A New Way for Hierarchical and Topological Clustering

Hanane Azzag and Mustapha Lebbah

1 Introduction

Clustering is one of the most important unsupervised learning problems. It deals with finding a structure in a collection of unlabeled data points. Hierarchical clustering algorithms are typically more effective in detecting the true clustering structure of a structured data set than partitioning algorithms. We find in literature several important research in hierarchical cluster analysis [Jain et al., 1999]. Hierarchical methods can be further divided to agglomerative and divisive algorithms, corresponding to bottom-up and top-down strategies, to build a hierarchical clustering tree. Another works concerning hierarchical classifiers are presented in [Jiang et al., 2010]. In this paper we propose a new way to build a set of self-organized hierarchical trees.

Self-organizing models (SOM) are often used for visualization and unsupervised topological clustering [Kohonen et al., 2001]. They allow projection in small spaces that are generally two dimensional. Some extensions and reformulations of SOM model have been described in the literature [Hammer et al., 2009], [Bishop et al., 1998, Rossi and Villa-Vialaneix, 2010]. Hierarchical version of SOM are also defined in [Vesanto and Alhoniemi, 2000]. A variety of these topological maps algorithms are derived from the first original model proposed by Kohonen. All models are different from each other but share the same idea: depict large datasets on a simple geometric relationship projected on a reduced topology (2D). SOM model has several tree-structured versions such as TS-SOM [Koikkalainen and Horppu, 2007], GH-SOM [Dittenbach et al., 2000], TreeSOM

Hanane Azzag · Mustapha Lebbah
Université Paris 13, Sorbonne Paris Cité,
Laboratoire d'Informatique de Paris-Nord (LIPN),
CNRS(UMR 7030),
99, avenue Jean-Baptiste Clément
Villetaneuse, 93430 France
e-mail: `firstname.secondname@lipn.univ-paris13.fr`

F. Guillet et al. (Eds.): *Advances in Knowledge Discovery & Management*, SCI 471, pp. 85–97.
DOI: 10.1007/978-3-642-35855-5_5 © Springer-Verlag Berlin Heidelberg 2013

[Samsonova et al., 2006] and SOM-AT [Peura, 1998]. Our approach should not be confused with these methods, it is totally different from TS-SOM, GH-SOM in which the map architecture has the form of a tree. Each neuron of map now becomes one node of tree. On the other hand TreeSOM proposed to generate a hierarchical tree where only the leaf nodes may get many data elements, and other nodes none. SOM-AT introduce matching and adjusting schemes for input data attribute trees. The most optimal tree is selected to represent input data.

Concerning the visual aspect of our studies, we can find in the literature several algorithms for visualizing hierarchical structures, which are mostly 2D. One may cite treemap method which recursively maps the tree to embedded rectangles [Johnson and Shneiderman, 1991, Shneiderman, 1992]. Hyperbolic displays have also been studied in 2D and 3D [Carey et al., 2003]. Another example is the cone tree [Robertson et al., 1991]: the root of the tree is the top of a cone. The subtrees of this root are all included in this cone. The size of a cone may depend on the number of nodes which are present in each tree. In this work we introduce a new method named MTM (Map Tree Map), that proposes a self-organizing treemap, which provides a simultaneously hierarchical and topological clustering. Each cell of map represents a tree structured data and treemap method provides a global view of the local hierarchical organization. Data moves toward a map of trees according to autonomous rules that are based on nearest neighborhood approach. The topological process of the proposed algorithm is inspired from SOM model and the rules for building tree are inspired from autonomous artificial ants method [Azzag et al., 2007, Slimane et al., 2003]. The rest of this paper is organized as follows: in section 2, we present both SOM model and proposed model. In section 3, we show the experimental results on several data sets. These data sets illustrate the use of this algorithm for topological and visual hierarchical clustering. Finally we offer some concluding comments of proposed method and the further research.

2 Hierarchical and Topological Clustering Model

We present in this paper a new model that provides a hierarchical clustering of data where each partition is a forest of trees organized in a 2D map. The obtained map is inspired from SOM algorithm and could be seen as a forest of trees.

2.1 Self-Organizing Maps

Self-organizing maps are increasingly used as tools for visualization and clustering, as they allow projection over small areas that are generally two dimensional. The basic model proposed by Kohonen (SOM: Self-Organizing-Map) consists of a discrete set C of cells called map. This map has a discrete topology defined by undirected graph, it is usually a regular grid in 2 dimensions. We denote p as the number of cells. For each pair of cells (c,r) on the map, the distance $\delta(c,r)$ is defined as the length of the shortest chain linking cells r and c on the grid without sub-trees. For each cell c this distance defines a neighbor cell; in order to control the neighborhood area, we introduce a kernel positive function $\mathcal{K}$ ($\mathcal{K} \geq 0$ and $\lim_{|x|\to\infty} \mathcal{K}(x) = 0$). We define the mutual influence of two cells c and r by $\mathcal{K}(\delta(c,r))$. In practice, as for traditional topological map we use smooth function to control the size of the neighborhood as:

$$\mathcal{K}(\delta(c,r)) = \exp\left(\frac{-\delta(c,r)}{T}\right) \tag{1}$$

Using this kernel function, T becomes a parameter of the model. As in the SOM algorithm, we increase T from an initial value T_{max} to a final value T_{min}. Let $\mathcal{R}^d$ be the euclidean data space and $\mathcal{A} = \{\mathbf{x}_i; i = 1,\dots,n\}$ a set of observations, where each observation $\mathbf{x}_i = (x_i^1, x_i^2, \dots, x_i^d)$ is a continuous vector in $\mathcal{R}^d$. For each cell c of the grid, we associate a referent vector $\mathbf{w}_c = (w_c^1, w_c^2, \dots, w_c^j, \dots, w_c^d)$ of dimension d. We denote by $\mathcal{W}$ the set of the referent vectors. The set of parameter $\mathcal{W}$, has to be estimated from $\mathcal{A}$ iteratively by minimizing a cost function defined as follows:

$$\mathcal{R}(\phi, \mathcal{W}) = \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{r \in C} \mathcal{K}^T(\delta(\phi(\mathbf{x}_i), r))||\mathbf{x}_i - \mathbf{w}_r||^2 \tag{2}$$

where ϕ assigns each observation $\mathbf{x}$ to a single cell in the map C. In this expression $||\mathbf{x} - \mathbf{w}_r||^2$ is a square of the Euclidean distance. At the end of learning, SOM provides a partition of p subsets.

2.2 *Proposed Model: Map Treemap*

The proposed model uses the same grid process, combined with a new concept of neighborhood. Our model seeks to find an automatic clustering that provides a hierarchical and topological organization of observations $\mathcal{A}$. This model is presented as regular grid in 2D that has a topological order of p cells. Each cell c is the 'root support' of a tree denoted by $Tree_c$ and each node $N_{\mathbf{x}_i}$ of the tree represents a data $\mathbf{x}_i$. More precisely the proposed model defines a forest of trees organized on a 2D map called C. Taking into account the proximity between two trees on the map C is a useful information which allows to define a topological neighborhood relation used in traditional topological maps. Thus, for each pair of trees $Tree_c$ and $Tree_r$ on the map, the distance $\delta(c,r)$ is defined as the length of the shortest chain linking cells r and c on the map associated to $Tree_c$ and $Tree_r$. To model the influence between $Tree_r$ and $Tree_c$ we use a neighborhood function $\mathcal{K}$ defined above (eq. 1). Thus, the mutual influence between $tree_c$ and $tree_r$ is defined by the function $\mathcal{K}^T(\delta(c,r))$ where T represents the temperature function that controls the size of the neighborhood. We associate to each tree a representative point denoted $\mathbf{w}_c$ which is a given data denoted $\mathbf{x}_i$ in $tree_c$ ($\mathbf{w}_c = \mathbf{x}_i \in tree_c$). Choosing a representative point allows easily adapting our algorithm to any type of data (categorical, binary, and mixed data data ... etc). The objective function of self-organizing trees is defined as follows:

$$\mathcal{R}(\phi, \mathcal{W}) = \sum_{\mathbf{c} \in \mathbf{C}} \sum_{\mathbf{x}_i \in Tree_c} \sum_{r \in C} \mathcal{K}^T(\delta(\phi(\mathbf{x}_i), r)) ||\mathbf{x}_i - \mathbf{w}_r||^2 \tag{3}$$

Minimizing the cost function $\mathcal{R}(\phi, \mathcal{W})$ is a combinatorial optimization problem. In practice, we seek to find the best (optimal) solution by using batch version. In this work we propose to minimize the cost function in the same way as "batch" version but using statistical characteristics provided by trees to accelerate the convergence of the algorithm. Three basic steps are necessary to minimize the cost function and are defined as follows:

1. **Assignment step.** Each datum $\mathbf{x}_i$ is connected in $Tree_c$ and forms a hierarchical relationship noted parent-child. We denote by $nodeChild(\mathbf{x}_i)$ the function, which provides all child nodes with the same parent node $N_{\mathbf{x}_i}$ associated to the data $\mathbf{x}_i$. At step $t = 0$, $nodeChild(\mathbf{x}_i) = \mathbf{x}_i$.

   Assignment step consists of finding for each observation $\mathbf{x}_i$ a best match tree called "Winner" using the assignment function named χ. This tree is also defined as winner tree. The children nodes of $\mathbf{x}_i$ ($nodeChild(\mathbf{x}_i)$). In other words, all nodes of tree $N_{\mathbf{x}_i}$ are recursively assigned to the winning tree. The assignment function is defined as follows:

   $$\chi(nodeChild(\mathbf{x}_i)) = \arg\min_r \sum_{c \in C} \mathcal{K}^T(\delta(r, c)) ||\mathbf{x}_i - \mathbf{w}_c||^2 \tag{4}$$

   where, $\mathbf{w}_c \in \mathcal{A}$
2. **Building Tree step.** In this step we seek to find the best position of a given data $\mathbf{x}_i$ in the $Tree_c$ associated to cell c. We use connections/disconnections rules inspired from [Azzag et al., 2007, Slimane et al., 2003]. Each data will be connected to its nearest neighbor. The particularity of the obtained tree is that each node N whether it is a leaf or an internal node represents a given data $\mathbf{x}_i$. In this case, $N_{\mathbf{x}_i}$ denotes the node that is associated to the data $\mathbf{x}_i$, $N_{\mathbf{x}_{pos}}$ represents current node of the tree and $N_{\mathbf{x}_{i+}}$ the node connected to $N_{\mathbf{x}_{pos}}$, which is the most similar (closest by distance) to $N_{\mathbf{x}_i}$. We also note V_{pos} the local neighborhood observed by $N_{\mathbf{x}_i}$ and the node connected $N_{\mathbf{x}_{pos}}$ in the concerned tree.

   Let $T_{Dist}(N_{\mathbf{x}_{pos}})$ be the highest distance value which can be observed among the local neighborhood V_{pos}. $\mathbf{x}_i$ is connected to $N_{\mathbf{x}_{pos}}$ if and only if the connection of $N_{\mathbf{x}_i}$ further increases this value. Thus, this measure represents the value of the maximum distance observed in the local neighborhood V_{pos}, between each pair of data connected to the current node $N_{\mathbf{x}_{pos}}$:

   $$\begin{aligned} T_{Dist}(N_{\mathbf{x}_{pos}}) &= Max_{j,k} \left\| N_{\mathbf{x}_j} - N_{\mathbf{x}_k} \right\|^2 \\ &= Max_{j,k} \left\| \mathbf{x}_j - \mathbf{x}_k \right\|^2 \end{aligned} \tag{5}$$

   Connections rules consist of comparing a node $N_{\mathbf{x}_i}$ to the nearest node $N_{\mathbf{x}_{i+}}$. In the case where both nodes are sufficiently far away ($||N_{\mathbf{x}_i} - N_{\mathbf{x}_{i+}}||^2 > T_{Dist}(N_{\mathbf{x}_{pos}})$)

the node $N_{\mathbf{x}_i}$ is connected to its current position $N_{\mathbf{x}_{pos}}$. Otherwise, the node $N_{\mathbf{x}_i}$ associated to $\mathbf{x}_i$ is moved toward the nearest node $N_{\mathbf{x}_{i+}}$. Therefore, the value T_{Dist} decreases for each node connected to the tree. In fact, each connection of a given data $\mathbf{x}_i$ implies a local minimization of the value of the corresponding T_{Dist}. Therefore it implies a minimization of the cost function (3).

It can be observed that, for any node of the tree, the value $T_{Dist}(N_{pos})$ is only decreasing, which ensures the termination and the minimization of the cost function. At the end of the tree construction step, each cell c of the map C will be associated to $tree_c$.

3. **Representation step.**
Minimizing the cost function $\mathcal{R}(\phi,\mathcal{W})$ with respect to $\mathbf{w}_c$ corresponds to finding the point that minimizes all local distances weighted by neighborhood function.

$$\mathbf{w}_c = \min_{\mathbf{w}_c \in tree_c} \sum_{\mathbf{x}_i \in \mathcal{A}} \mathcal{K}(\delta(c,\chi(\mathbf{x}_i))) \|\mathbf{x}_i - \mathbf{w}_c\|^2, \quad \forall c \in C \tag{6}$$

The temperature T evolves according to the iterations from T_{max} to T_{min} in the same way as traditional topological maps. In the practical case we use neighborhood function as following:

$$\mathcal{K}^T(x) = e^{\frac{-\delta(r,c)}{T}}$$

We present below the detail of MTM algorithm 4.

2.3 Topological Order in MTM Model

The decomposition of the cost function $\mathcal{R}$ that depends on the value of T, allows to rewrite its expression as follows:

$$\mathcal{R}^T(\chi,\mathcal{W}) = \left[\sum_c \sum_{r \neq c} \sum_{\mathbf{x}_i \in tree_r} \mathcal{K}^T(\delta(c,r)) \|\mathbf{x}_i - \mathbf{w}_r\|^2\right] + \left[\mathcal{K}^T(\delta(c,c)) \sum_c \sum_{\mathbf{x}_i \in tree_c} \|\mathbf{x}_i - \mathbf{w}_c\|^2\right]$$

where $\delta(c,c) = 0$

The cost function $\mathcal{R}$ is decomposed into two terms. In order to maintain the topological order between trees, minimizing the first term will bring trees corresponding to two neighboring cells. Indeed, if $Tree_c$ and $Tree_r$ are neighbors on the map, the value of $\delta(c,r)$ is lowest and in this case the value of $\mathcal{K}^T(\delta(c,r))$ is the highest. Thus, minimizing the first term has as effect reducing the value of the cost function. Minimizing the second term corresponds to the minimization of the inertia of points connected to the $Tree_c$ (in the case of Euclidean space). Minimizing this term is considered as applying hierarchical clustering algorithm (AntTree).

Algorithm 4. Detail of MTM algorithm

Input: Map C of n_c cells, learning set A, the number of iteration n_{iter}
Output: Map C of n_c empty cells or which contain sub-tree
for $c \in C$ **do**
    $\mathbf{w}_c = \mathbf{x}_i$
end for
{ /* random Initialization of the map */}
for $t = 1$ to n_{iter} **do**
    for $\mathbf{x}_i \in A$ **do**
        if first assignment of $\mathbf{x}_i$ **then**
            Find the "wining" cell $\chi(\mathbf{x}_i)$ by using the assignment function defined in (eq. 4)
            Associate the data $\mathbf{x}_i$ to a node $N_{\mathbf{x}_i}$,
            Connect the node $N_{\mathbf{x}_i}$ in the sub-tree $Tree_{\chi(\mathbf{x}_i)}$ by using connection rules
            Update the representative point $\mathbf{w}_c$ by using the defined expression (eq. 6)
        else
            Find the "wining" cell $c_{new} = \chi(nodechild(\mathbf{x}_i))$ by using function defined in 4 {/* t^{th} assignment for the data $\mathbf{x}_i$*/}
        end if
        if $c_{new} \neq c_{old}$ **then**
            Assign data $\mathbf{x}_i$ and the child node $nodechild(\mathbf{x}_i)$ to the new cell c_{new}
            Connect the node $N_{\mathbf{x}_i}$ and the child node in the sub-tree $tree_{c_{new}}$ by using connection rules.
            Update the two representative points $\mathbf{w}_{c_{old}}$ and $\mathbf{w}_{c_{new}}$ by using the defined function (eq.6)
        end if
    end for
end for

For different values of temperature T, each term of the cost function has a relative relevance in the minimization process. For large values of T, the first term is dominant and in this case, the priority is to preserve the topology. When value of T is lowest, the second term is considered in the cost function. In this case, the priority is to determine representative compact trees. Our model provides a solution to regularized AntTree algorithm: regularization is achieved through the constraint of ordering on the trees.

3 Comparatives Results

3.1 Visual Exploration of MTM

We have tested and compared the proposed algorithm on several datasets that have been generated with Gaussian and Uniform distributions. Others have been extracted from the machine learning repository [Blake and Merz, 1998] and have several difficulties (fuzzy clustering, no relevant feature, ...). Before comparing our numerical results, we present a map visualization with associated treemaps.

Treemap is a visualization technique introduced in [Shneiderman, 1992]. An important feature of treemaps is that it makes very efficient use of display space. Thus it is possible to display large trees with many hierarchical levels in a minimal amount of space (2D). Treemap can be helpful when dealing with large clustered tree. Treemaps lend themselves naturally to showing the information encapsulated in the clustering tree. Viewing a tree at some level of abstraction, the viewer is really looking at nodes belonging to some level in the tree. A treemap can display the whole structure of trees and allow the users to place the current view in context. In the proposed visualization technique, each tree is represented by a treemap. This aims to obtain an automatic organization of treemaps on a 2D map. Figure 1 shows an example of four tree structures with its corresponding treemaps. The positioning of tree nodes in a treemap is a recursive process. The nodes are represented as rectangles of various shapes. First, the children of the root are placed across the display area horizontally. Then, for each node N already displayed, each of N's children is placed across vertically within N's display area. This process is repeated, alternating between horizontal and vertical placement until all nodes have been displayed. We note that each rectangle is colored according to the real label of its corresponding node/data. This makes easy a visual comparison of homogeneous clusters.

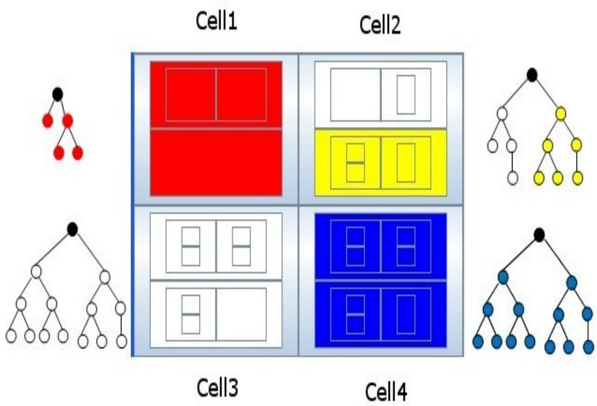


Fig. 1 Map treemaps representation: 2×2 MTM

In figure 1 each treemap represents a hierarchical organization of data belonging to cluster "tree". Thus, MTM approach has several properties that allow obtaining a simultaneous topological hierarchical clustering. We observe in figure 1 that data placed in the $tree_c$ are similar to $N_{\mathbf{x}_i}$ and the child nodes of $N_{\mathbf{x}_i}$ represent recursively subtrees that are dissimilar to their "sister" subtrees. In order to best analyze the obtained result, we have learned for each dataset 1×1 MTM in order to build a single treemap. Figures 2,3, and 4 display some example of 1×1 MTM and 4×4 MTM. Observing both maps on each dataset, we find that our algorithm provides a MTM, which is a multi-divisions of the main treemap. We can see that topological and hierarchical organization of data is more apparent. In order to visualize the coherence between intra-organization of treemaps and the label points, we assign one color to each label. In each figure (2, 3, 4), we distinguish two regions on the MTM that are dedicated to the pure and mixed clusters. Map presented in Figure 2.b shows diagonal from right to left is dedicated to one class (colored in blue) and the treemap positioned in the bottom right is a mixed cluster. We observe in this treemaps, that yellow point is positioned in a lower level on the tree, this behavior is normal since the yellow classes are situated in the neighborhood. Same remarks concern Lsun and Tetra dataset. In figure 4 observing the top right treemap (cell) and the bottom left, we can conclude on the level and the side where cluster will become mixed. Thus, this visual analysis is done using only 2D visualization unlike SOM method where we can not conclude on which level data is positioned. This visual system allows analyst to easily navigate trough the databases and to let the user easily interact with the data and perceive details, global context and shape of the tree.

3.2 Comparison with Other Clustering Methods

We remind here that MTM model provides more information than traditional hierarchical models, K-means or others. In this work we compare the obtained result with SOM model. In this case we adopt the same parameter: map size, initial and final neighborhood.

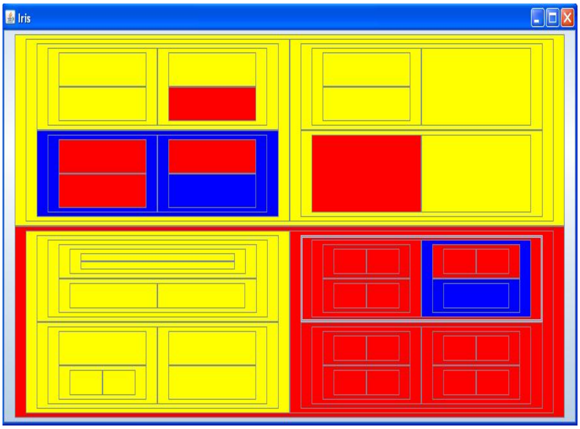
To measure the quality of map clustering, we adopt the approach of comparing results to a 'ground truth'. We use two criterions for measuring the clustering results. The first one is Rand index which measures the percentage of observation

Table 1 Competitive results obtained with AHC, MTM and SOM using the same parameter (map size, initial and final parameter T). *DB* is the Davides-Bouldin index.

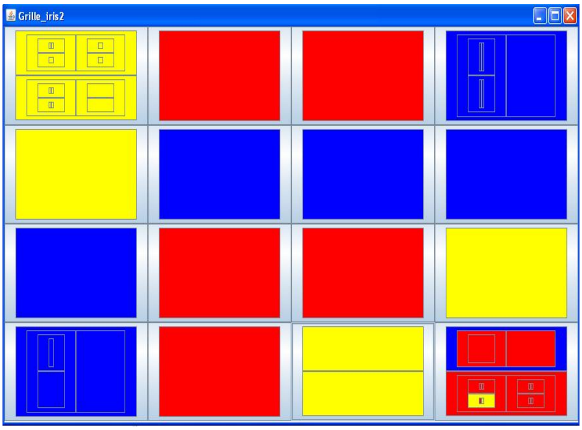
Datasets	size.	MTM		SOM		CAH	
	DB	Rand	*DB*	Rand	*DB*	Rand	
Atom(2)	800	1.4	0.88	1.47	0.51	0.81	0.77
Anneaux (2)	1000	0.80	0.61	0.90	0.51	0.50	0.55
ART1(4)	400	0.98	0.81	0.85	0.81	0.79	0.88
Demi-cercle(2)	600	0.58	0.60	0.67	0.5	0.55	0.48
Glass(7)	214	1.56	0.70	2	0.65	0.65	0.72
Hepta(7)	212	0.92	0.92	0.85	0.93	0.35	1.00
Iris(3)	150	1.06	0.75	1.03	0.75	0.43	0.77
Lsun(3)	400	0.97	0.71	1.09	0.72	0.54	0.85
Pima(2)	768	1.09	0.5	2.23	0.43	0.65	0.56
Target(6)	770	1.4	0.85	1.17	0.58	0.44	0.81
Tetra(4)	400	0.82	0.81	1.25	0.76	0.71	0.99
TwoDiamonds(2)	800	0.86	0.60	0.81	0.51	0.57	1.00

pairs belonging to the same class and which are assigned to same cluster of the map [Saporta and Youness, 2002]. The second index is Davides Bouldin criterion which [Davies and Bouldin, 1979] is used to determine the optimal number of centroids for K-means.

Table 1 reports clustering evaluation criterion obtained with MTM, SOM and AHC. MTM method provides results quite comparable to those obtained with SOM method on the majority of cases. Looking to columns (DB and Rand index) associated to MTM, we observe that DB index value is lower using our algorithm and rand index is highest near one for the majority of datasets.

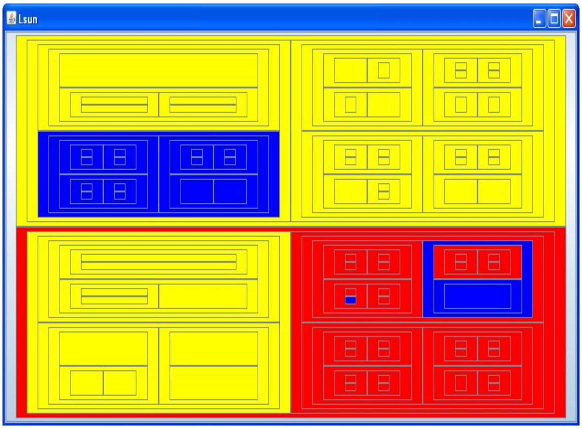


(a) 1×1 Treemap of data set

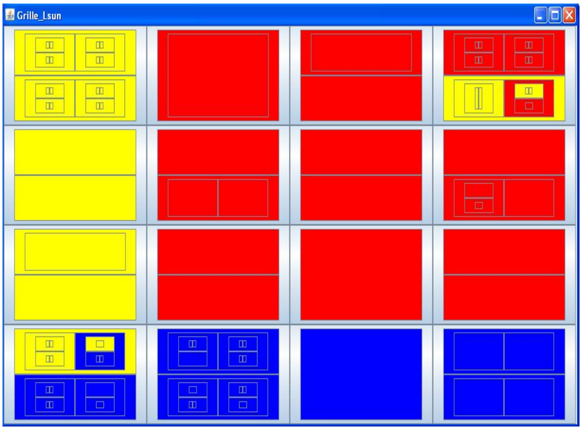


(b) 4×4 MTM

Fig. 2 Iris Dataset



(a) 1×1 Treemap of data set

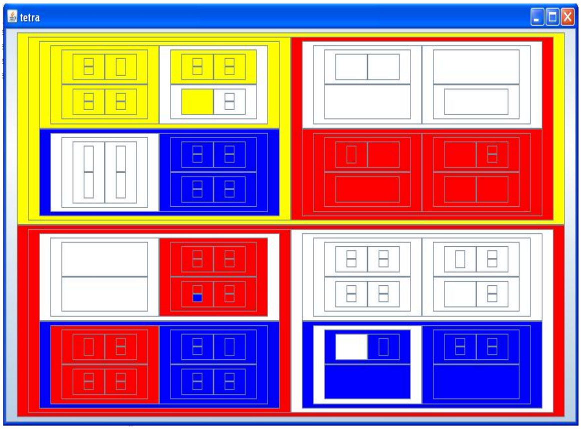


(b) 4×4 MTM

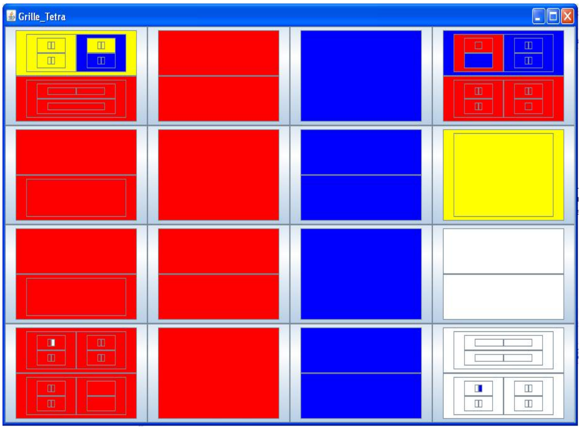
Fig. 3 Lsun Dataset

Concerning AHC method [Jain et al., 1999], we have used DB index to select the number of clusters. This justifies the best results of DB index obtained by AHC comparing to MTM. Indeed DB is lower for the majority of cases but not far away comparing to DB index obtained by MTM. Concerning Rand index values, MTM obtains similar results as AHC for the majority of cases.

Our purpose through this comparison is not to assert that our method is the best, but to show that MTM method can obtain quite the same good results as SOM or other well known clustering algorithms. Unlike SOM method or AHC, MTM does not require a posterior processing to analyze the structure of data belonging to clusters. MTM provides simultaneously hierarchical and topological



(a) 1×1 Treemap of data set



(b) 5×5 Map-Treemaps

Fig. 4 Tetra Dataset

clustering which is more interesting for visualization task. Thus MTM has two main advantages:

1. Complexity: we reduce the number of assignments. When an observation is re-assigned to another tree, the entire sub-trees associated to this observation will follow it into the new cell (see expression 4).
2. Data projection and rapid visualization: In our model, we don't need to use a traditional projection of the map to get an idea about the structure of data. Treemap organization of data presents a local structure for each cell of the map.

4 Conclusion and Perspectives

In this paper, we have presented a new algorithm dedicated to hierarchical clustering that has the following properties: it provides a local hierarchical clustering of data, that allows a better visualization of the obtained organization. It generates both 2D self-organization of the trees associated to each cell and hierarchical organization provided by tree. The obtained results have been compared to those obtained by traditional Kohonen algorithm (SOM) and AHC. This comparison shows that the proposed approach is promising and can be used in various applications of data mining. The major benefits of MTM approach are the following: MTM uncovers the hierarchical structure of the data allowing the user to understand and analyze large amounts of data. Using the various emerging trees at each cell being rather small in size, it is much easier for the user to keep an overview of the various clusters.

Results presented in this papaer are preliminary and much work still be done. It is obvious that using trees for data clustering greatly speeds up the learning process, we wish to generalize these algorithms to other kind of structures which may not be trees. The same principles seem to be applicable also to graphs. Also, it will be necessary to focus on the visual aspect of our approach. Indeed, we will develop a 2D/3D view of the different trees that result from the hierarchical clustering in order to allow an interactive exploration of data.

References

[Azzag et al., 2007] Azzag, H., Venturini, G., Oliver, A., Guinot, C.: A hierarchical ant based clustering algorithm and its use in three real-world applications. European Journal of Operational Research 179(3), 906–922 (2007)

[Bishop et al., 1998] Bishop, C.M., Svensén, M., Williams, C.K.I.: Gtm: The generative topographic mapping. Neural Computation 10(1), 215–234 (1998)

[Blake and Merz, 1998] Blake, C., Merz, C.: UCI repository of machine learning databases (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html

[Carey et al., 2003] Carey, M., Heesch, D.C., Rüger, S.M.: Info navigator: A visualization tool for document searching and browsing. In: Proc. of the Intl. Conf. on Distributed Multimedia Systems (DMS), pp. 23–28 (2003)

[Davies and Bouldin, 1979] Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Transactions on Pattern Recognition and Machine Intelligence 1(2), 224–227 (1979)

[Dittenbach et al., 2000] Dittenbach, M., Merkl, D., Rauber, A.: The growing hierarchical self-organizing map, pp. 15–19. IEEE Computer Society (2000)

[Hammer et al., 2009] Hammer, B., Hasenfuss, A., Rossi, F.: Median Topographic Maps for Biomedical Data Sets. In: Biehl, M., Hammer, B., Verleysen, M., Villmann, T. (eds.) Similarity-Based Clustering. LNCS, vol. 5400, pp. 92–117. Springer, Heidelberg (2009)

[Jain et al., 1999] Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3), 264–323 (1999)

[Jiang et al., 2010] Jiang, W., Raś, Z.W., Wieczorkowska, A.A.: Clustering Driven Cascade Classifiers for Multi-indexing of Polyphonic Music by Instruments. In: Raś, Z.W., Wieczorkowska, A.A. (eds.) Advances in Music Information Retrieval. SCI, vol. 274, pp. 19–38. Springer, Heidelberg (2010)

[Johnson and Shneiderman, 1991] Johnson, B., Shneiderman, B.: Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In: Proceedings of the 2nd Conference on Visualization 1991, VIS 1991, pp. 284–291. IEEE Computer Society Press, Los Alamitos (1991)

[Kohonen et al., 2001] Kohonen, T., Schroeder, M.R., Huang, T.S. (eds.): Self-Organizing Maps, 3rd edn. Springer-Verlag New York, Inc., Secaucus (2001)

[Koikkalainen and Horppu, 2007] Koikkalainen, P., Horppu, I.: Handling missing data with the tree-structured self-organizing map. In: IJCNN, pp. 2289–2294 (2007)

[Peura, 1998] Peura, M.: The self-organizing map of trees. Neural Process. Lett. 8(2), 155–162 (1998)

[Robertson et al., 1991] Robertson, G.G., Mackinlay, J.D., Card, S.K.: Cone trees: animated 3d visualizations of hierarchical information. In: CHI 1991: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 189–194. ACM Press, New York (1991)

[Rossi and Villa-Vialaneix, 2010] Rossi, F., Villa-Vialaneix, N.: Optimizing an organized modularity measure for topographic graph clustering: A deterministic annealing approach. Neurocomput. 73, 1142–1163 (2010)

[Samsonova et al., 2006] Samsonova, E.V., Kok, J.N., Ijzerman, A.P.: Treesom: Cluster analysis in the self-organizing map. neural networks. American Economic Review 82, 1162–1176 (2006)

[Saporta and Youness, 2002] Saporta, G., Youness, G.: Comparing two partitions: Some proposals and experiments. In: Proceedings in Computational Statistics, pp. 243–248. Physica Verlag (2002)

[Shneiderman, 1992] Shneiderman, B.: Tree visualization with tree-maps: A 2-D space-filling approach. ACM Transactions on Graphics 11(1), 92–99 (1992)

[Slimane et al., 2003] Slimane, A.M., Azzag, H., Monmarché, N., Slimane, M., Venturini, G.: Anttree: a new model for clustering with artificial ants. In: IEEE Congress on Evolutionary Computation, pp. 2642–2647. IEEE Press (2003)

[Vesanto and Alhoniemi, 2000] Vesanto, J., Alhoniemi, E.: Clustering of the self-organizing map. IEEE Transactions on Neural Networks 11(3), 586–600 (2000)