SoTree : Self-organizing of hierarchical clustering

Hanene Azzag and Mustapha Lebbah

University of Paris 13
LIPN-UMR 7030 - CNRS
99, avenue Jean-Baptiste Clément
93430 Villetaneuse, France
hanene.azzag@lipn.univ-paris13.fr
mustapha.lebbah@lipn.univ-paris13.fr

Abstract

We propose in this paper a new approach for hierarchical clustering. Our method called SoTree builds, autonomously and simultaneously, a topological and hierarchical partitioning of data's. Each "cluster" of the partition represents one cell of a 2D grid and is modelized by a tree. In the tree each node represents a given data. We then present the obtained results on standard data sets with varying difficulty. The preliminary results are encouraging and promising to continue in this direction.

1 Introduction

The accumulation of large structured dataset collected from different sources requires developing new methods for clustering and visualization in order to understand the informations involved in data. Mining and visualizing these structured dataset represent a real challenge in machine learning.

Data clustering is identified as one of major problems in data mining. Popularity and different variations linked to the clustering problem (Jain et al. 1988)(Jain et al. 1999), have given birth to a several methods. These methods can both use heuristic or mathematics principles.

In this work we introduce a new method named SoTree: Self-Organizing Tree which uses simultaneous clustering: hierarchical and topological. Data moves autonomously respecting differents rules, towards a 2D grid where each cell represents a tree structured data. Thus we will obtain in one pass: an Horizontal topological clustering on the 2d grid and a vertical hierarchical clustering for each cell. The topological function of our algorithm is based on Kohonen approach (Kohonen 2001)(Ultsch 2005) and the rule for building tree is based on biomimetic method (Azzag et al. 2006a) (Azzag et al. 2006b).

The remaining of this paper is organized as follows : in section 2, we present the main principles of our model and the rules proposed to build tree clustering. Section 3 is devoted to the methodology and experimental results. Finally section 4 concludes on this work and proposes some perspectives.

2 Proposed model

A variety of 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 to present large datasets on a simple geometric relationship projected on a reduced topology (2D). The model that we propose uses the same grid process, combined with a new concept of neighborhood.

Our model seek to find an automatic clustering that provides a hierarchical topological organization of a set of observations $A = \{\mathbf{x}_i \in \mathcal{R}^d, i = 1..n\}$ where $\mathbf{x}_i = (x_i^1, x_i^2, ..., x_i^j, ..., x_i^d)$. This model is presented as regular grid in a 2D dimensions which have a topological order of n_c cells. This map has a discrete topology defined by an undirected graph. Each cell c is the root of a subtree denoted $Tree_c$ and each node $N_{\mathbf{x}_i}$ of the subtree represents a data $\mathbf{x}_i$. More precisely our model defines a set of subtrees projected on a 2D grid called $\mathcal{C}$.

Taking into account the proximity between two subtrees on the map $\mathcal{C}$ is a useful information which permit us to define a topological neighborhood relation previously used in traditional topological maps. Thus, for each pair of cells c and 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 associated to subtrees $Tree_c$ et $Tree_r$.

To model the influence of two cell r and c (depending on their proximity), we use a neighborhood function defined from a kernel positive function $\mathcal{K}$ ($\mathcal{K} \geq 0$ and $\lim_{|y|\to\infty} \mathcal{K}(y_i) = 0$). The mutual influence between two subtrees $tree_c$ and $tree_r$ which are roots of two cells c and r will be defined by the function $\mathcal{K}^T(\delta(c,r))$ where T represents the temperature function to control the size of the neighborhood.

We also associate to each subtree a representative point noted w_c that is a given data noted $\mathbf{x}_i$ in the subtree $tree_c$ ($\mathbf{w}_c = \mathbf{x}_i \in tree_c$). Choosing a representative point permit us to easily adapt our algorithm to any type of data. We need just to define a new measure of similarity.

The quality of the subtrees partitioning is defined by the following cost function :

$$\mathcal{R}(\chi, \mathbf{w}) = \sum_{\mathbf{x}_i \in A} \sum_{r \in C} \mathcal{K}^T(\delta(\chi(\mathbf{x}_i), r)) ||\mathbf{x}_i - \mathbf{w}_r||^2 \quad (1)$$

where χ assigns each data $\mathbf{x}_i$ to the unique cell c of the map.

Minimizing the cost function $\mathcal{R}$ is a combinatorial optimization problem. In practice, we seek to find the best (optimal) solution by using batch version (Kohonen 2001).

In this work we propose to minimize the cost function in the same way as "batch" version but using statistical characteristics provided by sub-trees (associated to each cell) to accelerate the convergence of the algorithm. Three basic steps necessary for minimizing the cost function are defined as follows :

- **Step of building Tree**

  After each assignment of a given data $\mathbf{x}_i$ to a cell c, we will seek to find the best position in the tree $Tree_c$ associated to this cell. To do this, we will use local rules of connections/ disconnections inspired by hierarchical clustering named AntTree (Azzag et al. 2006a).

  The particularity of the obtained tree is that each node N whether leaf or internal node represents a given data $\mathbf{x}_i$. $N_{\mathbf{x}_i}$ denotes the node that will be connected and 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 us denote by $T_{Dist}(N_{\mathbf{x}_{pos}})$ 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 $N_{\mathbf{x}_i}$ increases further this value. Thus, this measure defines the value of the maximum distance observed in the local neighborhood V_{pos}, between each pair of data connected to the current node $N_{\mathbf{x}_{pos}}$:

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

  In other words, connections rules consist to compare a node $N_{\mathbf{x}_i}$ to the nearest node $N_{\mathbf{x}_{i+}}$. In the case where both nodes are sufficiently far away ($\|N_{\mathbf{x}_i} - N_{\mathbf{x}_{i+}}\|^2 > T_{Dist}(N_{\mathbf{x}_{pos}})$) then the node $N_{\mathbf{x}_i}$ will be connected to its current position $N_{\mathbf{x}_{pos}}$.

  Otherwise, the node $N_{\mathbf{x}_i}$ associated to data $\mathbf{x}_i$ will be moved toward the nearest node $N_{\mathbf{x}_{i+}}$. Therefore the value T_{Dist} will decrease 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}.

  At the end of the tree construction step, each cell c of the map $\mathcal{C}$ will be associated to a sub-tree $tree_c$. Connections rules are based on Nearest Neighbor approach. Each data will be connected to its nearest neighbor.

- **Assignment step**

  Each data $\mathbf{x}_i$ is connected in the sub-tree $Tree_c$ forming a hierarchical relation noted parent-child. We use *nodeChild*($\mathbf{x}_i$) function which provided all child node of a same node

parent $N_{\mathbf{x}_i}$ associated to the data $\mathbf{x}_i$. At the first $t = 0$, $nodeChild(\mathbf{x}_i) = \mathbf{x}_i$.

In the same way as traditional topological maps, assignment step consist to find for each given data $\mathbf{x}_i$ a cell called "Winner" using the affectation function named χ. This cell will be also designated as winner cell for all k-nearest-neighbors of $\mathbf{x}_i$. In other words, a complete root sub-tree $N_{\mathbf{x}_i}$ will be recursively assigned to the winning cell. The assignment function is defined as following:

$$\chi(nodeChild(\mathbf{x}_i)) = \arg\min_{r} \sum_{c \in C} \mathcal{K}^T\left(\delta(r,c)\right) \|\mathbf{x}_i - \mathbf{w}_c\|^2 \tag{3}$$

By minimizing the cost function, the properties of Euclidean distance permit to obtain a compact and separated clusters: observations in the same sub-tree are close (compactness) and far away from observations assigned to other sub-trees (separation).

- **Representation step**

Minimizing $\mathcal{R}$ with respect to $\mathbf{w}_c$ corresponds to find the point that minimizes all local distances.

$$\mathbf{w}_c = min_{\mathbf{w}_c \in tree_c} \sum_{\mathbf{x}_i \neq \mathbf{w}_c ; \mathbf{x}_i \in tree_c} \|\mathbf{x}_i - \mathbf{w}_c\|^2, \quad \forall c \in C \tag{4}$$

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 :

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

3 Results

We have evaluated and compared our algorithms on several datasets that have been generated with gaussian and uniform distributions. Others have been extracted from the machine learning repository (Blake et al. 1998) and have several difficulties (fuzzy clustering, no relevant feature). Before comparing our numerical results, we present a visualization of the obtained map with their associated trees.

In the figure 1 we present the visual results obtained by our approach with *Iris* dataset. Sub-trees placed on each cell represent the "vertical" clustering provided by SoTree. Each sub-tree represent one cluster which contains all data of corresponding cell. Thus SoTree approach has several properties that permit us to obtain a well hierarchical clustering.

In this work we have developed rules that respect the following properties: Each node $N_{\mathbf{x}_i}$ is the most representative of its sub-tree. We observe that data placed in the sub-tree $tree_c$ are similar to $N_{\mathbf{x}_i}$ and the child node of $N_{\mathbf{x}_i}$ represents recursively sub-trees that are dissimilar to their "sister" sub-trees.

Table 1 summarizes obtained results. We can notice that SoTree is the method which approximates the best the number of classes. We notice however that the real number of clusters is not necessarily representative of the real number of clusters (or point clouds) in the data. The obtained purity is approaching 100%. Comparing with the two other algorithms, we can conclude that SoTree is better than AntTree and have similar performance AHC method.

4 Conclusions et perspectives

In this work we have developed a new method of unsupervised hierarchical clustering that has the following properties: It provides a local hierarchical clustering of data, this allows a better visualization of data organization for each cell in the map. It also generate both a "vertical" self-organization of the sub-trees in the cells and an "horizontal" organization provided by the topology and the neighborhoods function.

As perspectives, our results are preliminary and much work will be done. Actually comparing with AHC and AntTree methods we have seen that our approach obtains competitive results on several datasets. As future work, it will be wise to focus on the visual aspect of our approach. Indeed we will develop a 2D/3D view of the different trees resulting from the hierarchical clustering to allow an interactive exploration of data.

(1) - Input: Map $\mathcal{C}$ of n_c cells, learning set A , the number of iteration n_{iter}
(2) - Output : Map $\mathcal{C}$ of n_c empty cells or which contain sub-tree
(3) **for** c**to** $\in C$
(4) $\mathbf{w}_c = \mathbf{x}_i$ /* random Initialization of the map */
(5) **for** $t = 1$ **to** n_{iter}
(6) **for** $\mathbf{x}_i$ **to** $\in A$
(7) - $T = T^{max} \times \left(\frac{T^{min}}{T^{max}}\right)^{\frac{t}{n_{iter}-1}}$
(8) **if** first affectation of $\mathbf{x}_i$ **then**
(9) - Find the "wining" cell $\chi(\mathbf{x}_i)$ by using the affectation function defined in (eq. 3)
(10) - Associate the data $\mathbf{x}_i$ to a node $N_{\mathbf{x}_i}$,
(11) - Connect the node $N_{\mathbf{x}_i}$ in the sub-tree $Tree_{\chi(\mathbf{x}_i)}$ by using connections rules to build the tree
(12) - Update the representative point $\mathbf{w}_c$ by using the defined function (eq. 4)
(13)
(14) **else** /* t_{th} affectation for the data $\mathbf{x}_i$*/
(15)
(16) - Find the "wining" cell $c_{new} = \chi(nodeSon(\mathbf{x}_i))$ by using function defined in 3
(17) **if** $c_{new} \neq c_{old}$ **then**
(18) - Affect data $\mathbf{x}_i$ and the child node $nodeSon(\mathbf{x}_i)$ to the new cell c_{new}
(19) - Connect the node $N_{\mathbf{x}_i}$ and the child node in the sub-tree $tree_{c_{new}}$ by using connections rules.
(20) - Update the two representative points $\mathbf{w}_{c_{old}}$ and $\mathbf{w}_{c_{new}}$ by using the defined function (eq.4)
(21)
(22)
(23)

Alg 1: Detail of $SoTree$ algorithm

References

(Azzag et al. 2006a).Hanane Azzag, Christiane Guinot, and Gilles Venturini. Data and text mining with hierarchical clustering ants. pages 153189, 2006.

(Azzag et al. 2006b). Hanene Azzag, Christiane Guinot, and Gilles Venturini. Swarm Intelligence and Data Mining, volume 34 of Swarm intelligence and data mining, Studies in Computational Intelligence, chapter Data and text mining with hierarchical clustering ants, pages 153190. Springer-Verlag, 2006.

(Blake et al. 1998). C.L. Blake and C.L Merz. Uci repository of machine learning databases. technical report. Technical report, University of California, Department of information and Computer science, Irvine, CA, available at: ftp://ftp.ics.uci.edu/pub/machine-learning-databases, 1998.

(Jain et al. 1988). Anil K. Jain and Richard C. Dubes. Algorithms for clustering data. Prentice Hall advanced reference series:Computer Science, 1988.

(Jain et al. 1999). A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. ACM Computing Surveys, 31(3):264323, 1999.

(Kohonen 2001). T. Kohonen. Self-organizing Maps. Springer Berlin, 2001.

(Ultsch 2005). A Ultsch. Clustering with SOM: U*C. In In Proc. Workshop on Self-Organizing Maps, Paris, France.http://www.uni-marburg.de/fb12/datenbionik/, pages 75 82, 2005.

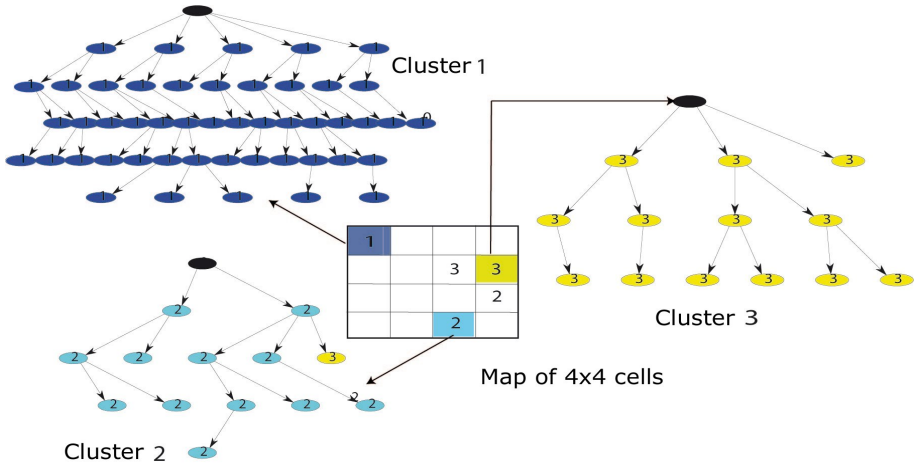

Figure 1: Visual results obtained by Iris dataset.

Datasets (Real. Class.)	Dim.	SoTree P_R (C_f)	AntTree P_R (C_f)	AHC P_R (C_f)
IRIS (3)	150	0.88 (5)	0.94 (5)	0.88 (3)
TWODIAMONDS (2)	800	0.99 (2)	0.99 (7)	0.99 (2)
ART1 (4)	400	0.83 (4)	0.77 (8)	0.84 (5)
ART4 (2)	200	1.0 (2)	0.98 (4)	01.0 (3)
ART6 (4)	400	1.0 (4)	0.93 (4)	01.0 (5)
GLASS (7)	214	0.36 (4)	0.45 (9)	0.49 (3)
THYROID (3)	215	0.73 (3)	0.88 (9)	0.84 (5)

Table 1: Competitive results obtained with SoTree approach, Ascending Hierarchical clustering and AntTree Algorithm. C_f is the number of found clusters and P_R the purity of clusters