An Elliptical-Shaped Density-Based Classification Algorithm for Detection of Entangled Clusters

Stanley Smith, Mylene Pischella and Michel Terré
CNAM CEDRIC/LAETITIA
292 rue Saint-Martin - 75003 Paris, France
Email: stanley.smith@isen-lille.fr; mylene.pischella@cnam.fr; terre@cnam.fr;

***Abstract*—We present a density-based clustering method producing a covering of the dataset by ellipsoidal structures in order to detect possibly entangled clusters. We first introduce an unconstrained version of the algorithm which does not require any assumption on the number of clusters. Then a constrained version using a priori knowledge to improve the bare clustering is discussed. We evaluate the performance of our algorithm and several other well-known clustering methods using existing cluster validity techniques on randomly-generated bi-dimensional gaussian mixtures. Our simulation results show that both versions of our algorithm compare well with the reference algorithms according to the used metrics, foreseeing future improvements of our method.**

I. Introduction

With the fast-increasing amount of data stored and conveyed by modern applications the need to rely on efficient and effective procedures to retrieve relevant information has gained considerable concern. Cluster analysis is one such approach designed to extract underlying structures from the data which has garnered a lot of interest over the last decades. A wide variety of methods and algorithms have been developed and successfully applied in manifold applications including image processing, document retrieval and data mining [1]. However, a major and persistent limitation to many clustering techniques is the detection of merged or entangled clusters. We thus propose a density-based algorithm which attempts to partially circumvent this difficulty by enforcing the shape of the clusters as ellipsoidal. We present a first version of our method independent of the number of real clusters in the data, then discuss a constrained improvement of the raw classification. We evaluate the performance of our algorithm and compare it with reference clustering algorithms, showing that both versions yield conclusive results which can serve as the baseline for further refinements.

The rest of the paper is organized as follows : in section II we introduce some mathematical notations and give a brief overview of the clustering problem, citing several well-known existing classification methods from the literature. Section III is devoted to the description and discussion of our elliptical clustering technique. Section IV sets the assumptions and evaluation criteria used to conduct our simulations and presents some comparative performance results between our algorithm and standard clustering methods. General conclusions and perspectives for our future work are outlined in section V.

II. Notations and related work

A. Mathematical notations and definition of clustering

We first set some mathematical notations which are used throughout the paper : scalars are be written using lower-case letters ; vectors are be written using lower-case bold letters ; matrices are written using upper-case bold letters, and the transpose operation by a T-exponent (*e.g.* $\mathbf{A}^T$); finite sets is written using upper-case cursive letters ; the cardinal and complement of a set are written using the absolute value notation $|\cdot|$ (*e.g.* $|\mathcal{A}|$) and an overline (*e.g.* $\overline{\mathcal{A}}$), respectively.

We formally introduce the clustering problem as follows : given a dataset $\mathcal{D}$ of N multidimensional observations denoted by $\mathbf{x}_n$, where $n \in \{1, \cdots N\}$, we aim at classifying the dataset in K meaningful clusters $\mathcal{C}_k$, $k \in \{1, \cdots K\}$, such that all members of any class are more similar to each other than elements taken from different classes. For the sake of simplicity we shall consider that the data points are D-dimensional real vectors, $\mathbf{x}_n \in \mathbb{R}^D$, which allows us to rely on metrics such as the euclidean distance to evaluate similarity between observations.

B. Properties of clusterings and existing algorithms

From a general point of view clustering techniques can usually be characterized by three properties. First, the number of clusters K required to sort the data may be known in advance or used as a constraint on the algorithm, but may also be a result of the classification process itself. Besides, one may consider by classifying the observations a partitioning of the dataset $\mathcal{D}$, meaning that each $\mathbf{x}_n$ exclusively belongs to a single cluster $\mathcal{C}_k$. Fuzzy clusterings on the other hand allow data points to belong to several classes with different membership degrees. Finally, some clustering methods are capable of detecting and handling the presence of outliers, which correspond to irrelevant observations. We here define outliers with respect to a clustering as observations which do not belong to any cluster, *i.e.* $\mathbf{x_n} \in \cap_{k=1}^{K} \overline{\mathcal{C}_k}$.

More specific distinctions can be made between clustering techniques based on the underlying procedure used to perform the classification. Some algorithms aim at optimizing an objective cost function, the absolute minimum of which corresponds to the best clustering of the dataset. One of the most widely-used representative of this class of algorithms is k-means [2]. k-product [3] is another such optimization-based clustering

ISBN 978-0-9928626-7-1

algorithm designed to find the centroids $\left(\{\mu_{\mathbf{k}}\}_{k\in\{1,\cdots K\}}\right)$ of the clusters. Its utility function is defined as

$$J_{KP}\left(\{\mu_{\mathbf{k}}\}_{k\in\{1,\cdots K\}}\right)=\sum_{n=1}^{N}\prod_{k=1}^{K}\parallel \mathbf{x_n}-\mu_{\mathbf{k}}\parallel^2 \qquad (1)$$

and has been proved to be convex in one and two dimensions. Density-based clustering algorithms represent classes as dense regions of the observation space. DBSCAN (Density Based Spatial Clustering of Applications with Noise) [4] and OPTICS (Ordering Points to Identify the Clustering Structure) [5] are popular members of this class of methods. Other famous techniques such as the E-M (Expectation-Maximization) algorithm [6] adopt a probabilistic description of the data and make use of statistical tools to classify it. Hierarchical clusterings rely on iteratively merging or splitting clusters [7]. Many other approaches and refinements exist in the literature [8].

Clustering methods may usually encounter two major limitations. First, the classification produced by a given algorithm can be very sensitive to its initialization configuration, thus requiring several runs to extract the best possible result. Such approaches relying on the minimization of non-convex functions like k-means are typically concerned with this difficulty. Another persistent drawback of classification procedures is that they often fail to detect and handle entangled clusters. We attempt to address the latter with our proposed algorithm.

III. The elliptical clustering algorithm

We now introduce our elliptical clustering algorithm, a density-based classification technique which performs a covering of the dataset by elliptical-shaped structures. In its original version this classification technique does not require any assumption on the number of real classes, but it also admits a constrained version which aims at improving the final clustering by a post-processing of the raw result. While by construction particularly suited for gaussian mixtures of comparable variances, it can be applied to any type of dataset. In the first two sections we describe the underlying concepts motivating the elliptical clustering as well as the classification process, then in section III-C we address how the knowledge of the number of real clusters K can be enforced in the resulting clustering. Section III-D provides a complete overview of the algorithm by means of a pseudo-code.

A. The weighted density function

The elliptical clustering method is based on a representation of clusters as dense regions of the observation space. Similarly to the DBSCAN algorithm, it alternates two steps : firstly detecting a new cluster, then expanding it, until all or enough data points have been processed. However, while DBSCAN relies on local density criteria, we here adopt a more global point of view and represent the overall density of the dataset by means of weighted hyperbolic tangent kernel functions. The resulting density estimation function, denoted by $\rho_{\mathbf{w}}(\mathbf{x})$, is defined as

$$\rho_{\mathbf{w}}(\mathbf{x})=\frac{1}{\sum_{n=1}^{N}w_n}\sum_{n=1}^{N}w_n\left(1-\tanh\left(\parallel \mathbf{x_n}-\mathbf{x}\parallel^2\right)\right) \qquad (2)$$

where $\mathbf{x}$ is a D-dimensional vector and $\mathbf{w}$ a N-dimensional weight vector whose each entry w_n is associated with observation $\mathbf{x_n}$. The function $\rho_{\mathbf{w}}(\mathbf{x})$ takes maximal values for all $\mathbf{x}$ lying in regions with a high concentration of observations, and cancels when $\mathbf{x}$ is far enough from the data. In the context of probability distribution mixtures, these maxima correspond to the modes of each of the component of the mixture, which for unimodal symmetric probability densities such as gaussian distributions coincide with the mean value, provided that the components are separated enough so that their respective modes are still modes of the overall mixture. Hence, by finding all the maxima of the density function, one can in principle recover the centroids of all the clusters in the dataset. Since in general extracting the maxima of a density function such as $\rho_{\mathbf{w}}$ in one row is analytically impossible, we rather try to find them successively, so that a new cluster is extracted at each iteration of the algorithm. Computing the gradient of (2) with respect to $\mathbf{x}$, we find that the locations of the maxima obey an auto-coherent equation that we solve iteratively. Given an estimation $\mathbf{x^{(c)}}$ of the location of one of the maxima of $\rho_{\mathbf{w}}$ computed at the current iteration c of the maximization process, the updated estimation at the next iteration $c+1$ is given by

$$\mathbf{x}^{(c+1)}=\frac{\sum_{n=1}^{N}w_nC_n(\mathbf{x}^{(c)})\mathbf{x_n}}{\sum_{n=1}^{N}w_nC_n(\mathbf{x}^{(c)})} \qquad (3)$$

where $C_n(\mathbf{x})=1-\left(\tanh\left(\|\mathbf{x_n}-\mathbf{x}\|^2\right)\right)^2$. Since the variations of the density function only occur close to the data points, a random observation point is chosen as the initial estimation $\mathbf{x}^{(0)}$. The maximization step stops when the variations of $\mathbf{x}^{(c)}$ with respect to the iteration counter c are small enough, and the resulting vector, denoted by $\mathbf{x}^*$, defines the centroid of the new cluster to be expanded. As the starting point to find a cluster is to randomly pick a data point to initialize the maximization step, it is necessary to find the members of a given cluster as soon as its centroid has been extracted in such a way that all observations found to be part of this cluster can be removed from the possible initialization set for the next centroid detection step.

B. Elliptical-shaped classification

In an attempt to circumvent the tendency of density-based algorithms to merge entangled clusters, we constrain the classes generated during the expansion step to have ellipsoidal shapes. The motivation behind this choice is to encapsulate enough data points in a simple geometrical shape so that the cluster is well-represented, without assimilating points that would possibly belong to another class, hence giving the algorithm the ability to better distinguish entangled clusters. We thus need to specify both the orientation or principal axes $\{\mathbf{u_d}\}_{d\in\{1,\cdots,D\}}$ and the corresponding semi-axes $\{a_d\}_{d\in\{1,\cdots,D\}}$ to fully parametrize the cluster. In a D-dimensional space, $D-1$ orthonormal direction vectors are chosen, the last one being imposed by orthogonality. A possible choice for the first direction is the vector defined by

ISBN 978-0-9928626-7-1 © EURASIP 2017

the initial and final estimations of the centroid of the current cluster, *i.e.* $\mathbf{u}_1 = \frac{(\mathbf{x}^{(0)}-\mathbf{x}^*)}{\|\mathbf{x}^{(0)}-\mathbf{x}^*\|^2}$, so that if $D = 2$ the orientation of the ellipse is systematic. The extension of the algorithm in higher dimension would require additional arbitrary choices and is under study. Then, starting from the centroid $\mathbf{x}^*$ and for each previously defined direction $\mathbf{u_d}$, the corresponding semi-axis a_d is obtained by sequentially searching for the closest point in that direction for which either a minimum of the density function is found, revealing the presence of an entangled cluster, or a given threshold value is reached. This critical value, denoted by ρ_{th}, serves as a maximal extension that an isolated class can achieve, and is adaptative of the inner density of the cluster under consideration, which is related to the amplitude of its maximum, $\rho_{\mathbf{w}}(\mathbf{x}^*)$. We express this threshold value as $\rho_{th} = \rho_{\mathbf{w}}(\mathbf{x}^*)(1 - T_h)$, where T_h is an external parameter strictly lying between 0 and 1. T_h acts as a balance between the typical size of the ellipsoids found by the algorithm and its ability to split partially merged clusters. In our applications the value 0.7 has shown to be a good compromise, but specific applications where for instance a priori knowledge on the relative positions of the clusters is available would lead to other choices. We then express the semi-axis a_d as

$$a_d = \min\left(\sup(I), \inf\{t \in I \backslash \{0\} | \tilde{\rho}'_{\mathbf{w}}(t) = 0\}\right) \qquad (4)$$

where $I = \{t \in \mathbb{R}^+ | \tilde{\rho}_{\mathbf{w}}([-t \;\; t]) \subset [\rho_{th} \;\; \tilde{\rho}_{\mathbf{w}}(0)]\}$ and $\tilde{\rho}_{\mathbf{w}}(t) = \rho_{\mathbf{w}}(\mathbf{x}^* + t\,\mathbf{u_d})$. We finally define the cluster $\mathcal{C}$ as the set of data points lying inside the generated ellipsoid

$$\mathcal{C} = \{\mathbf{x_n} \in \mathcal{D} | (\mathbf{x_n} - \mathbf{x}^*)^T \mathbf{P}^T \mathbf{A}^{-2} \mathbf{P} (\mathbf{x_n} - \mathbf{x}^*) \leq 1\} \qquad (5)$$

where $\mathbf{P} = (\mathbf{u_1}, \cdots, \mathbf{u_D})$ is the orthogonal change of basis matrix induced by the orientation chosen for the ellipsoid, and $\mathbf{A}$ is the diagonal matrix whose entries are the semi-axes, $A_{ij} = a_i \delta_{ij}$ with δ_{ij} the Kronecker delta. The list of possible initialization points for the next centroid detection step is then updated by removal of the identified class members.

The weight vector $\mathbf{w}$ introduced in (2) allows to tune the relative contribution of each data point to the overall density. It is used to provide a higher weight to the clusters that have not yet been identified, which leads to a better detection of entangled classes. We then choose $w_n = \frac{1}{|\mathcal{C}_k|}$ if $\mathbf{x_n}$ belongs to the class $\mathcal{C}_k$, and $w_n = 1$ if $\mathbf{x_n}$ does not belong to any class yet. By convention, if a given point is found to belong to several classes, its weight is computed similarly based on the cluster with greater cardinality. The weight vector is updated as soon as the membership of the current cluster is established using (5), so that data points that have not been classified yet contribute more and more to the density profile. Adding this weight vector may however generate spurious clusters due to the remaining points neglected by the elliptical clustering in the class expansion process. Besides, as more clusters are discovered the set of unclassified points tends to get sparser, leading also to the apparition of few-membered, irrelevant classes of outlier observations. These two cumulative effects result in an overestimation of the number of clusters in the dataset. This is illustrated in Fig. 1 on a two-dimensional gaussian mixture with two components. The black and red ellipses on the top plot correspond to irrelevant and meaningful clusters, respectively. The bottom plot shows how the extraction of the right component and update of the weight vector allows both an easier detection of the second class and the emergence of excess clusters. A first solution to prevent the algorithm from generating residual clusters is to impose beforehand the ratio of points to be clustered, and use it as a stopping criterion for the whole classification procedure. We denote η this second parameter of the algorithm. Increasing η lowers the probability of missing a real class in the dataset, but at the cost of a greater amount of residual clusters and computation time. As this paper is utterly concerned with the accuracy of our method a value of 0.95 was chosen in all simulations.

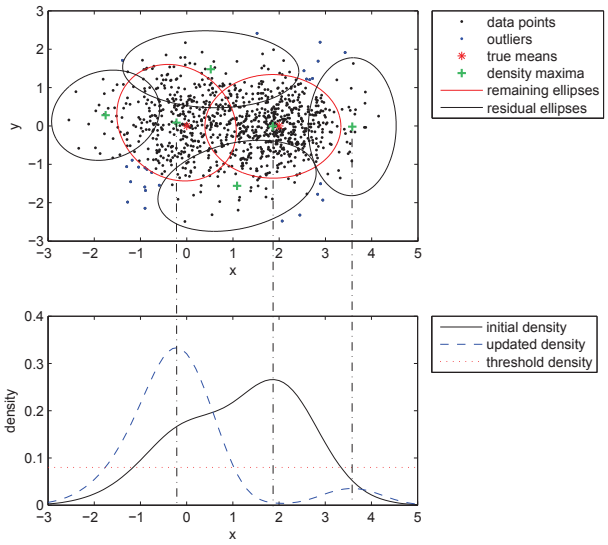

Fig. 1. Top : Elliptical classification of two entangled bi-dimensional gaussian distributions with weights $(0.5, 0.5)$, means $(0, 0)$, $(2, 0)$ and spherical covariance matrices 0.7 and 0.5, respectively. Bottom : Density profile $\rho_{\mathbf{w}}$ in the x direction at the initial state (black curve) and after the extraction of the fist cluster (blue curve) using the density threshold ρ_{T_h} (red line).

C. Post-processing of the elliptical clustering

The raw classification returned by our elliptical algorithm can be thought of as a covering of the dataset by K' ellipsoidal structures. A first immediate preliminary step to reduce the number of classes is to remove those clusters that only have a single member, and which actually correspond to outlier observations. Then, we enforce the number of real classes K as a contraint to further reduce the number of clusters. Among all pairs of clusters having a non-empty intersection, we select the one which achieves the maximum intersection relatively to the size of the smaller cluster.

$$(k_1, k_2) = \underset{\substack{k \in \{1, \cdots K'\} \\ k' > k}}{\operatorname{argmax}} \left(\frac{|\mathcal{C}_k \cap \mathcal{C}_{k'}|}{\min\left(|\mathcal{C}_k|, |\mathcal{C}_{k'}|\right)} \right) \qquad (6)$$

ISBN 978-0-9928626-7-1 © EURASIP 2017

The most irrelevant cluster $\mathcal{C}_{k^*}$ is then defined as the smaller one in the resulting pair

$$k^* = \operatorname{argmin}\left(|\mathcal{C}_{k_1}|, |\mathcal{C}_{k_2}|\right) \tag{7}$$

Finally, we remove $\mathcal{C}_{k^*}$ by redistributing its members among all clusters that share a non-empty intersection with it. Let $\mathcal{V}_{k^*} = \{k \in \{1, \cdots, K'\} \backslash \{k^*\} | \mathcal{C}_{k^*} \cap \mathcal{C}_k \neq \emptyset\}$ be the corresponding class labels, and $d_M(\mathbf{x_n}, \mathcal{C}_k)$ the Mahalanobis distance between observation $\mathbf{x_n}$ and class $\mathcal{C}_k$, defined as

$$d_M(\mathbf{x_n}, \mathcal{C}_k) = \left(\mathbf{x_n} - \mathbf{x_k}^*\right)^T \mathbf{\Sigma}_k^{-1} \left(\mathbf{x_n} - \mathbf{x_k}^*\right) \tag{8}$$

with $\mathbf{x_k}^*$ and $\mathbf{\Sigma}_k$ the centroid and covariance matrix of $\mathcal{C}_k$ respectively. We assign to each class $\mathcal{C}_k$ of $\mathcal{V}_{k^*}$ all data points of $\mathcal{C}_{k^*}$ for which the minimum of the Mahalanobis distance with respect to all other classes of $\mathcal{V}_{k^*}$ is achieved. We can thus write the update of cluster $\mathcal{C}_k$ as

$$\mathcal{C}_k \leftarrow \mathcal{C}_k \cup \{\mathbf{x_n} \in \mathcal{C}_{k^*} | \underset{k' \in \mathcal{V}_{k^*}}{\operatorname{argmin}}\, d_M(\mathbf{x_n}, \mathcal{C}_{k'}) = k\} \tag{9}$$

These two steps of detection of residual clusters and re-assigment of the corresponding data points is repeated until either $K' = K$, or there are no more non-empty intersection between any pair of clusters. Finally, a last optional phase consists in classifying the outlier observations to the remaining clusters replacing $\mathcal{C}_{k^*}$ and $\mathcal{V}_{k^*}$ by $\cap_{k=1}^{K'} \overline{\mathcal{C}_k}$ and $\{1, \cdots, K'\}$ in (9), respectively. An important point to mention is the fact that even after the above pruning phase, the algorithm may still overestimate the number of clusters. Moreover, if the raw classification of the algorithm is such that $K' < K$, no further processing is performed. The knowledge of K is then rather used as a guide to improve the bare elliptical classification than as an absolute requirement. As a matter of fact, in situations where the true number of classes is out of reach, a mere estimation of K may be enough for the algorithm to yield satisfying results.

D. Pseudo-code and additional comments

The overall proceeding of the elliptical clustering is summarized in Algorithm 1 as a pseudo-code. The post-processing phase can be considered to begin with the suppression of single-element clusters, but this step can also be included in the unconstrained version of the algorithm since it does not require the parameter K. The outliers classification step is optional as well. Because the observations used as the initial points for the maximization of $\rho_{\mathbf{w}}$ are chosen randomly, the resulting classification is non deterministic, and thus one may have to perform several runs in order to achieve the best possible clustering accessible by the algorithm.

Algorithm 1 Pseudo-code of the elliptical clustering algorithm

```
function elliptical_clustering(η, T_h, (K))
// Phase 1 : elliptical clustering of the data
K' ← 0, w ← (1, ··· , 1)
while | ∪_{k=1}^{K'} C_k| < N × η do
  choose x^(0) as a random x_n in ∩_{k=1}^{K'} C̄_k
  find x*_{K'+1} by iteratively maximizing ρ_w(x) using (3)
  for d = 1 ··· D do
    choose u_d orthogonal to {u_1, ··· u_{d-1}}
    compute a_d using (4)
  end for
  define C_{K'+1} using (5)
  update w and increment K'
end while
// Phase 2 : pruning of irrelevant clusters
add all single-element clusters to the set of outliers
while K < K' and ∪_{k=1}^{K'} V_k ≠ ∅ do
  select the cluster to remove using (6) and (7)
  reassign members of the removed cluster using (9)
  decrement K'
end while
// Phase 3 : classification of outliers
reassign outlier points using (9)
end
```

IV. Performance evaluation

A. Clustering validity metrics

The performance of the proposed algorithm is evaluated comparatively to an ideal, gold standard classification of the dataset using metrics based on the so-called confusion matrix. Given the target classification $\mathcal{C} = \{\mathcal{C}_k\}_{k \in \{1, \cdots K\}}$ and the clustering under evaluation $\mathcal{C}' = \{\mathcal{C'}_{k'}\}_{k' \in \{1, \cdots K'\}}$ the $K \times K'$ entries of the confusion matrix are defined as $m_{kk'} = |\mathcal{C}_k \cap \mathcal{C'}_{k'}|$. Starting from this general tool a wide variety of clustering evaluation criteria can be derived. For our performance tests we selected two indices, namely the normalized Mirkin index and the Variation of Information.

The Mirkin index [9] is a pair-counting based metric measuring the number of misclassified data points. It is given by

$$M_{\mathcal{C},\mathcal{C}'} = \sum_{k=1}^{K} |\mathcal{C}_k|^2 + \sum_{k'=1}^{K'} |\mathcal{C'}_{k'}|^2 - 2 \sum_{k=1}^{K} \sum_{k'=1}^{K'} |\mathcal{C}_k \cap \mathcal{C'}_{k'}|^2 \tag{10}$$

The resulting expression is always positive and null if the two classifications are equal. An additional normalization of the Mirkin index can be achieved in order to grant it the N-invariance property, meaning that the index is dependant on the relative proportions of the classes and not directly on the number of data points N. This normalized version is obtained by dividing the bare Mirkin index by N^2 [10].

The Variation of Information [10] is an information-based measure which evaluates the gain and loss of uncertainty about a given classification under the assumption that the other is known. Formally speaking, it is defined as

$$VI_{\mathcal{C},\mathcal{C}'} = H(\mathcal{C}|\mathcal{C}') + H(\mathcal{C}|\mathcal{C}') \tag{11}$$

where $H(\mathcal{C}|\mathcal{C}') = -\sum_{k=1}^{K} \sum_{k'=1}^{K'} \frac{m_{kk'}}{N} \log_2 \left(\frac{m_{kk'}}{m_{k\cdot}}\right)$ is the conditional entropy of classification $\mathcal{C}$ knowing classification $\mathcal{C}'$ and $m_{k\cdot} = \sum_{k'=1}^{K'} m_{kk'}$. The Variation of Information is zero if and only if both classifications are equal (up to a permutation in the class labels) and is naturally N-invariant.

ISBN 978-0-9928626-7-1 © EURASIP 2017

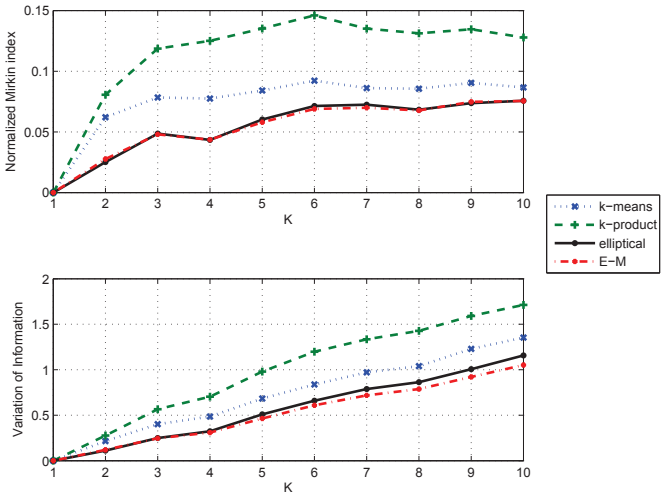

Fig. 2. Performance of the constrained elliptical clustering and several other clustering algorithms

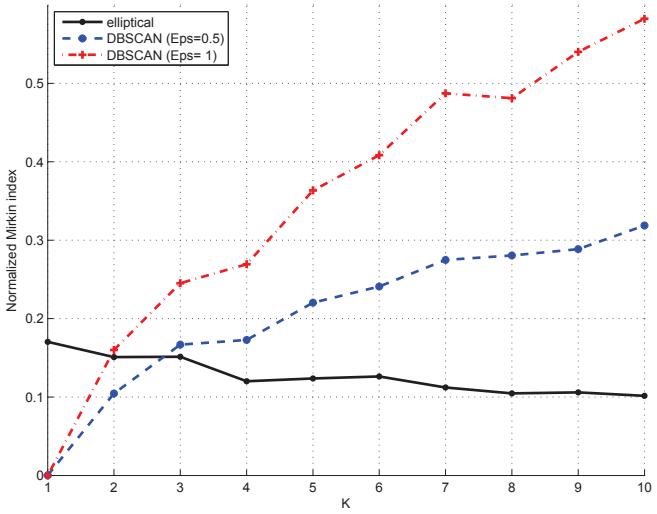

Fig. 3. Performance of the unconstrained elliptical clustering and DBSCAN with $MinPts = 5$

B. Simulation assumptions and results

To evaluate the quality of the classifications produced by our algorithm we have generated 200 random sets of 1000 two-dimensional sample points using gaussian mixture distributions for each value of the mixture component number K from 1 to 10. The weights, means and covariance matrices of each gaussian component are randomly chosen according to a uniform distribution for each independant parameter. The means range from -10 to 10 in each direction, and the maximum value for the covariances is set to 1. The indices are averaged over all datasets having the same number of classes.

Fig. 2 shows the performance curves for k-means, k-product, E-M and the constrained elliptical clustering. For these initialization-dependant algorithms the best result over 100 runs have been extracted before the averaging. Both plots tend to show that our algorithm provides better classification results than k-means and k-product. We also note how similar the performance curves of the elliptical and E-M algorithms are, though E-M seems slighty better. Since for gaussian mixtures the probability distribution components actually are of ellipsoidal shape, it is indeed expected that both approaches are particularly well-suited for this class of datasets.

Fig. 3 depicts the normalized Mirkin index obtained for the unconstrained elliptical algorithm and DBSCAN for two sets of its parameters Eps and $MinPts$. While DBSCAN tends to yield better clusterings for a small range of K values, it then considerably drifts in accuracy. This is because as the number of clusters increases while the observation space range is fixed, more and more entangled classes appear and DBSCAN considers them as single entities. The bare elliptical method on the opposite overestimates the number of classes but doing so carries more relevant information about the actual clustering of the data. This also accounts for the latter having a non-zero Mirkin index in the trivial case $K = 1$.

V. Conclusion

In this paper we have introduced a density-based clustering approach relying on ellipsoidal-shaped clusters to improve the detection of entangled classes. Our simulations have shown that our algorithm yields comparable performance with several standard algorithms in both its constrained and unconstrained versions. This preliminary work gives a first insight on the possibilities of our method and foresees future improvements as well as further comparisons on other types of mixtures and datasets with more advanced existing clustering techniques.

References

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering : A review," *ACM Computing Surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[2] L. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob., Vol. 1*. Univ. of Calif. Press, 1967, pp. 281–297.

[3] N. Paul, M. Terre, and L. Fety, "A global algorithm to estimate the expectations of the components of an observed univariate mixture," *Advances in Data Analysis and Classification*, vol. 1, no. 3, pp. 201–219, 2007.

[4] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*. AAAI Press, 1996, pp. 226–231.

[5] M. Ankerst, M. M. Breunig, H. P. Kriegeland, and J. Sander, "Optics: Ordering points to identify the clustering structure," in *Proc. ACM SIGMOD99 Int. Conf. on Management of Data*. ACM Press, 1999, pp. 49–60.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[7] J. C. Gower and G. J. S. Ross, "Minimum spanning trees and single linkage cluster analysis," *Journal of the Royal Statistical Society. Series C*, vol. 18, no. 1, pp. 54–64, 1969.

[8] R. Xu and I. D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[9] B. Mirkin, *Mathematical Classification and Clustering*. Kluwer Academic Press, 1996.

[10] M. Meilă, "Comparing clusterings by the variation of information," in *Learning Theory and Kernel Machines*, 2003, pp. 173–187.

ISBN 978-0-9928626-7-1 © EURASIP 2017