An idea of a clustering algorithm using support vector machines based on binary decision tree

Halima ELAIDI, Younes ELHADDAR, Zahra BENABBOU, Hassan ABBAR
Laboratory of Information and Decision Support Systems.
Hassan 1st University, Settat, Morocco
Email: ha.elaidi@uhp.ac.ma, y.elhaddar@uhp.ac.ma, jo.benabbou@gmail.com, h.abbar@encgsettat.ac.ma

***Abstract*—Clustering is a technique which is commonly known in the domain of machine learning as an unsupervised method,it aims at constructing from a set of objects some different groups which are as homogeneous as possible. On the other hand support vector machines (SVM) and binary decision trees (BDT) were proposed and developed as supervised learning techniques where the output assembly is previously known. In this work we will try to build a clustering algorithm that uses the two supervised methods we cited above.**

***Index Terms*—Clustering, support vector machine, binary decision tree, supervised learning, unsupervised learning.**

I. Introduction

Our main idea is inspired from the approach of "decision tree based multi-class support vector machine" [7], which is a supervised method and some other scientific productions using the same principle of coupling "decision trees" and "support vector machines"[5][6][8][9]. The approach we are talking about is just like any other decision tree algorithm[1]: it has as a starting point a root which should contain all the records of the data set. Then, by using the two-class SVM algorithm, we can construct two classes with the highest value of dissimilarity possible (functions measuring the dissimilarity, differ from one algorithm to an other). Next, we deal with the produced classes with the same manner, until reaching the leaves. Thus, the objective of our paper is to propose an extension of that approach to unsupervised learning (clustering).

The key idea of the algorithm proposed by this paper is to construct a decision tree (DT) in which each decision node is a two-class SVM, and each two-class SVM produces the two most homogeneous clusters possible, then we determine the hyperplane that separates the produced groups. We repeat the procedure until the stopping criterion is reached. And as a result, a binary decision tree is constructed.

The number of clusters obtained it also depends on the stopping criterion. The stopping criterion of this algorithm can take two forms: the group to be separated contains only one element, or all possible distances between processed group of elements are less than the minimum distance below which data separation can never be executed. The effectiveness of our algorithm requires that the distance threshold is to be given and fixed by experts in the studied field.

II. Support Vector Machine

Support vector machines (SVM), introduced by vapnik [1], are famous classification techniques based on the theory of statistical learning and have been successfully applied to classification and regression problems[4].

A. Linear Support Vector Machines

We will start with the simplest case: the data are linearly separable. In the case of non-separate data, the analysis results in a quadratic programming problem.

Considering a binary problem on the learning set $\{(x_i, y_i)_{i=1}^n\} \in \mathbb{R}^l \times \{+1, -1\}$, a hyperplane separating positive and negative examples exists in the case of linearly separable data (a "separator hyperplane"). The points x which lie on the hyperplane satisfy w.x + b = 0, where w is normal to the hyperplane, x represents the argument, and b is a constant value. Thus, the problem for the linearly separable case could be formulated as (1).

$$\begin{cases} \underset{w,b}{Min} & \frac{1}{2}\|w\|^2 \\ s.t. & y_i(w^T x_i + b) \geq 1 \\ & \qquad i=1..n \end{cases} \tag{1}$$

Now our problem will be transformed with the Lagrangian method to a problem of quadratic programming (QP). in order to derive the optimal hyperplane.To do this, we first convert the constrained problem given by (1) into the unconstrained problem:

$$L_p = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i \{y_i[w^T x_i + b] - 1\} \tag{2}$$

We must now minimize L_p with respect to w, b, and simultaneously require that the derivatives of LP with respect to all the α_i vanish,

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{3}$$

$$\sum_i \alpha_i y_i = 0 \tag{4}$$

Since these are equality constraints in the dual formulation, we can substitute them into Eq. (2) to give

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j x_i . x_j \tag{5}$$

978-1-5386-4396-9/18/31,00$ ©2018 IEEE

The solution of our problem now comes from either the minimization of L_P or the maximization of L_D. Note that L_Pp denotes the primal problem and L_D denotes the dual problem.

Notice that there is a Lagrange multiplier α_i for every training point. In the solution, those points for which $\alpha_i > 0$ are called "support vectors".

B. The Non-Separable Case

In the case where the data are not separable, we opt for two approaches to find the solution. The first approach is called soft margin SVM which transfers the problem to (6)

$$\begin{cases} \underset{w,b}{Min} \quad \frac{1}{2}\|w\|^2 + C\sum_i \zeta_i \\ s.t. \quad y_i(w^T x_i + b) \geq 1 - \zeta_i \\ \qquad\qquad \zeta_i \geq 0, \quad i=1..n \end{cases} \tag{6}$$

where C is a parameter that must be chosen by the user, a large C means the assignment of a higher penalty to errors.nor their Lagrange multipliers, appear in the Wolfe dual problem, which becomes:

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j x_i . x_j \tag{7}$$

subject to:

$$0 \leq \alpha_i \leq C \tag{8}$$

$$\sum_i \alpha_i y_i = 0 \tag{9}$$

The solution is again given by

$$w = \sum_{i=1}^{N_s} \alpha_i y_i x_i \tag{10}$$

where N_S is the number of support vectors. Thus the only difference from the optimal hyperplane case is that the α_i now have an upper bound of C.

The other approach is to map the data from the input space into a higher dimensional feature space using a mapping which we will call ϕ:

$\phi : \mathbb{R}^l \longrightarrow \mathcal{H}$

The use of this mapping or know what ϕ, will not be taken into account because the learning algorithm will only answer data through the dot product in $\mathcal{H}$. i.e. on functions of the form $\phi(x_i).\phi(x_j)$.

Since the solution in feature space involves only inner products of the mapped points, one can obtain the optimal hyperplane by kernel trick. Finally, the decision function is achieved as (11)

$$h(x) = \sum_i y_i \alpha_i k(x, x_i) \tag{11}$$

and $f(x) = sign(h(x))$ is determined as the classification output, where ai is the lagrange multiplier of x_i.

III. Unsupervised learning: clustering

clustering is an unsupervised method of machine learning which aims at dividing data into groups(clusters), each group should contain the most similar objects[2]. Technically speaking, the idea behind the clustering algorithms is logically simple, since it is based on introducing a measure of similarity between objects and consider the ones having the highest levels of similarity as belonging to the same cluster, while the rest of entities which are dissimilar are kept in different clusters. The similarity measure is in most cases a distance meter such as Euclidean distance, Manhattan distance, Minkowski distance[3]... Thus, the quality of a given clustering algorithm depends on the measure of similarity (dissimilarity) chosen to be used while implementing such an algorithm.

We often distinguish the "hierarchical", "partitional" and "Density Based" methods:

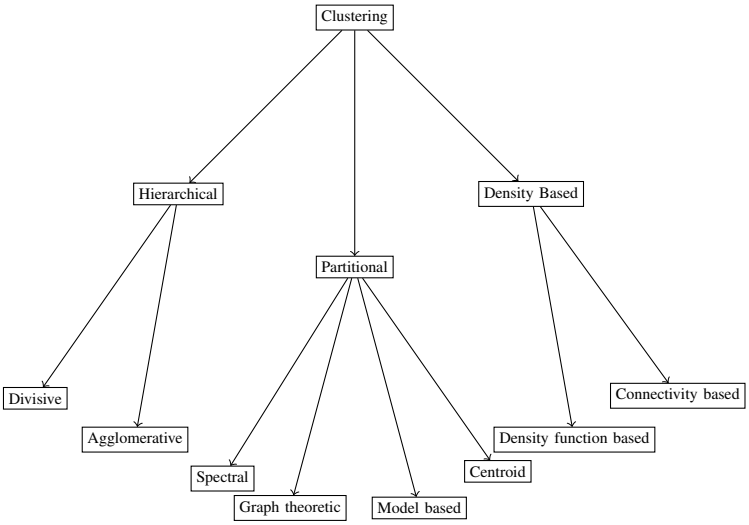

Figure 1 : The most known clustering techniques[3].

- **Hierarchical clustering**:regroups algorithms which build a hierarchy of clusters, they are either top-down algorithms (divisive) or bottom-up algorithms (agglomerative).
- **Partitional clustering**:consists of constructing a partition of the data-set's elements and make of them a set of k clusters.One of the most commonly used partitional clustering algorithms is the K-means.
- **Density Based clustering**:aims at locating the regions of data with high density which are separable by the use of regions with lower density.

A. Proposed method

In this paper, we propose a clustering algorithm based on the tree structured multi-class SVM. Our proposed tree is necessary binary, since at each node, the SVM which is originally formulated for a two class problem, is applied.

We assume that our algorithm only needs two parameters: The distances between the elements of generated groups at each node and the partitioning stop criterion which is

a distance threshold below which the separation stops, i.e. the minimum distance with which separation can be made.

Let N be the number of data-set patterns denoted as X_i, where i=1,2, ... ,N.

The euclidean distance between ith element and jth element, is as follows:

$$Ed_{ij} = ||X_i - X_j|| \tag{12}$$

In our proposed clustering method, we start from the top of the decision tree which contains all the entities of the training data, and we construct a matrix which contains all the possible distances between the elements of the training data. On the basis of these calculated distances and by using SVM, we construct the first hyperplane which must separate two entirely different groups (clusters). The idea of separation we had, we decided to call it **the open window technique** in fact, on the basis of the matrix of distances obtained, we treat the two most distant points and we will consider that the entities of our database represent points belonging to a window, by opening this window its points that were the most distant become the closest points, while those who were closest become the farthest.

Thus, each of these two farthest points of which we talked previously becomes the representative element of a group, so that each of them forms a cluster. By using the same matrix of distances obtained at the beginning, for each point of the remaining we observe its distance from the two points representing the two groups, finally we conclude that each treated point must belong to the group with which it has the smallest distance. So we get two totally different groups.

For the given nodes, we repeat the same operations until reaching the stopping criterion i.e.

- The cluster obtained contains only one data.
- The distance between the two furthest points within a group is less than a minimum distance threshold fixed in advance.

To simplify the explanation we propose a simple example illustrating the different steps of our algorithm. Therefore, consider the following cloud of points:

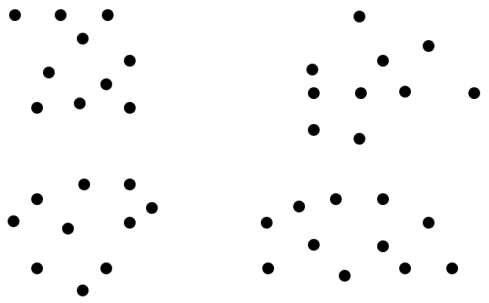

Figure 2.1 : The data of an illustrative example.

For this set of points, we calculate the matrix of Euclidean distances linking all the points without exception, then we observe the two most distant points as shown in our example in the figure below:

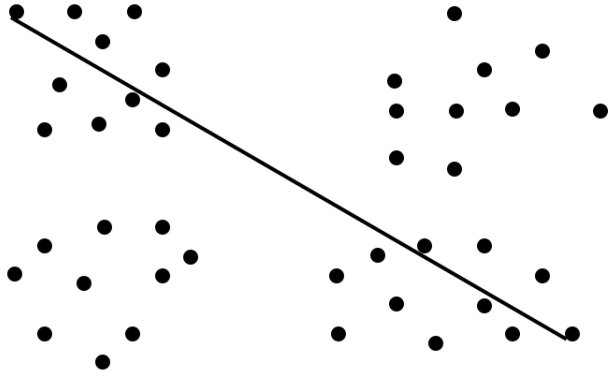

Figure 2.2 : Determination of the two most distant data.

At this level we are about to build two groups, each represented by one of these two points. In a first place, we begin by determining the line linking the two points (we note it (D)), to be able to construct the two other lines passing each one by one of the two points and which are perpendicular to (D), lets note (always on the base of our explanatory example) the line passing by the right representative point as (A) and the other one passing by the left representative point as (B). Which is pretty clear on the following figure:

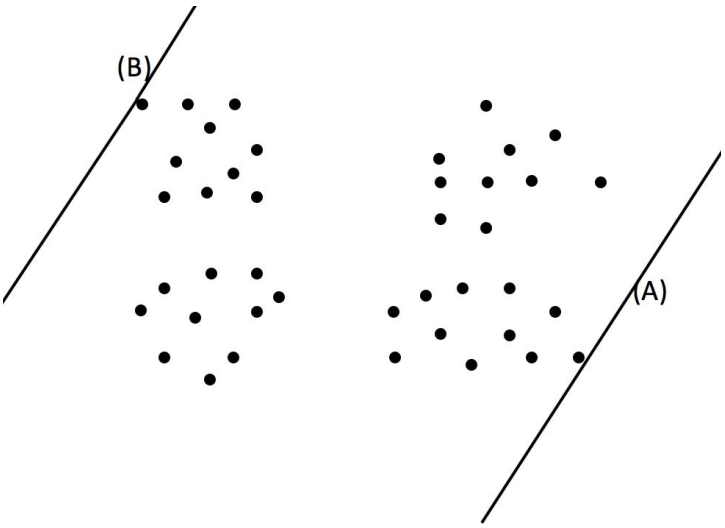

Figure 2.3 : Division of the database into two homogeneous groups.

As we already mentioned, we can use the same matrix of distances we built at the beginning and observe their distances to the two representative points of our two groups.Thereby, each element of the data set belongs to the group having the minimum distance with its representative.

Afterward, each point belonging to the group on the right side (using our last figure) will rotate 180 degree around the axis (A). In the same way, the elements belonging to the other group will make a rotation (of 180 degree as well) but in the opposite direction around the axis (B). Hence the

appellation of "open window technique".

Since the rotation is 180 degrees, this geometric transformation is identical to the orthogonal symmetry, i.e. we can use the principal of orthogonal symmetry (or axial symmetry) to build the image of each point with respect to the axes (A) or (B) (depending on the group to which the point belongs), then we continue the work with only the images of the points by axial symmetry. Next, by applying the SVM separation technique, let the group of points of the right part be the class 1 and the group of the left part the class -1, and calculate a hyperplane which separates the two classes.

so we will have two groups completely and easily separated, as shown on the following figure:

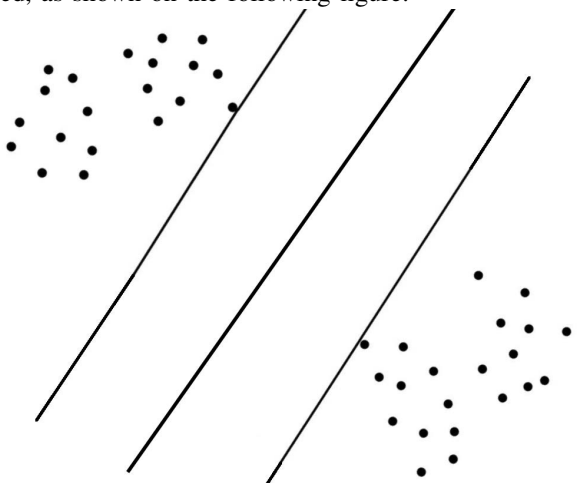

Figure 2.4 : The resulting groups of the first separation at the root node.

we retreat the resulting groups following the same steps until reaching one of the stop criteria we cited above.

Finish the above steps, we assume that the stopping criterion fixed at the beginning has led us to the following result (always in the context of our example):

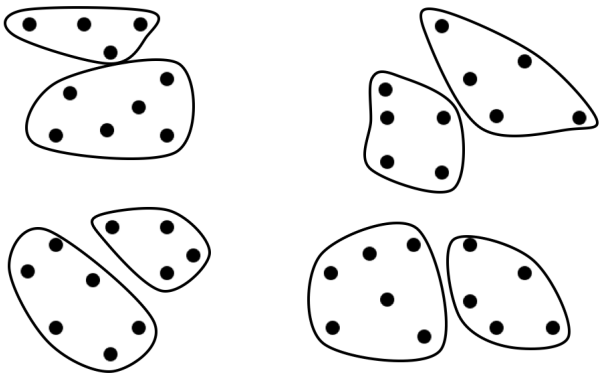

Figure 3.1 : The resulting groups using the proposed algorithm.

Let figure 3.2 be the obtained tree, structured by following the steps of our proposed algorithm:

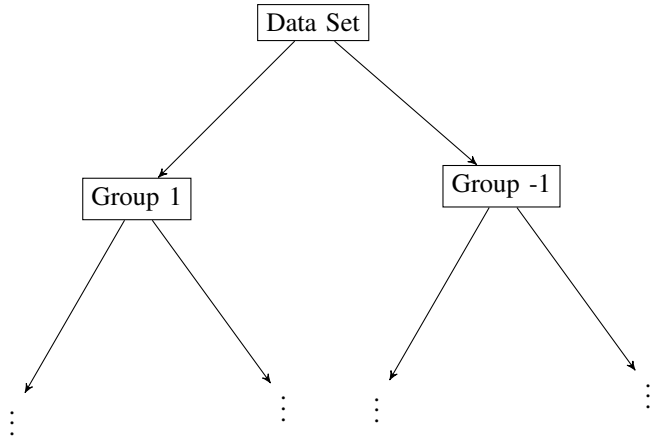

Figure 3.2 : The shape of the resulting decision tree.

IV. The algorithm steps

Step1: Give a distance threshold below which the separation stops: d_s.
Step2: Establish the Euclidean Distance Matrix.
Step3: Find the two most distant points i and j, i.e. the maximum euclidean distance Ed_{ij}.

- ***If*** such two points exist and $Ed_{ij} \geq d_s$
- ***Then*** group 1 $\leftarrow pointi$ And group -1 $\leftarrow pointj$.

Step4: For each point k of the remaining points (other than i and j) compare Ed_{ik} and Ed_{jk}:

- ***If*** $Ed_{ik} < Ed_{jk}$
  - ***Then*** point k $\in$ group 1
- ***Else***
  - ***If*** $Ed_{ik} > Ed_{jk}$
    - ***Then*** point k $\in$ group -1
  - ***Else*** the point k is indifferent between groups 1 and -1

Step5: Determine the two axes of rotation:

- Axis 1: passes by the point i and perpendicular to the line linking the points i and j.
- Axis 2: passes by the point j and perpendicular to the line linking the points i and j.

Step6:

- Each point in group 1 will rotate around axis 1 with 180Âř.
- Each point in group -1 will rotate around axis 1 with -180Âř.

Step7: Using the SVM, build the hyperplane that separates the 2 groups.
Step8: For each of the resulting groups, return to step 3.
Step9: End.

V. Conclusion

In this paper, we proposed a novel clustering algorithm based on two efficient supervised techniques, which are binary decision trees and support vector machines.The clusters obtained towards the end of our new algorithm must be entirely different, because the methods used to differentiate

them and which are at the base methods that were distinguished to the classification problems, are precise and having a minimum rate of error.

Furthermore, the idea on which our algorithm is based, minimizes absolutely the dissimilarity within each of the produced clusters, for the simple reason that the most dissimilar clusters should be separated at the very beginning of the tree. Because quite simply they are easy to separate.

It is real that our proposed algorithm produces homogeneous clusters. But it is also quite clear that for a treated data set, the accomplished separation is rigorous, because some of the resulting clusters can be merged together. Thus, in a future work we will propose a method that will constitute the remedy of this little problem, so that the algorithm can give more efficient results.

REFERENCES

[1] V.N. Vapnik. *The nature of statistical learning theory, Second Edition, Springer, pp. 131-145*, 2000.

[2] B.Mirkin. *"Clustering for Data Mining: A Data Recovery Approach, pp 112-136"*, 2005.

[3] J. Irani, N. Pise and M. Phatak. *"Clustering Techniques and the Similarity Measures used in Clustering: A Survey, pp: 10-11"*, 2016.

[4] C. Cortes and V. Vapnik. *"Support-Vector Networks, Machine Learning, Vol.20, No.3, pp. 273-297"*, 1995.

[5] K.P. Bennett and J. A. blue *"A Support Vector Machine Approach to Decision Trees"*, Troy, New-York, 1998.

[6] U. Fayyad *"A Tutorial on Support Vector Machines for Pattern Recognition"*, Kluwer Academic Publishers, Boston. Manufactured in The Netherlands., 1998.

[7] F. Takahashi and S. Abe *"Decision Tree Based Multiclass Support Vector Machine"*, Graduate School of science and technology. Kobe University - Kobe,Japan, 2002

[8] S. Cheong, S. H. oh and S. Y. Lee *"Support Vector Machines with Binary Tree Architecture for Multiclass Classification"*. Korea Advanced Institute of Science and Technology,Daejeon, Korea, 2004.

[9] J.Guo, N.Takahashi and W.Hu. *"An efficient algorithm for multi-class support vector machines"*, 2008.