case.

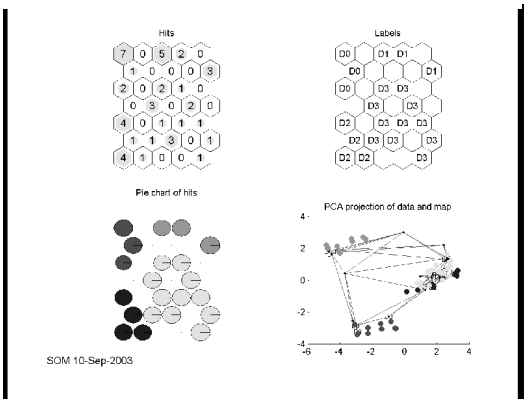
3.2 Effectiveness studies

To test the effectiveness on categorical data, NCSOM is compared with a standard batch SOM algorithm. In the latter, the categorical values are transformed to continuous integers in random order for nominal features or in nature order for ordinal features.

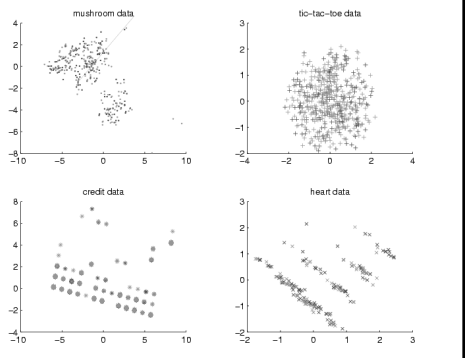
Evaluation is a process to evaluate the quality of clustering algorithms. The quality of SOMs is usually measured based on quantization precision and topology preservation [18]. The former is typically estimated by the squared quantization error, namely, average distance between input vectors and corresponding best-matching units. The smaller quantization error is, the better the trained map matches to data. The latter is estimated by topology error, namely the number of inputs to which the best-matching unit and next-best-matching unit are not adjacent on the map grid. Distortion integrates quantization and topology measures, defined as the weighted average of distances between input samples and map units.

When the true clusters are known, confusion matrix and rand index are appropriate and commonly used measures for clustering evaluation. Confusion matrix detects how closely the composition of obtained clusters matches to true partition structure. Based on pairwise comparison, rand index [3] is defined as the percent of pairs of instances that locate in either the same or different clusters in both true and obtained clustering. The rand index reaches one if the obtained clusters and true clusters match to each other perfectly. Both confusion matrix and rand index are appropriate for the one-class/one-cluster case [20]. Because the neurons are much more than real clusters, a set of subclusters are obtained as the result of SOM. In such cases, the purity of subclusters is important to final clusters (the instances of a subcluster always belong to the same cluster in future summarization), so SOM clustering can be evaluated by the percent of majority vote [17]. Each unit is identified as the dominating class label (major vote) of its Voronoi set, and instances having different classes are identified as errors. Finally, the purity of subclusters is calculated as the percentage of instances clustered correctly.

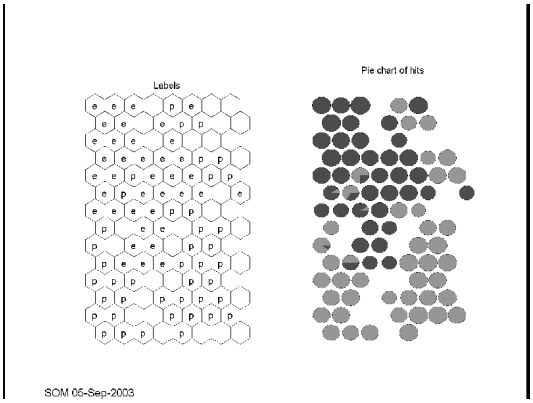
In this experiment, the full data set is divided into 10 folds and only 90% data are used for map training and labeling in each run. The quality of derived map is evaluated in terms of the purity of subclusters. For the sake of minimal initialization effect, we conduct 10 trials for each subset, starting from randomly initialized map and then learning through two phases. In rough training, the map is trained for a small number of epoches with large neighborhood radius. In fine-tuning training, the map is trained for a big number of epoches with small radius. The results of different data sets are summarized in Table 1. As expected, NCSOM performs better on all data sets than standard SOM treating categorical features as numeric. Typically, NCSOM reports more than 5% improvement on credit data, which confirms the effectiveness of our methodology on categorical data. For heart data, treating some features as ordinal produces better results than pure nominal features. Compared to the accuracy of two classes, the separation of 'sick' class into four subspecies results in significant decrease of accuracy.



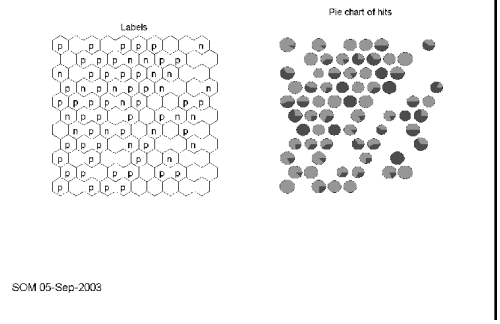
(a) SOM results of soybean



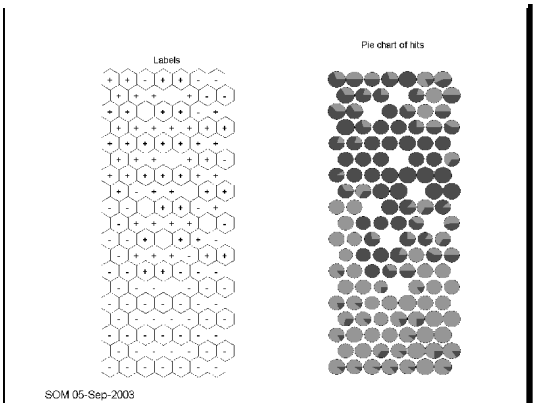
(b) PCA visualization of data



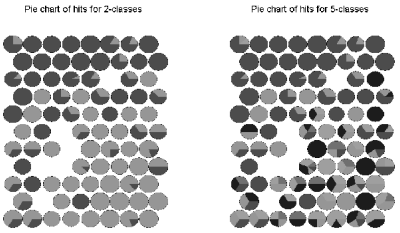
(c) SOM results of mushroom



(d) SOM results of tic-tac-toe



(e) SOM results of credit



(f) SOM results of heart

Fig. 1. Results of five data sets

data sets	#instance	#features	#classes	NCSOM	Standard SOM
soybean	47	35	4	0.9988	0.9770
mushroom	500	22	2	0.9648	0.9558
tic-tac-toe	500	9	2	0.7896	0.7732
credit	690	15	2	0.8529	0.7958
heart	303	13	2	0.8728	0.8659
heart	303	13	5	0.7152	0.7047

Table 1. Comparison of two approaches

SOMs can be used as classifiers after labeled with classified samples. To test the performance of NCSOM on classification tasks, experiments are conducted using the same arguments as COBWEB, a well-known concept clustering algorithm for categorical data described in [20]. In each trial, the map is trained on 90% of data, then neurons are labeled by the majority vote of projected instances. Afterwards, the evaluation is only performed on the remaining data by calculating the rand index between real labels and obtained labels. Table 2 shows the performance achieved by two algorithms, NCSOM and COBWEB (the results of COBWEB were reported in [20]). As we observed, NCSOM outperforms COBWEB on soybean and tic-tac-toe. NCSOM behaves somewhat worse than Cobweb on mushroom, possibly due to the random effect of subset generation. A small subset of only 50 instances fails to explicitly capture the characteristic of data distribution. It was reported that NCSOM gets statistically higher accuracy to 72.4% when a subset of 200 samples was used.

data	# instances	# attributes	NCSOM	COBWEB
soybean	47	35	0.946	0.849
tic-tac-toe	100	9	0.54	0.475
mushroom	50	22	0.619	0.667

Table 2. Comparison of NCSOM and COBWEB

4 Conclusions

SOMs have been widely used in data clustering as valuable tools due to the unique properties on data summarization and visualization. Normally, standard SOMs are applicable to numeric features through arithmetic operations on distance calculation and map evolution. In this paper, we present an approach to handle categorical data in batch SOM algorithms. The performance of proposed algorithms is demonstrated on some real data sets. In future work, we expect to deploy the proposed algorithm for data exploring on some real world problems which have been studied through previous and current funded research projects.

5 Acknowledgements

The authors would like to thank Dr. Robert Detrano of V.A. Medical center as the principal investigator for the collection of Cleveland clinic heart data.

References

1. Alan Agresti: Categorical data analysis. Wiley series in probability and mathematical statistics, John Wiley & Sons, New York (1990)
2. Qin Ding, Maria Canton, David Diaz, Qinghua Zou, Baojing Lu et al.: Data mining survey. `http://midas.cs.ndsu.nodak.edu/~ding/`
3. Arthur Flexer: On use of self-organizing maps for clustering and visualization. In J.M. Zytkow and J. Rauch (eds.), Principles of Data Mining and Knowledge Discovery, Proceedings of the 3rd European Conference (PKDD'99), Prague, Czech Republic, Lecture Notes in Artificial Intelligence 1704, Springer (1999) 80-88
4. Venkatesh Ganti, Johannes Gehrke, Raghu Ramakrishnan: CACTUS-clustering categorical data using summaries. Knowledge Discovery and Data Mining (1999) 73-83
5. Sudipto Guha, Rajeev Rastogi, Kyuseok Shim: ROCK: a robust clustering algorithm for categorical attributes. Information Systems **25(5)** (2000) 345-366
6. Zhexue Huang: Clustering large data sets with mixed numeric and categorical values. In Lu Hongjun, Motoda Hiroshi, Liu Huan (eds), Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery & Data Mining. Singapore, World Scientific (1997) 21-34
7. Zhexue Huang: Extensions to the k-means algorithms for clustering large data sets with categorical values. Data Mining and Knowledge Discovery **2** (1998) 283-304
8. Anil K. Jain, M. Narasimha Murty, Patrick J. Flynn: Data clustering: a review. ACM Computering Survey **31(3)** (1999) 264-323
9. Teuvo Kohonen: Self-organizing maps. Springer Verlag, Berlin, Second edition (1997)
10. Teuvo Kohonen, Jussi Hynninen, Jari Kangas, Jorma Laaksonen: SOM PAK: the Self-Organizing Map program package. Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science (1996)
11. F. Leisch, A. Weingessel et al.: Competitive learning for binary valued data. International Conference on Artifcial Neural Networks, Skoevde, Sweeden, Springer
12. Fernando Lourenco, Victor Lobo, Fernando Bacao: Binary-based similarity measures for categorical data and their application in self-organizing maps. JOCLAD 2004 - XI Jornadas de Classificacao e Anlise de Dados, April 1-3 , Lisbon (2004)
13. Catherine L. Blake, Chris J. Merz: UCI Repository of machine learning databases. University of California, Department of Information and Computer Science UCI Machine Learning Repository (1998)
14. Robert Munro: Classification and analysis in supervised mixture-modelling. University of Sydney, technical report **536**
15. Nuno Marques and Ning Chen.Border Detection on Remote Sensing Satellite Data Using Self-Organizing Maps. EPIA'03-11th Portuguese Conference on Artificial Intelligence, 4th International Workshop on Extraction of Knowledge from Databases (EKDB'03), Fernando Moura Pires, Salvador Abreu (eds.), Springer, Beja, Portugal (2003) 294-307
16. Laboratory of computer and information sciences & Neural networks research center, Helsinki University of Technology: SOM Toolbox 2.0
17. Luis Talavera, Javier Bejar: Integrating declarative knowledge in hierarchical clustering tasks. In Proceedings of the international symposium on intelligent data anlysis. Amsterdam, Netherlands, Springer-Verlag, 211-222

18. Juha Vesanto: Data mining techniques based on the self-organizing map. M.S. Thesis (1997)
19. Juha Vesanto, J. Himberg, E. Alhoniemi, J. Parhankangas: Self-organizing map in matlab: the SOM toolbox. In Proceedings of the Matlab DSP Conference, Espoo, Finland (1999) 35-40
20. Kiri Wagstaff, Claire Cardie: Clustering with instance-level constraints. In Proceedings of the 7th International Conference on Machine Learning (2000) 1103-1110
21. Xuegong Zhang, Yanda Li: Self-organizing map as a new method for clustering and data analysis. In Proceedings of International Joint Conference on Neural Networks (IJCNN) (1993) 2448-2451