A Graph Based Framework for Clustering and Characterization of SOM

Rakia Jaziri[1], Khalid Benabdeslem[2], and Haytham Elghazel[2]

[1] University of Paris13 - LIPN CNRS 7030
99 Av J.B. Clement, 93430 Villetaneuse, France
[2] University of Lyon1 - LIESP EA 4125,
43 Bd du 11 Novembre, 69622 Villeurbanne, France
rakia.jaziri@lipn.univ-paris13.fr, {kbenabde,elghazel}@univ-lyon1.fr

Abstract. In this paper, a new graph based framework for clustering characterization is proposed. In this context, Self Organizing Map (SOM) is one popular method for clustering and visualizing high dimensional data, which is generally succeeded by another clustering methods (partitional or hierarchical) for optimizing the final partition. Recently, we have developed a new SOM clustering method based on graph coloring called McSOM. In the current study, we propose to automatically characterize the classes obtained by this method. To this end, we propose a new approach combining a statistical test with a maximum spanning tree for local features selection in each class. Experiments will be given over several databases for validating our approach.

Keywords: SOM, Characterization, Graphs.

1 Introduction

Clustering is an important task in knowledge discovery systems. It aims to discover the best structure from unlabelled data. To this end, the principle conducts a process of organizing objects into homogeneous groups. The clustering methods can be grouped into five families including partitioning, hierarchical, density-based, grid-based or model-based [11]. In this paper, we are interested in models and particularly those using self-organizing map (SOM) [13]. This technique is a prominent tool for high-dimensional data analysis since it provides a substantial data reduction that can be used for visualizing and exploring properties of data. Generally, two important issues need to be clarified after clustering data by SOM: how to optimize the number of neurons and how to characterize the obtained classes. To deal with the first issue, several authors have investigated SOM clustering in sequential ways [17] or simultaneous ones [3]. However these approaches don't take into account the topological neighborhood relations offered by SOM. For that reason, we present a recent developed method based on coloring of graphs where more details can be found in [6]. This method is called McSOM for Minimal coloring of SOM. It represents an extension of the minimal graph coloring technique for clustering.

After clustering task, it is important to select the optimal feature subset which is relevant for characterizing each obtained class. This task is generally done by experts or by automatic characterizing methods. In this context, several important research topics in cluster analysis and feature weighing are discussed in [2] [7] [8] [9]. For such approaches, the search for an optimal subset of features is built into the clustering construction making these techniques specific of a given learning algorithm. In [5] the authors discover the relevant features using filter techniques by looking only at the intrinsic property of data. The disadvantage is that this method ignores the interaction with the clustering algorithm and that most proposed techniques are univariate thus ignoring feature dependencies.

In contrast of these weighting algorithms, we propose a graph based approach for unsupervised feature selection. For each obtained class, our approach attempts to find a "Pivot" features using a statistical test and maps them into a maximum spanning trees for selecting their related subset of features that characterize this class. The rest of the paper is organized as follow: in section 2 we describe the batch version of SOM model. In section 3, we briefly present McSOM, a recent developed approach for clustering of SOM. The section 4 will be devoted to the used statistical test and maximum spanning trees for feature selection. Finally, we provide experimental results on several databases and a conclusion on our proposed work.

2 Self-Organizing Map: SOM

SOM is used nowadays through numerous domains and has been successfully applied in numerous applications. It is a very popular tool used for visualizing high dimensional data spaces. SOM can be considered as doing vector quantization and/or clustering while preserving the spatial ordering of the input data rejected by implementing an ordering of the codebook vectors (also called prototype vectors, cluster centroids or reference vectors) in a one or two dimensional output space. The SOM consists of neurons organized on a regular low-dimensional grid, called the map. More formally, the map is described by a graph (V, E). V is a set of m interconnected neurons having a discret topology defined by E. For each pair of neurons (c, r) on the map, the distance $\delta(c, r)$ is defined as the shortest path between c and r on the graph. This distance imposes a neighborhood relation between neurons (Fig. 1). Each neuron c is represented by a p-dimensional reference vector $w_c = \{w_c^1, ..., w_c^p\}$ from $\mathcal{W}$ (the set of all map's neurons), where p is equal to the dimension of the input vectors. The number of neurons may vary from a few dozen to several thousand depending on the application.

The SOM training algorithm resembles *k-means* [15]. The important distinction is that in addition to the best matching reference vector, its neighbors on the map are updated. The end result is that neighbouring neurons on the grid correspond to neighbouring regions in the input space.

The SOM algorithm is proposed on two versions: Stochastic (on-line) or batch (off-line) versions. In this paper, we use the second one which is deterministic and fast. The batch version of SOM is an iterative algorithm in which the whole

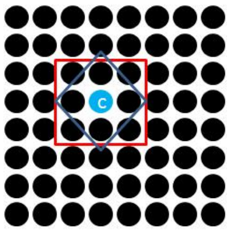

Fig. 1. Two dimensional topological map with 1-neighborhood of a neuron c. Rectangular (red) with 8 neighbors and diamond (blue) with 4 neighbors.

data set is presented to the map before any adjustments are made. In each training step, the data set is partitioned according to the Voronoi regions of the map reference vectors. More formally, we define an affectation function f from R^p (the input space) to C, that associates each element z_i of R^p to the neuron whose reference vector is "closest" to z_i (for the Euclidean distance). This function induces a partition $P = \{P_c; c = 1...m\}$ of the set of individuals where each part P_c is defined by: $P_c = \{z_i \in \Omega; f(z_i) = c\}$. This represents the **assignment step**.

Next, an **adaptation step** is performed when the algorithm updates the reference vectors by minimizing a cost function, noted $\mathcal{E}(f, \mathcal{W})$. This function has to take into account the inertia of the partition P, while insuring the topology preserving property. To achieve these two goals, it is necessary to generalize the inertia function of P by introducing the neighborhood notion attached to the map. In the case of individuals belonging to R^p, this minimization can be done in a straight way. Indeed new reference vectors are calculated as:

$$w_r^{t+1} = \frac{\sum_{i=1}^{n} h_{rc}(t) z_i}{\sum_{i=1}^{n} h_{rc}(t)} \tag{1}$$

where $c = arg\min_r \|z_i - w_r\|$, is the index of the best matching unit of the data sample z_i, $\|.\|$ is the distance mesure, typically the Euclidean distance, and t denotes the time. $h_{rc}(t)$ is the neighborhood kernel around the winner unit c. In practice, we often use $h_{rc} = e^{-\frac{\delta_{rc}}{2T^2}}$ where T represents the neighborhood raduis in the map. It is decreased from an initial value T_{max} to a final value T_{min}.

Consequently, h_{rc} is a non-increasing function of time and of the distance of unit r from the winner unit c. The new reference vector is a weighted average of the data samples, where the weight of each data sample is the neighborhood function value $h_{rc}(t)$ at its winner c.

3 Graph Coloring of SOM: McSOM

In this section, we briefly present our recently developed graph based approach for clustering SOM. Let be $\Omega = \{z_1, z_2, \ldots, z_n\}$ ($z_i \in \Re^p$) a data set clustered with SOM into m neurons of a topological map. If we consider this map as an undirected graph $G(V, E)$ where V is a set of vertices (neurons) and E is a set of edges (similarities between neurons), we can use the graph theory methods for clustering SOM neurons. The main idea is to use neighbourhood relations to constrain the construction of the *threshold graph* (indispensable for the coloring algorithm) and the possible vertex selections during the building of the minimal coloring of this graph.

The *minimal coloring based clustering approach* requires a *non complete edge-weighted graph* $G_{>\theta} = (V, E_{>\theta})$ to return a partition P_k of $\mathcal{W} = \{w_1, w_2, \ldots, w_m\}$ neurons (reference vectors) set. In order to incorporate the SOM neighbourhood relations into the clustering problem, our first modification concerns the construction of this non-complete graph that will be presented to the *Largest First* (*LF*) algorithm of *Welsh and Powell* [18]. This *non complete edge-weighted graph* is now given by $G_{>\theta,\alpha} = (V, E_{>\theta,\alpha})$, where $E_{>\theta,\alpha}$ is the edge set, where for each two vertices v_i and v_j in V (corresponding to (w_i, w_j) in the map), the edge $(v_i, v_j) \in E_{>\theta,\alpha}$ *if* $\mathcal{D}(w_i, w_j) > \theta$ *or* $\mathcal{SN}(w_i, w_j) > \alpha$ where $\mathcal{D}$ is the dissimilarity matrix, $\mathcal{SN}$ is the *SOM rectangular - neighbourhood order matrix*, θ is the *dissimilarity* threshold and α is the *SOM rectangular-neighbourhood order* threshold. This proposal offers the possibility to perform the *minimal coloring based clustering approach* multiple runs, each of them increasing the value of α.

The neurons to be clustered are now depicted by a non-complete edge-weighted graph $G_{>\theta,\alpha}$. Additional modifications of the *minimal coloring based clustering approach* are considered in order to improve the quality of clustering by incorporating the topological relations offered by SOM. The changes concern now the *LF* algorithm used for coloring the graph $G_{>\theta,\alpha}$. In fact, after sorting the vertices of $G_{>\theta,\alpha}$ by decreasing degree, the *LF algorithm* starts from the vertex of V which has the maximum degree Δ (the maximum number of edges from V). The algorithm colours this vertex using the color one. Then, it tries to color the remaining vertices (by respecting the decreasing order of their degree) according to the following principle: each vertex is placed in the first color class for which it has no neighbours. If no such color exists, then a new color is created for it. In the end of the algorithm, each neuron will be colored and neurons having the same color, belong to the same class. Theoretical and practical details can be found in [6].

4 Characterizing McSOM: G-Select

In the previous section, another way for clustering SOM using a graph coloring based approach is presented. Once clustering done, a data partition with an optimal number of classes is obtained. In this section, we introduce a new graph base approach, called G-Select (for Graph based feature selection), to find the local feature subsets that characterize each class.

4.1 Determining "Pivot" Features

The first step of G-Select aims to define the statistical test which allows to determine the "Pivot" features. A "Pivot" feature is the most relevant one in a class, i.e. it is the feature which contributes most in grouping observations in the associated class.

The statistical test for a given feature in a given class defines a test value ($\mathcal{T}$) which is simple but practically very important [14].

Let be a sample of size n, a class K found by McSOM with cardinality n_K ($n_K < n$). The test value of a feature j in K can be defined by:

$$\mathcal{T}_K(j) = \frac{\mu_{jK} - \mu_j}{\sqrt{\frac{n-n_K}{n-1} \times \frac{\sigma_j^2}{n_K}}} \tag{2}$$

Where u_j is the mean value of j in the global sample, its variance is σ_j^2 and its mean in the class is μ_{jK}.

$\mathcal{T}$ can be considered as a comparison test between means of different sets. It is classically used to measure dissimilarity between the overall distribution of a feature and that observed in the considered class. We note that if all features participate in the construction of the partition, as in our case, it is not appropriate to define a hypothetic threshold for selecting the most important ones according to $\mathcal{T}$ since this value quantifies the relevance of each feature individually and independently of the other ones [14]. Thus, $\mathcal{T}$ will be use just for ranking theses features and determining the best one which will serve as a "Pivot" in the construction of tree in the next section. Subsequently, a "Pivot" feature p^* of one class K is defined as follow:

$$p_K^* = arg \max_{1 \leq j \leq p} |\mathcal{T}_K(j)| \tag{3}$$

4.2 Maximum Spanning Tree for Characterization

In this section, we show how to automatically detect the set of features which have a strong multiple correlation and are directly linked to the "Pivot" feature. For that, we propose to perform a maximum spanning tree based method. This technique requires a matrix of weights between vertices (features in our case). We calculate, thus, a matrix of correlations for each class K, where a correlation between two features j and j' is defined in:

$$corr(j, j') = \frac{1}{\sigma_j \sigma_{j'}} \times \frac{1}{n_K} \sum_{k=1} (z_k^j - \mu_{jK})(z_k^{j'} - \mu_{j'K}) \tag{4}$$

For each K, we can define a weighted graph $G_K(D, CORR)$ is defined, where D is the set of all features (vertices) and $CORR$ is the set of edges valuated according to (4).

A maximum spanning tree $G_K^{'}$ is a connected and acyclic sub-graph of G_K, for which the sum of edge weights is maximum. This graph allows to detect the features which are directly linked to the "Pivot" feature without using any threshold. Fig. 2

On the left side of Fig. 2, G represents a complet graph where all features are weighted using the correlation values. On the right, we can see the maximum

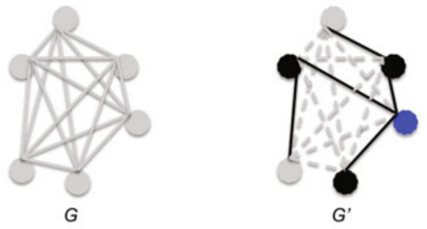

Fig. 2. G: Original Graph; G':Maximum spanning tree

spanning tree G' obtained from G where are showed: the pivot feature (blue vertex), the features linked to the pivot features (dark vertices) and the edges which contribute to the construction of the tree (solid lines).

For constructing $G^{'}_{K}$ of a class K, we use the optimized algorithm of Prim [4]:

Let be V and E two empty sets. First, we affect to V one feature from D. The goal is to find the edge (j, j') of $\overline{V} \times V$ having the maximum weight ($\overline{V} = (D - V)$) and to update j in V and (j, j') in E.

The algorithm proceeds as follow:

Let be j_k a selected feature from $\overline{V}$ at step k and assigned to V. We note $V^* = V \cup \{j_k\}$ and $\overline{V}^* = \overline{V} - \{j_k\}$. Thus, at step $k+1$ we seek :

$$\max_{(j,j') \in \overline{V^*} V^*} corr(j, j') = \max_{j \in \overline{V^*}} \max_{j' \in V^*} corr(j, j') \tag{5}$$

A marking procedure consists, in step k, after selecting j_k, to keep for each $j \in \overline{V^*}$ the values $\max_{j' \in V^*} corr(j, j')$. These values are calculated by the following updating formula :

$$\max_{j' \in V^*} corr(j, j') = max(\max_{j' \in V} corr(j, j') corr(j_k, j)) \tag{6}$$

Then, in step k we memorize for each $j \in \overline{V^*}$ the following values:
- $Pred(j)$, the farthest vertex of j in V^*:

$$Pred(j) = arg \max_{j' \in V*} corr(j, j') \tag{7}$$

- $L(j)$, the length between j and $Pred(j)$:

$$L(j) = \max_{j' \in V^*} corr(j, j') \tag{8}$$

Consequently, the construction of the tree is done from two vectors : $Pred$ and L of dimension p and from the correlation matrix $CORR$ of dimension $p \times p$. After $(p-1)$ iterations, the $(p-1)$ edges are $(j, Pred(j))$.

Finally, for extracting the features which characterize the class K, we detect the "Pivot" feature p^* according to (3) and we select then its directly linked features from the obtained spanning tree.

5 Results

We performed several experiments on four known problems from UCI Repository of machine learning databases [1]. They are voluntarily chosen (Table.1) for comparing our approach (G-Select) with another characterizing clustering methods.

Table 1. Characteristics of used databases

Data sets	N	p	#labels
Wave	5000	40	3
Madelon	2000	500	2
Wdbc	569	30	2
Spamb	4601	57	2

Remark that the used UCI data sets include class information (labels) for each observation. These labels are available for evaluation purposes but not visible to the clustering algorithm. Remember that the objective was to perform unsupervised classification that correctly identifies the underlying classes when the number of clusters is not predefined.

In general, the result of clustering is usually assessed on the basis of some external knowledge about how clusters should be structured. The only way to assess the usefulness of a clustering result is indirect validation, whereby clusters are applied to the solution of a problem and the correctness is evaluated against objective external knowledge. This procedure is defined by [11] as "validating clustering by extrinsic classification" and has been followed in many other studies. Thus, two statistical-matching schemes called Purity and Rand index [10] are used for the clustering accuracy.

5.1 Clustering Characterization

First, we give some details over the values assigned to different parameters for all used databases. For the construction of the maps, we have used an heuristic proposed by Kohonen [13] for automatically providing the initial number of neurons and the dimensions ($nrows \times ncolumns$). Thus, for *Wave*, $Dimension = 26 \times 14$, for *Wdbc*, $Dimension = 30 \times 4$, for *Madelon*, $Dimension = 17 \times 13$ and for *Spamebase*, $Dimension = 38 \times 9$. Then, we remind that the Euclidian distance is applied to define the dissimilarity level $\mathcal{D}$ between two p-dimensional referent vectors in the map. Moreover, for each *SOM rectangular-neighborhood order* threshold α (chosen between 1 and the number of SOM' rows), McSOM is performed multiple runs, each of them increasing the value of the dissimilarity threshold θ. Once all neighborhood and dissimilarity threshold values passed, the algorithm provides the optimal partitioning which maximizes *Generalized Dunn's* ($Dunn_G$) quality index [12]. This index is designed to offer a compromise between the inter-cluster separation and the intra-cluster cohesion. So, it is more appropriate to partition data set in compact and well-separated clusters.

$$Dunn_G = \frac{\min_{i,j,i \neq j} d_a(C_i, C_j)}{\max_h s_a(C_h)} \tag{9}$$

- $s_a(C_i)$ is the *average distance* within the cluster C_i
- $d_a(C_i, C_j)$ is the *between-cluster separation*

We start now by analyzing *Wave* data set because we know a priori that it is composed of 21 relevant features (the first ones) and 19 noisy features. The aim is to see if G-Select is able to select just the relevant ones. We have presented this database with all 40 features to McSOM and we have automatically obtained 3 classes with $< Purity, Rand > = < 0.55, 0.6706 >$. The number of classes corresponds exactly to the real defined one. Then, we have applied G-Select over these 3 classes and we have obtained for each one its characterizing features. Thus, G-Select provided for each class its "Pivot" feature which is directly linked to a subset of features extracted from the maximum spanning tree over all features. So, [14, *15*, 16, 17, 19] are selected for the first class, [2, 6, *7*, 8, 30] for the second class and [1, 9, 10, *11*, 12] for the third one (the numbers between ** represent "Pivot" features). We can see that G-Select allows the selection of 14 relevant features (with one noisy feature: 30) for the characterization. Nevertheless, The approach provides good rates on *Purity* and *Rand* with 0.5680 and 0.6713, respectively. Theses rates are better than those found by another weighting based characterization approach (lwd-SOM: $< 0.5374, 0.6068 >$, lwo-SOM: $< 0.5416, 06164 >$) in [8] with a minimum number of features (Table.2).

Table 2. G-Select vs lwo-SOM and lwo-SOM over *Wave* database

	lwd-SOM	lwo-SOM	G-Select
Classes:[features]	cl_1:[6-15] cl_2:[4-10] cl_3:[7-19]	cl_1:[3-8,11-16] cl_2:[8-11,14-19] cl_3:[3-20]	cl_1:[14-17,19] cl_2:[2,6-8,30] cl_3:[1,9-12]
Purity	0.5374	0.5416	0.5680
Rand	0.6068	0.6164	0.6713

We provide in Table 3 a comparison of the quality of McSOM before and after feature selection by G-Select over the other databases. In Table.4 we present the results of our approach versus lwd-SOM and lwo-SOM over the same databases.

We can see from all these tables that G-Select provides:

- An automatic characterization of classes from clustering, with the selection of an optimal number of features,
- An important elimination of noise,
- An improvement of SOM clustering after feature selection
- A good rates on *Purity* and *Rand* after selection compared to another characterization based methods.

Table 3. Clustering accuracy before and after feature selection

Data sets	Before Selection		After Selection	
	Rand	Purity	Rand	Purity
Wdbc	0.6769	0.7979	0.8843	0.9385
Madelon	0.5085	0.5825	0.5094	0.5945
Spambase	0.5195	0.6059	0.5197	0.6624

Table 4. Local feature selection for each class. [features]; $< Purity >$; [*]: [49 65 106 129 242 339 344 356 443 454 476 494].

Data sets	# cl	lwd-SOM	lwo-SOM	G-Select
Wdbc	2	$cl_1 - cl_9$: [4,24] $< 0.6274 >$	$cl_1 - cl_9$: [4,24] $< 0.8682 >$	cl_1:[7,8,23,28] cl_2:[3,7,8,28] $< 0.9285 >$
Madelon	2	cl_1:1; cl_2: [91, 281, 403-424] $< 0.5242 >$	cl_1:1; cl_2:[242, 417-452] $< 0.5347 >$	$cl_1 - cl_4$: [*] $< 0.5945 >$
Spamb	2	cl_1:56 ; cl_2:57 $< 0.6103 >$	cl_1:56 ; cl_2:57 $< 0.6413 >$	cl_1:[26,32,34,55] cl_2:[27,29,31,32, 34,40,43] $< 0.6624 >$

6 Conclusion and Future Work

We proposed in this paper a graph based framework for clustering and characterizing Self-organizing map. The proposal concerns more particularity G-Select which provides local feature selection from clustering obtained by a recently developed method called McSOM. in this approach we combined a statistical test for detecting "Pivot" features from obtained classes, with maximum spanning tree for extracting subsets of relevant features allowing their characterization. Some interesting issues can be raised from this work, for example the construction of graphs from several "Pivot" features and the optimization of the proposed approach to extend it for large databases analysis.

References

1. Asuncion, A., Newman, D.: UCI machinelearning repository (2007), `http://www.ics.uci.edu/mlearn/MLRepository.html`
2. Benabdeslem, K., Lebbah, M.: Feature selection for self organizing map. In: IMAC/IEEE ITI, pp. 45–50 (2007)
3. Cabanes, G., Bennani, Y.: A simultaneous two-level clustering algorithm for automatic model selection. In: ICMLA, pp. 316–321 (2007)
4. Cormen, T.H., Leiserson, E.C.E., Rivest, R.L., Stein, C.: Introduction to algorithms. MIT Press and McGraw-Hill (2001)

5. Dash, M., Choi, K., Scheuermann, P., Liu, H.: Feature selection for clustering-A filter solution. In: Proceedings of the IEEE International Conf. on Data Mining, pp. 115–122 (2002)
6. Elghazel, H., Benabdeslem, K., Kheddouci, H.: McSOM: Minimal coloring of self organizing map. In: Huang, R., Yang, Q., Pei, J., Gama, J., Meng, X., Li, X. (eds.) Advanced Data Mining and Applications. LNCS (LNAI), vol. 5678, pp. 128–139. Springer, Heidelberg (2009)
7. Frigui, H., Nasraoui, O.: Unsupervised learning of prototypes and attribute weights. Pattern Recognition 37(3), 567–581 (2004)
8. Grozavu, N., Bennani, Y., Lebbah, M.: From feature weighting to cluster characterization in topographic unsupervised learning. In: IEEE International Joint Conference on Neural Network, pp. 1005–1010 (2009)
9. Huang, J.Z., Ng, M.K., Rong, H., Li, Z.: Automated feature weighting. IEEE Trans. Pattern Anal. Mach. Intell. 27, 657–668 (2005)
10. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classication 2, 193–218 (1985)
11. Jain, A., Murty, M.: Data clustering: A review. ACM Computing Surveys 31, 264–323 (1999)
12. Kalyani, M., Sushmita, M.: Clustering and its validation in a symbolic framework. Pattern Recognition Letters 24(14), 2367–2376 (2003)
13. Kohonen, T.: Self organizing Map. Springer, Berlin (2001)
14. Lebart, L., Morineau, A., Warwick, K.: Multivariate descriptive statistical analysis. John Wiley and Sons, New York (1984)
15. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: 5-th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
16. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66, 846–850 (1971)
17. Vesanto, J., Alhoniemi, E.: Clustering of the self organizing map. IEEE Transactions on Neural Networks 11(3), 586–600 (2000)
18. Welsh, D.J.A., Powell, M.B.: An upper bound for the chromatic number of a graph and its application to timetabling problems. Computer Journal 10(1), 85–87 (1967)