Journal of AI and Data Mining
Published online:

Extracting Prior Knowledge from Data Distribution to Migrate from Blind to Semi-Supervised Clustering

Z. Sedighi* and R. Boostani

Electrical & Computer Department, Shiraz University, Shiraz, Iran.

Received 19 November 2016; Revised 13 February 2017; Accepted 09 July 2017
**Corresponding author: sedighi.63@gmail.com (Z. Sedighi).*

Abstract
Although several works have been conducted to improve the clustering efficiency, most of the state-of-art schemes suffer from the lack of robustness and stability. This paper aims to propose an efficient approach to elicit a prior knowledge in terms of must-link and cannot-link from the estimated distribution of raw data to convert a blind clustering problem into a semi-supervised one. In order to estimate the density distribution of data, the Weibull Mixture Model is utilized due to its high flexibility. Another contribution of this work is to propose a new hill and valley seeking algorithm to find the constraints for a semi-supervised algorithm. The proposed valley-seeking algorithm does not require any user-defined parameter. It is assumed that each dominant density peak stands on a cluster center; therefore, the neighbor samples of each center are considered as the must-link samples, while the near-centroid samples belonging to different clusters are considered as the cannot-link ones. The proposed approach is applied to a standard image dataset (designed for clustering evaluation) of Berkeley University along with some UCI datasets. The results achieved on both databases demonstrate the superiority of the proposed method compared to the conventional clustering ones.

Keywords: *Semi-supervised, Clustering, Valley-seeking Scheme, Weibull Mixture Model.*

1. Introduction

Clustering techniques are used in a vast variety of data mining applications such as stream mining [15], image segmentation (clustering)[13], multi-objective systems [21], and spam filtering [9]. The conventional strategies of cluster forming are hierarchical [11], flat [17], graph-based [8], and density-based [14]. Each category of the mentioned methods has its own drawbacks. For instance, incorporating the tree structure into the clustering has led to the development of hierarchical clustering algorithms such as divisive and agglomerative (e.g. single and complete linkage) [5, 22], whereas hierarchical algorithms are faced with some challenges such as selecting an effective termination criterion, lack of back-tracking, and heavy computational burden.

Despite the simplicity of the flat clustering methods such as *K*-means [10], they still suffer from the lack of learning stability due to high sensitivity of their performance to their initial cluster centers. Graph-based algorithms like shared nearest neighbor (SNN) [6] prune a considerable number of instances as noisy samples, and do not assign them to any cluster. Density-based clustering methods try to form clusters in the directions of dense regions [20]. Nevertheless, among the mentioned cluster forming algorithms, density-based methods have attracted much attention since they try to make a relation among the samples in the dense regions and then consider each set of connected samples as a cluster. Although each density-based algorithm has its own shortcomings, this approach is more consistent with the nature of data.

In the case of having a prior knowledge about a part of our samples in terms of must-link and cannot-link connections, the problem of blind clustering is converted into semi-supervised clustering. Extracting such constraints require the confirmation of experts, which is an expensive and time-consuming process; consequently, it is a big deal to convert an unsupervised problem into a

semi-supervised one, and the conventional methods fail to make this conversion for a wide range of problems. The main contribution of this work is to elicit a prior knowledge from the nature of data and then convert a blind clustering problem into a semi-supervised one.

1.1 Literature review

To the best of the authors' knowledge, there is no research work similar to the method proposed in this paper. Nevertheless, among different strategies of clustering (e.g. hierarchical, flat, graph-based, and density-based), it can be said that the family of density-based clustering algorithms has the highest degree of similarity to the proposed approach. Therefore, the density-based clustering techniques are introduced here and their pros and cons are analyzed.

Density-based Spatial Clustering of Applications with Noise (DBSCAN) [7] is the most famous algorithm among the density-based clustering methods. In fact, other density-based schemes are known as different derivations of DBSCAN. This method allows us to form clusters with arbitrary shapes in the directions of dense regions. The strongest property of DBSCAN is its low sensitivity to noisy and outlier samples. Also the complexity of this algorithm is quite low due to doing just one time scanning for each point; consequently, DBSCAN is suitable for handling large datasets (big data), and is vastly applied to the data mining applications. Nevertheless, the main flaw of DBSCAN is its high sensitivity of its user-defined parameters. Moreover, this method is not capable of detecting the gradient of density within a cluster. In order to overcome this deficiency, distributed DBSCAN (DDBSCAN) is proposed to detect clusters that are in hierarchy or clusters with different densities separately [1]. In contrast to DBSCAN, DDBSCAN still suffers from a high computational complexity and is sensitive to the user-defined parameters.

Ordering Points to Identify the Clustering Structure (OPTICS) is an efficient algorithm that can be considered as an extension of the DB-Scan method in which all instances are evaluated one by one, and the suitable radius along with the nearest core point of each instance is determined [2]. In this way, all user-defined parameters for each instance are adaptively determined, and finally, an extended DB-Scan is run over the processed data. Therefore, solving the problem of parameter dependency in DBSCAN is one of the advantages of the OPTICS algorithm. Nevertheless, OPTICS cannot guarantee to find the optimum radius for each point, and finds a suitable radius for all instances.

Density-based clustering (DenClue) is another density-based clustering method, which first tries to find the whole distribution of data by finding the local distribution of samples using an influence function [12]. The estimated distribution of DenClue is better than that of DBSCAN in terms of quality since it locally estimates a certain kernel for each sub-space, while DBSCAN starts with a certain radius that definitely is not optimal for all regions. After computing the gradient of the estimated kernel density functions to find the density attractors, a hill-climbing algorithm tries to group the samples that are located in the vicinity of each density attractor. The bottleneck of this method appears when the dimension of data increases.

The main contribution of this work is to extract a prior knowledge in terms of the must-link and cannot-link constraints by estimating the data distribution [24]. Here, Weibull Mixture Model (WMM) was chosen to estimate the data distribution due to its flexibility to model each arbitrary cluster with a low number of Weibull functions [19]. Next, the distribution is partitioned into primary clusters by proposing an efficient valley-seeking algorithm. Consequently, the constraints are extracted from the primary clusters by recognizing the must-link samples as the samples located around the distribution of each hill (center of each cluster) and the cannot-link samples as the must-link samples of different clusters. By this trick, the blind clustering problem is automatically converted into a semi-supervised one. Finally, the most proper number of clusters is found according to the best Silhouette score. Although a few similar works have been carried out for clustering using WMM [16, 18], their valley-seeking algorithms require a few user-defined parameters that cannot be automatically found from the data, whereas in the proposed method, the suggested valley-seeking scheme does not require any parameter, and it is fully automated.

The rest of this paper is structured as what follows. Section 2 explains the details of the compared clustering methods, and next, the proposed approach is presented. Section 3 introduces the evaluation methods and expresses the datasets. In Section 4, the experimental results produced by each one of the methods are separately presented, and the benefits and shortcomings of each scheme are discussed. Finally, in Section 5, the paper is concluded and,

at the end, a new horizon to the future works is presented.

2. Methods

In this part, our objective is to introduce the implementation details of the following algorithms: K-Means, DB-SCAN, OPTICS, DenClue, Single-Linkage, Complete-Linkage, and SNN. As we see, in addition to the density-based clustering methods, flat, hierarchical, and graph-based clustering methods are explained in this work. Next, the proposed method is expressed using WMM as the distribution estimator and a new valley-seeking algorithm to determine the clusters.

2.1. *K*-Means

K-means is the most famous and effective flat clustering scheme that has been utilized in many applications. This method is randomly initialized by a certain number of cluster centers (*K*) defined by the user. Then the samples are assigned to the nearest cluster center. At each step, each cluster center is updated according to the cluster samples, and the points are again assigned to the new centers. This process continues until changes of the clusters' centers do not exceed a pre-defined threshold in two successive iterations.

2.2. DB-SCAN

The procedure of DBSCAN clustering algorithm can be explained as what follows. At first, it randomly selects a point (*p*) and considers this point as the center of a circle with radius *Eps*. The algorithm checks whether at least the *MinPts* number of samples are located in that circle or not. If the answer is yes, this point is considered as a core point; in contrast, the neighbor samples are checked. *Eps* and *MinPts* are user-defined parameters but in most papers, the value for *MinPts* has been set to 4. The neighbors that are placed within the circle of each core point are called direct reachable, and those indirectly connected to this point are called indirect reachable. Next, the algorithm evaluates a new point of data if there is no density-connected point from previous core points and repeats the above steps until all points are processed.

2.3. OPTICS

The OPTICs algorithm is known as an extension of the DBSCAN algorithm, which tries to automatically optimize its parameters [2]. The main components of this algorithm include the core distance and the reachable distance. OPTICS finds the neighbors of each sample with different radiuses and compares the number of neighbors for each radius to the *MinPts* parameter. Similar to DBSCAN, the samples that can construct a cluster regarding the *MinPts* parameter are selected as the core points.

If the graph of radius/samples is drawn, the core point samples and their corresponding radius are determined [2]. Each core point with its connected neighbors regarding the adjusted reachable distance is considered as a cluster. In other words, OPTICS adaptively finds the radius for different regions, and each cluster is grown from the densest region of that cluster. This algorithm ensures us to find clusters with different densities. Moreover, OPTICS is sensitive to the density gradient within each cluster and divides the cluster into clusters with uniform density.

2.4. DenClue

Among the density-based clustering algorithms, DenClue is the most similar approach to the proposed method in this paper. DenClue has two important phases. In the first phase, the density function is estimated in terms of summation of influence functions. In the second one, each cluster is characterized as the samples located around a local maximum point of the overall probability density function. In the case of using the continuous influence function, the overall density function is continuous at each point, and the density attractors (clusters) can be derived by a hill-climbing method taking a gradient from the overall density function. The influence function, summation of influence functions to construct the density, and the gradient of the whole density function are presented in (1), (2) and (3), respectively.

$$f_{Gaussian}(x, x_i) = e^{-\frac{d(x,x_i)^2}{2\delta^2}} \tag{1}$$

$$f^{D}_{Gaussian}(x) = \sum_{i=1}^{N} e^{-\frac{d(x,x_i)^2}{2\delta^2}} \tag{2}$$

$$\nabla f^{D}_{Gaussian}(x) = \sum_{i=1}^{N} \frac{d(x, x_i)}{\delta^2} e^{-\frac{d(x,x_i)^2}{2\delta^2}} \tag{3}$$

where, $d(x,x_i)$ is the Euclidian distance between x and x_i, δ is the variance of each Gaussian (influence) function, and N is the number of influence functions to construct the density. As far as the noisy and outlier points are located in low-dense regions, these samples have a very low effect on the whole performance.

2.5. Shared Nearest Neighbor (SNN)

Most challenges occur when the clustering methods are faced with groups of samples with different population, different densities, different shapes or dataset with noisy and outlier values. The SNN method tries to deal with all the mentioned problems. The algorithm first finds the nearest neighbors of each data point, and then redefines the similarity between two points using the number of nearest neighbor points that are common between the two points as the edge weight that connects these two points in the graph. Using this new definition of similarity, SNN prunes the noise and outlier samples because they do not connect to any point. In contrast, SNN identifies core points and then creates clusters around the cores. These clusters do not contain all points but finely represent different sets of connected points as clusters.

2.6. Standard semi-supervised clustering

In the case of having no prior knowledge, the clustering becomes an unsupervised process, while in some applications, there is little information available about a subset of samples. The problem of clustering a set of samples when prior knowledge (in terms of the must-link and cannot-link samples) is available about subsets of samples is called the semi-supervised clustering method. The must-link and cannot-link samples are normally determined by experts. Since the cost of labeling or finding the hidden constraints is very high, just the constraints among a small subset of samples are determined [24]. Recent studies have shown that when these limitations (must-link and cannot-link samples) are fed to the clustering process, the accuracy of clustering is significantly increased.

2.7. Proposed algorithm

In spite of using the summation of Gaussian functions to estimate the distribution of data (e.g. GMM), here, the Weibull functions are employed, which can incorporate a degree of skewness to the components. It is obvious that by incorporating the shape parameter to the Weibull functions, we can build a complex distribution with a lower number of Weibull functions compared to the Gaussian functions. In the following, first the Weibull function is introduced, and then the Weibull Mixture Model (WMM) is explained, and finally, the proposed algorithm is expressed.

2.7.1. Weibull distribution

Weibull distribution is one of the most flexible distributions in statistics, which can be adapted to the distribution of data with a few data points. This function is more flexible than the Gaussian function because its parameters allow the Weibull shape to become asymmetric. In other words, it has a shape parameter that regulates the skewness of the function toward the left or right direction. The Weibull distribution is defined as follows:

$$f(x)=\begin{cases}\frac{\beta}{\alpha}\left(\frac{x-L}{\alpha}\right)^{\beta-1}e^{-\left(\frac{x-L}{\alpha}\right)} & x\geq L\\ 0 & otherwise\end{cases} \tag{4}$$

where, $\alpha, \beta,$ and L are the scale, shape, and location parameters, respectively. For this distribution, $the\ \alpha\ and\ \beta$ parameters should be positive. L is often assumed to be zero, and in this case, the three-parameter Weibull distribution becomes a two-variable function that is more common. In the case of L = 0 and $\beta=1$, this distribution forms a shape similar to an exponential distribution. Similarly, when $\beta=3.25$, the distribution is almost as glaring as the normal distribution.

2.7.2. Weibull mixture model (WMM)

Murthy *et al.* [18] have introduced the multivariate Weibull mixture model, in which several Weibull functions are linearly summed by different weights in order to model each complex arbitrary distribution. The WMM model is composed as follows:

$$y_i=\sum_{j=1}^{m}a_{ij}y_{ij} \tag{5}$$

where, y_{ij} is the j^{th} Weibull distribution with parameters α_{ij}, β_{ij}, and a_{ij} determining the weight of the j^{th} Weibull function in the mixture model. The multivariable p dimensional Weibull function is described in (6):

$$R(x)=\prod_{i=1}^{p}\sum_{j=1}^{m}a_{ij}\exp\left\{-(\beta_{ij}x_i^{a_{ij}}+\frac{\beta x_0}{p})\right\} \tag{6}$$

where, x_0 = Max(x_1, ..., x_p) > 0. One of the methods used to find the final distribution, similar to determining the kernel density estimation, is to convolve each sample like Dirac delta function form to the multivariate Weibull distribution function of the observed data, which is explained as follows:

$$p_{KDE}(x)=R_{s-1}(x)\times p_s(x)=\sum_{i=1}^{N}\alpha_i R_s(x-x_i) \tag{7}$$

where, $p_s(x)$ is the Weibull distribution of a new example, $R_{(s-1)}(x)$ is the Weibull kernel for the observed data, and $\hat{p}_{KDE}(x)$ is the approximate

kernel density estimation of the whole distribution at point *x*.

2.7.3. Gap statistic

In order to find the correct number of Weibull functions in WMM, the gap statistic method [19], as a well-known manner of estimating the number of clusters, is employed. Consider a *d*-dimensional dataset with n independent observations. Let $d(x,y)$ be the Euclidean distance between two observations x and y. Assume that the data is categorized in k different clusters $C_1, ...,C_k$, and $n_r = |C_r|$ is the number of data belonging to the cluster r. The average distance of samples within the r^{th} cluster is denoted as D_r, which is determined as follows:

$$D_r = \sum_{x,y \in C_r} \frac{d(x,y)}{2n_r} \tag{8}$$

and, w_k is the summation of the within class average distances of all k clusters:

$$w_k = \sum_{r=1}^{k} D_r \tag{9}$$

The main idea of the gap statistic method [19] is the standardization of graph $log(w_k)$ by comparing its expectation under a suitable null reference distribution of data. The optimal number of clusters is the k at which w_k is the farthest point under this reference curve:

$$Gap_n(K) = E\{\log w_k\} \tag{10}$$

where, $E(.)$ is the expectation of examples with size n from a reference distribution. The estimated value of k is the value that maximizes (10).

2.7.4. Eliciting constraints from density function

After training WMM with the right number of Weibull functions determined by gap statistics, the clusters should be extracted from the estimated distribution. Local maxima of the density function can be easily obtained by taking a gradient of that density. The points placed below each local hill are considered as the must-link points. Since these points are located on the densest region of the distribution, we can consider these neighbor points belonging to a cluster, and consider them as the must-link samples.

One of the questions in the described algorithm is that how many samples around each density mean (below the density hill) should be selected as the must-link samples. In order to answer this question, some parameters are defined. Let n_0 and N_0 be the number of samples in the clusters with the lowest and highest population, respectively. Accordingly, the following relation describes the worst portion of populations among the clusters:

$$I = \frac{n_0}{N_0} \tag{11}$$

The number of selected samples as must-link samples around the center of each cluster is determined as follows:

$$k_{\min} = \frac{n_c}{k} \times I \tag{12}$$

where, n_c is the total number of samples in the dataset and k is the number of clusters.

2.7.5. Eliciting clusters from estimated density

Similar to the most clustering algorithms, here, small-size clusters are eliminated, and their samples are assigned to the other clusters; therefore, to avoid the density peak of clusters being close together, the distance between two hills should exceed a threshold. In other words, within a large cluster, the distribution function might fluctuate; consequently, each local peak should not be considered as a center for a new cluster. Therefore, just those density hills can be considered as the center of clusters that have a significant distance to each other; in addition, each cluster center should be the maximum hill in its vicinity.

In order to find the number of clusters, after calculating the approximate number of clusters (k) using the gap statistic, the real number of clusters is considered in the interval [k/2, 2k], similar to the Iso-Data clustering algorithm [3]. After determining the constraints (described in the former part), the semi-supervised clustering algorithm is executed, and the clustering index is determined to assess how well the clusters are formed.

The Silhouette method is one of the famous clustering validation methods that evaluates a cluster according to the score of its samples [4]. In other words, it gives a score to each sample, which measures the belongingness of that sample to the located cluster compared to that of the other clusters.

For each data sample i, let $a(i)$ be its average dissimilarity to the other samples in that cluster and $b(i)$ be the lowest average dissimilarity between this sample to the other clusters. The silhouette score of the i^{th} sample, denoted as $s(i)$, is defined as follows:

$$s(i)=\begin{cases}1-a(i)/b(i) & if\ a(i)\langle b(i)\\ 0 & if\ a(i)=b(i)\\ \dfrac{b(i)}{a(i)}-1 & if\ a(i)\rangle b(i)\end{cases} \quad (13)$$

From the above equation, it is clear that s(i) is limited in the interval of [-1,1]. As *s(i)* gets close to one, the proper silhouette score is achieved. For each cluster, the average of all silhouette scores of its samples measures the validity of that cluster. We used the summation of silhouette scores of all clusters as the goodness of the number of achieved clusters. The number of clusters that provides the highest silhouette score is determined as the best number of clusters. In order to clarify different stages of the proposed method, the following pseudo-code is presented in figure 1.

Set the initial number of clusters to k/2
For c = k/2:1:2k

- *Hills are obtained by taking a gradient from the estimated density.*
- *These points are then sorted in an ascending order.*
- *The first c numbers of them are selected.*
- *Set the threshold distance as the average distances of hills.*
- *The must-link samples are chosen as the set of k_{min} samples around each cluster center.*
- *The semi-supervised clustering algorithm is executed, and the clustering index is determined.*

End

The best number of clusters is the one that maximizes the average silhouette score.

Figure 1. Pseudo-code of proposed algorithm.

3. Datasets and evaluation methods

In this section, at first, the datasets employed are described, and then the evaluation methods are introduced to assess the proposed method in comparison with the other implemented methods. In this work, both the labeled and unlabeled data are used to assess the clustering methods.

One way to validate the clustering method is to execute it on a standard dataset, in which the clusters are known as a priori; therefore, we can measure the clustering error. Here, a standard image dataset prepared in Berkeley University is used to assess the methods.

Here, some of the datasets in the UCI machine learning database are used to evaluate the compared methods. The selected datasets cover all the possible cases in terms of high and low input dimensions including noise and clean instances and different numbers of classes (here clusters). Important features associated with these datasets are shown in table 1.

Table 1. Description of selected UCI datasets in terms of number of instances, dimensions, and classes.

Data sets	#Instances	#dimensions	#clusters
Iris	150	4	3
Bupa	345	6	2
Vehicle	846	18	4
Breast-cancer	286	9	2
Glass	214	9	6

4. Experimental results and discussion

In this section, the results of applying the proposed methods (described in Section 2) to the data (described in Section 3) are presented. In order to show the suitability of WMM, the proposed method is executed over the data distribution estimated by both GMM and WMM. The results obtained are presented in two sub-sections; Case#1 demonstrates the results on the selected UCI datasets and Case#2 exhibits the results on the image dataset. It must be noted that there are two approaches for evaluating a clustering method. The first approach uses a labeled data that is blindly (without label) applied to a clustering method such as that through the clustering learning. It means that learning of the clustering method is carried out without the use of data labels. In the second approach, the input data does not contain any label, and when the clustering algorithm groups the samples into clusters, the goodness of the algorithm is assessed using some criteria like mean square error, discriminability among the clusters via distance.

In this work, the first approach is utilized, in which after blind clustering, we can precisely determine the purity of the clusters as the clustering accuracy.

4.1. Case #1

The results observed in table 2 illustrate the accuracy of the clustering methods on the five selected UCI datasets (described in Table 1). After applying the silhouette method to select the best number of clusters, the clustering accuracies for different schemes are demonstrated in table 2.

As we can see, the proposed method by GMM and WMM provides significantly a much higher clustering accuracy compared to the other state-of-the-art clustering methods. By employing the same number of core functions (Gaussian and Weibull) to estimate the density of each dataset, the proposed method using WMM produces

slightly better results than those of GMM. This supremacy implies the significance of WMM compared to GMM.

The number of employed core functions for each dataset is chosen through the cross-validation phase such that the selected number of core functions provides the highest value for the expectation maximization (EM). The selected number of core functions for the Iris, Bupa, Vehicle, Breast-Cancer, and Glass are 3, 2, 3, 2, and 5, respectively. One can say that the number of core functions by GMM and WMM is not necessarily equal; it is right but the selected number of core functions for GMM and WMM for each dataset is considered the minimum number core function selected by GMM and WMM. The reason of supremacy of WMM to GMM rises from the high capability of the Weibull function in moto estimate each arbitrary shape. Although when the number of Gaussian functions increases, they are able to model arbitrary shapes but in the case of a limited number of core functions, the Weibull function performs better than the Gaussian function.

Table 2. Clustering accuracies (in %) of compared clustering methods on selected UCI datasets.

Data sets	K-means	DBSCAN	OPTICS	DenClue	SNN	Proposed method by GMM	Presented method by WMM
Iris	86.7	68.7	69.6	72.0	65.4	90.5	95.1
Bupa	40.1	50.6	53.3	65.1	46.8	69.1	77.0
Vehicle	23.8	52.9	54.6	63.1	35.2	63.2	75.6
Breast-Cancer	62.4	65.5	66.2	68.8	50.9	70.1	79.5
Glass	56.1	35.3	38.7	45.1	20.5	56.7	64.1

The bar chart shown in figure 2 graphically shows the drastic superiority of the proposed method compared to the other implemented methods. In addition, the T-test is executed on several runs of the clustering algorithms, and the results of the proposed method are significantly ($P < 0.05$) superior to the counterparts.

It should be mentioned that like the Gaussian distribution, the Weibull distribution is also able to fit on a symmetric shape; but when the data distribution is crooked to the left or right, the Weibull function can easily be adapted to this asymmetry, while one Gaussian cannot be lonely fitted on a skewed distribution with a high accuracy; therefore, several Gaussian functions need to be added for modeling such data distribution.

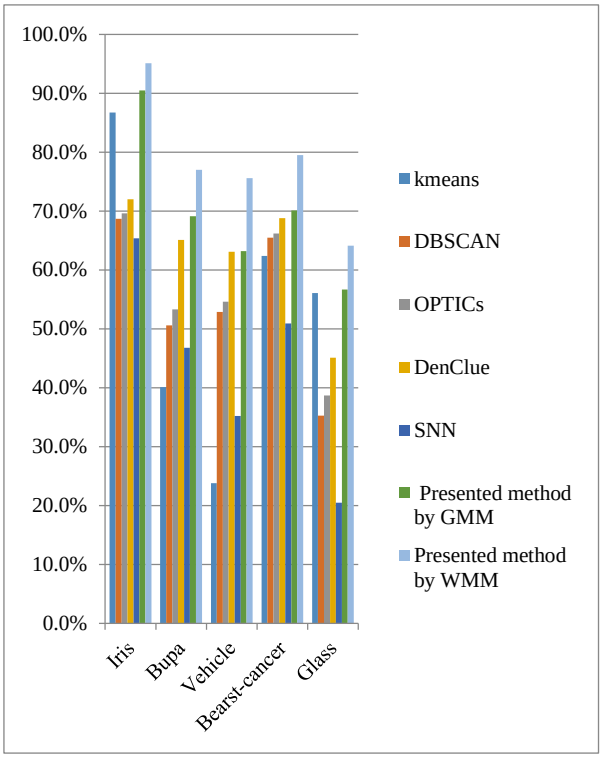

Figure 2. Clustering accuracy of compared clustering methods on UCI datasets.

Since the distribution of the input data is unknown in practice, using the Weibull functions enables us to deal better with the unknown data and finely arrange the samples in different clusters with arbitrary shapes. In addition, modeling each cluster with a very low number of Weibull functions provides good interpretability to describe the structure of data.

4.2. Case #2

In this part, the intensity values within each image are clustered (segmented) into uniform areas in which each area (cluster) contains the pixels with fairly similar intensity values. After applying each one of the clustering methods to the images, the segmented areas can be compared to the correct information in the dataset in order to determine the accuracy of each clustering method. The average clustering accuracies over the images for the mentioned clustering methods are represented in table 3. As we can see, the clustering accuracy of the proposed method is significantly higher than the other compared methods.

Table 3. Clustering accuracies (in %) of compared clustering methods on image clustering dataset.

Image number	K-means	DBSCAN	OPTICS	DenClue	SNN	Proposed method by GMM	Presented method by WMM
#8068	38.5	40.3	42.1	72.3	70.3	66	88.3
#3063	60.3	60.8	63.3	55.1	53.6	63	68.5
#6064	19.6	20.2	23.7	28.7	25.6	30	47.4

Figure 3 shows the segmented areas (clusters) for the image #8068 in the dataset by the implemented clustering methods.

Figure 3. Segmentation results obtained from SNN (Left-top), DBSCAN (Right-top), GMM (Left-down), and WMM (Right-down).

As it can be observed, the SNN algorithm could not correctly segment the border points of the clusters. The reason comes back to this fact that the boarder points are considered as noisy samples, and are not assigned to any cluster. Incidentally, the DBSCAN results are not interesting; this deficiency comes back to this reality that different image segments are not very uniform, and the gradient of pixel intensities within each cluster is noticeable. Since DBSCAN considers a certain radius for all of the space, it cannot finely segment the areas that are in hierarchy. The proposed method using GMM and WMM provides better performance than the others but the segmented areas by WMM are obviously more accurate than those of GMM. This superiority was statistically proved ($P < 0.05$). Nevertheless, the proposed method using WMM could not cluster the beak and shadow of the swan.

As mentioned in sub-section 4.1., the number of employed core functions for each image is chosen through the cross-validation phase such that the selected number of core function resulted in a higher accuracy. The selected number core functions for images #8086, #3063, and #6064 were selected to be 3, 5, and 4, respectively.

4.3. Computational complexity

Since the computational complexity of WMM and Gaussian mixture model (GMM) is similar, expect one more learning parameter that WMM has compared to GMM, here, the computational complexity of GMM was determined. The complexity of GMM is O(kn), where n is the size of the dataset and k is the number of mixtures [23].

5. Conclusion

Clustering methods are encountered with some challenges such as validation of clusters, finding a proper number of clusters, measuring the accuracy of clusters (e.g. purity), limitation on the supremum and infimum number of samples within a cluster, and maximum variance of each cluster. In this work, we proposed a novel technique to automatically elicit the constraints from the estimated density of data in order to convert a blind clustering problem into the semi-supervised problem. Since performance of semi-supervised clustering techniques is higher than a blind one, the proposed scheme can drastically improve the clustering performance for real applications. The proposed technique is general and does not require any prior knowledge for its valley seeking part. The results achieved on two datasets demonstrated that the proposed model provided much higher results on two categories of datasets compared to the state-of-the-art methods in terms of clustering accuracy.

References

[1] Ali, T. & Asghar, S., et al. (2010). Critical analysis of dbscan variations, in Information and Emerging Technologies (ICIET), 2010 International Conference on, pp. 1-6.

[2] Ankerst, M. & Breunig, M. M., et al. (1999). OPTICS: ordering points to identify the clustering structure, in ACM Sigmod Record, pp. 49-60.

[3] Ball, G. H. & Hall, D. J. (1965). ISODATA, a novel method of data analysis and pattern classification, DTIC Document.

[4] de Amorim, R. C. & Hennig, C. (2015). Recovering the number of clusters in data sets with noise features using feature rescaling factors, Information Sciences, vol. 324, pp. 126-145.

[5] Defays, D. (1977). An efficient algorithm for a complete link method, The Computer Journal, vol. 20, pp. 364-366.

[6] Ertöz, L. & Steinbach, M., et al. (2004). Finding topics in collections of documents: A shared nearest neighbor approach, in Clustering and Information Retrieval, ed: Springer, pp. 83-103.

[7] Ester, M. & Kriegel, H.-P., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise, in Kdd, pp. 226-231.

[8] Flake, G. W. & Tarjan, R. E., et al. (2004). Graph clustering and minimum cut trees, Internet Mathematics, vol. 1, pp. 385-408.

[9] Halder, S. & Tiwari, R., et al. (2011). Information extraction from spam emails using stylistic and semantic features to identify spammers, in Information Reuse and Integration (IRI), 2011 IEEE International Conference on, pp. 104-107.

[10] Hartigan, J. A. & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm, Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 28, pp. 100-108.

[11] Hastie, T. & Tibshirani, R., et al. (2009). "Unsupervised learning," in The elements of statistical learning, ed: Springer, pp. 485-585.

[12] Hinneburg, A. & Keim, D. A. (1998). An efficient approach to clustering in large multimedia databases with noise, in KDD, pp. 58-65.

[13] Jain, A. K., Murty, M. N., et al. (1999). "Data clustering: a review," ACM computing surveys (CSUR), vol. 31, pp. 264-323.

[14] Kriegel, H. P. & Kröger, P., et al. (2011). "Density- based clustering, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 1, pp. 231-240.

[15] Lu, Y.-H. & Huang, Y. (2005). Mining data streams using clustering, in 2005 International Conference on Machine Learning and Cybernetics, pp. 2079-2083.

[16] Mair, P. & Hudec, M. (2009). Multivariate Weibull mixtures with proportional hazard restrictions for dwell- time- based session clustering with incomplete data, Journal of the Royal Statistical Society: Series C (Applied Statistics), vol. 58, pp. 619-639.

[17] Manning, C. D. & Raghavan, P., et al. (2008). "Scoring, term weighting and the vector space model," Introduction to Information Retrieval, vol. 100, pp. 2-4.

[18] McNicholas, P. D. (2011). On model-based clustering, classification, and discriminant analysis, Journal of the Iranian Statistical Society, vol. 10, pp. 181-190.

[19] Murthy, D. P. & Xie, M., et al. (2004). Weibull models vol. 505: John Wiley & Sons.

[20] Sclove, S. L. (1977). Population mixture models and clustering algorithms, Communications in Statistics-Theory and Methods, vol. 6, pp. 417-434.

[21] Shahsamandi Esfahani, P. & Saghaei, A. (2017). "A Multi-Objective Approach to Fuzzy Clustering using ITLBO Algorithm," Journal of AI and Data Mining, vol. 5, pp. 307-317.

[22] Sibson, R. (1973). SLINK: an optimally efficient algorithm for the single-link cluster method, The computer journal, vol. 16, pp. 30-34.

[23] Verbeek, J. J. & Vlassis, N., et al. (2003). Efficient greedy learning of Gaussian mixture models, Neural computation, vol. 15, pp. 469-485.

[24] Wagstaff, K. & Cardie, C. (2000). Clustering with instance-level constraints, AAAI/IAAI, vol. 1097.

نشریه هوش مصنوعی و داده کاوی

روشی نوین جهت تبدیل یک مسئله خوشه‌بندی به خوشه‌بندی نیمه سرپرست و بهبود کارایی آن

زینب صدیقی* و رضا بوستانی

گروه برق و کامپیوتر، دانشگاه شیراز، شیراز، ایران.

ارسال ۲۰۱۶/۱۱/۱۹؛ بازنگری ۲۰۱۷/۰۲/۱۳؛ پذیرش ۲۰۱۷/۰۷/۰۹

چکیده:

خوشه‌بندی یک دسته‌بندی بدون نظارت از نمونه‌ها به گروه‌ها می‌باشد. مسئله خوشه‌بندی در زمینه‌های مختلفی توسط محققین به طرق مختلف مطرح شده و این مسئله بیانگر این است که خوشه بندی یکی از روش‌های مهم و اساسی در داده کاوی می‌باشد. هدف این مقاله، استفاده از روش‌های خوشه‌بندی نیمه نظارت جهت بهبود کارایی روش‌های خوشه‌بندی سنتی و افزایش دقت خوشه‌بندی آن‌هاست. این هدف از طریق استخراج اطلاعات داده‌ها بوسیله روش‌های آماری در خوشه‌بندی بدون نظارت و سپس ارائه این اطلاعات به عنوان دانش اولیه به روش‌های نیمه نظارت تحقق می‌پذیرد. بررسی تحقیقات انجام شده نشان از قدرت و کاربرد فراوان روش‌های خوشه‌بندی نیمه نظارت در گروه‌بندی و نتایج دقیق و علمی حاصل از آن‌ها دارد. برخی روش‌های خوشه‌بندی متداول و روش تبدیل مسائل بدون نظارت به خوشه‌بندی نیمه نظارت بعنوان روش مورد استفاده به طور دقیق بررسی شده است. روش پیشنهادی این مقاله در دو مرحله، ابتدا بر روی مجموعه داده‌ی مصنوعی و سپس بر روی برخی از داده‌های موجود در مجموعه داده‌های یادگیری ماشین UCI پیاده‌سازی و عملکرد آن مورد بررسی قرار گرفته است. نمودار میله‌ای حاوی اطلاعاتی در مورد دقت و کارایی و مقایسه روش‌های خوشه‌بندی گذشته و روش پیشنهادی این مقاله مورد بحث و بررسی قرار گرفته که نشان دهنده این است که روش پیشنهادی این مقاله در مقایسه با سایر روش‌های متداول موجود از سرعت و دقت بهتری برخوردار است.

کلمات کلیدی: خوشه‌بندی، روش‌های نیمه نظارت، خوشه‌بندی مبتنی بر کرنل، خوشه‌بندی مبتنی بر کرنل وایبل.