Expert Systems with Applications 161 (2020) 113690

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A density-based approach for querying informative constraints for clustering

Ahmad Ali Abin [a,*], Viet-Vu Vu [b]

[a] Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran
[b] VNU Information Technology Institute, Vietnam National University, Hanoi, Viet Nam

ARTICLE INFO

Article history:
Received 20 August 2019
Revised 20 April 2020
Accepted 22 June 2020
Available online 30 June 2020

Keywords:
Constrained clustering
Density tracking
Must-link
Cannot-link

ABSTRACT

During the last years, constrained clustering has emerged as an interesting direction in machine learning research. With constrained clustering, the quality of results can be improved by using constraints if a high-quality set of constraints is selected. Querying beneficial constraints is a challenging task because there is no metric for measuring the quality of constraints before clustering. A new method is proposed in this study that estimates density and impurity of data points on different adjacency distances and calculates centrality for each data point by applying a density tracking approach on the obtained densities. The obtained information is then used to select a set of high-quality constraints. Multi-resolution density analysis to more accurately estimate the point-point relationship of data, data density tracking in order to estimate the impurity and centrality of data, and selection of constraints from skeleton of clusters in order to discover the intrinsic structure of data can be mentioned as the most important contributions of this study. To verify the effectiveness of the proposed method, we conducted a series of experiments on real data sets. The obtained results show that the proposed algorithm can improve the clustering process compare with some recent reference algorithms.

1. Introduction

Clustering is an important task in the process of knowledge discovery in data mining. Clustering has wide applications in economic science, computer vision (Saha, Alok, & Ekbal, 2016), healthcare, information retrieval (Janani & Vijayarani, 2019) and the world wide web. In the past ten years, the problem of clustering with side information (known as constrained clustering) has become an active research direction to improve the quality of the results by integrating knowledge to the algorithms (Basu, Davidson, & Wagstaff, 2008; Basu, Banerjee, & Mooney, 2004; Yan, Zhang, Yang, & Hauptmann, 2006; Yin, Chen, Hu, & Zhang, 2010; Yeung & Chang, 2007; Grira, Crucianu, & Boujemaa, 2008; Ye, Liu, & Lou, 2015; Schwenker & Trentin, 2014; Faur & Schwenker, 2014; Junhua, Miao, Xingming, Wenxing, & Youjun, 2019; Śmieja, Struski, & Tabor, 2017). Constrained clustering uses a small set of side information to obtain the expected partitioning of data. Must-link (ML) and cannot-link (CL) constraints are the most widely used forms of side information in constrained clustering (Basu et al., 2008). A must-link constraint indicates that two points of the data set should be grouped in the same cluster while a cannot-link constraint imposes that the points should be grouped in different clusters (See Fig. 1).

Beside with the researching of the constrained clustering algorithm, the problem of choosing good constraints for constrained clustering is a crucial task. With a data set including n points, it can produce a lot of candidates for constraints (i.e. $\frac{n(n+1)}{2}$ pairs), even some poorly chosen constraints can lead to degrading the performance of clustering processes (Davidson, Wagstaff, & Basu, 2006). Moreover, it is very difficult to evaluate how the quality of a set of constraints are. In fact, with the same constrained clustering algorithm, the quality of the clustering process heavily depends on the quality of constraints that are given. Fig. 2 shows the relationship between constrained clustering and active learning problem in the context of expecting the best side information for constrained clustering. In recent years, the topic of active learning for constraints selection from users have been attracted a lot of attention. Many algorithms have been done for the problem of choosing the good constraints for constrained clustering, we can cite here the work of Grira et al. for fuzzy C-Means clustering in Grira et al. (2008), the work of Basu et al. for K-Means clustering in Basu et al. (2004), the work of Vu et al. for all kinds of constrained clustering in Vu, Labroche, and Bouchon-Meunier, 2012, the work of Abin et al. in (Abin & Beigy, 2014, 2015; Abin, 2019).

* Corresponding author.
E-mail addresses: a_abin@sbu.ac.ir (A.A. Abin), vuvietvu@vnu.edu.vn (V.-V. Vu).

https://doi.org/10.1016/j.eswa.2020.113690
0957-4174/

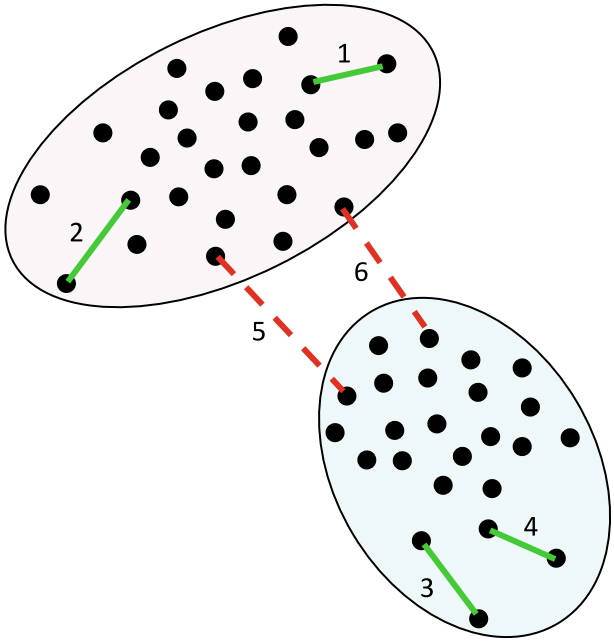


Fig. 1. An illustration of must-link (solid green lines 1 to 4) and cannot-link (dashed red lines 5 to 6) constraints in an example data.

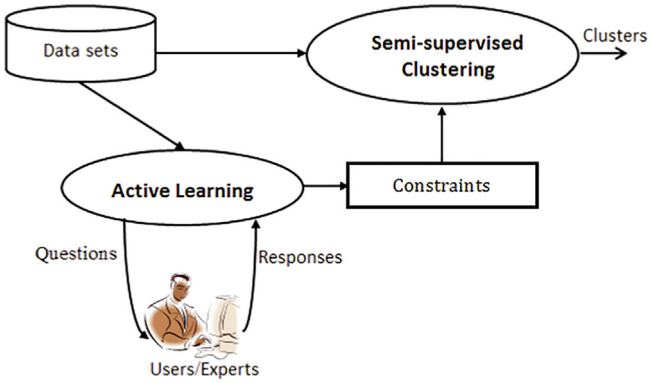


Fig. 2. Active learning and constrained clustering.

Most of the methods in the field of constraint selection have used the idea that constraints that resolve the ambiguity between clusters are useful, and thus have tried to select constraints from the boundaries of clusters. They have ignored the fact that the constraints that provide information about the skeleton of clusters can also be very useful. Constraints that provide useful information about the skeleton of clusters allow clustering algorithms to better discover complex-shaped clusters. In this work, we address the problem of beneficial constraints selection by considering three key assumptions on the usefulness of constraints based on the density relationship between data points. Based on these assumptions, we select constraints that provide helpful information about the boundary and skeleton of clusters. The proposed method estimates the density and impurity of data points on different adjacency distances in the first step and then applies a density tracking method on the obtained densities to calculate the centrality for each data point. Finally, the proposed method selects a set of high-quality constraints by using the information of density and impurity of data based on the above-mentioned assumptions.

1.1. Contributions of this work

Most of the existing studies in constraint selection are built upon assumptions on the usefulness of constraints without considering the density relationship between data points. In this study, we propose a new density-based approach for querying constraints. Like most existing methods, the proposed method tries to select constraints from the boundary of clusters. In addition, the proposed method uses the new idea of selecting constraints for the skeleton of clusters to provide helpful information about the skeleton of the clusters during the selection of constraints. Selecting constraints from the skeleton of clusters, along with constraints that provide information about cluster boundaries makes clustering algorithms accurate in determining the boundaries of clusters and discovering clusters with complex shapes.

Therefore, in selecting constraints, we have used three key assumptions about the usefulness of constraints. Based on these assumptions, a candidate constraint can be considered as a useful constraint if at least one of the following conditions is present: (1) It provides helpful information about the boundary points of clusters or, (2) it provides helpful information about the boundary between different clusters or, (3) It provides helpful information about the skeleton of clusters. Based on the first assumption, a selected constraint helps clustering algorithms to precisely determine the boundary for each cluster. On the other hand, constraints queried based on the second assumption dissolve the ambiguity of adjacent clusters and help clustering algorithms to precisely discover the boundary of clusters. Finally, constraints chosen by the third assumption give clustering algorithms useful information about the shape and distribution of clusters. Such constraints can be very useful for clustering algorithms that discover the shape of clusters or learn distance metrics by using constraints. Multi-resolution density analysis to more accurately estimate the point-point relationship of data, data density tracking in order to estimate the impurity and centrality of data, and selection of constraints from skeleton of clusters in order to discover the internal structure of data can be mentioned as the most important innovations of this study.

1.2. Organization of this paper

Density tracking as a new definition of the point-point relationship of data is explained in Section 3. The proposed method for querying constraints is given in Section 4. Section 6 analyses the proposed method from the idea behind and limitations. Experimental results are reported in Section 5. Finally, the conclusion and future directions are given in Section 7.

2. Related work

As mentioned above, the problem of choosing good constraints set for constrained clustering is a crucial task. In this section, we review the principal works that show the effectiveness of the constraints collection from users in literature.

The work of Klein et al. (Klein, Kamvar, & Manning, 2002) may probably be the first work to show the benefit when the constraints are carefully chosen for constrained clustering. In Basu et al. (2004), active learning for constraints collection based on the min–max method has been proposed; this algorithm is named as FFQS – Farthest First Query Selection. The FFQS algorithm includes two steps: *Explore* and *Consolidate*. The *Explore* steps identifies a set of CL constraints such that at the end of this step, at least one point has appeared in each cluster. To this aim, we need to know the number of clusters in prior and the set of collected cannot-link constraints forms a *skeleton* of the clusters. After that,

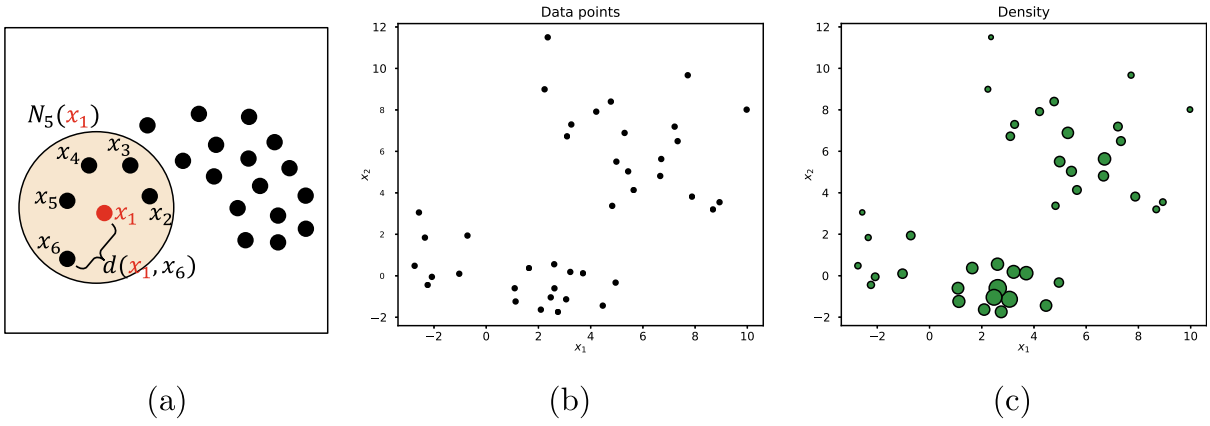


Fig. 3. (a) An illustration of density calculation on sample data point x_1, (b) A synthetic data set and, (c) Density of points in (b) depicted by solid green circles around. Lower density is depicted by a circle with a smaller radius.

at each iteration, the farthest data point from existing *skeleton* is chosen for queries from users. The second step *Consolidate* randomly picks a point, not in the *skeleton* and queries it against each point in the skeleton, until the user decides to stop labeling the constraints. An extension of the method is proposed in 2008 (Mallapragada, Jin, & Jain, 2008). The limitation of this method is that it can not collect constraints for the non-partition clustering such as constraints density-based clustering or constraint hierarchical clustering.

In Grira et al. (2008), Grira et al. introduced a method but for constrained *C*-Means clustering. The main idea of the proposed method is based on the measure of the border points in the clustering process. However, this method is only designed for the Fuzzy *C*-Means method. In Pei, Liu, and Fern (2015), Pei et al. proposed active query selection based on the Bayesian clustering method. With the method, an active learning framework that (1) iteratively selects the most informative pair of instances to query an oracle, and (2) updates the model parameters based on the obtained pairwise constraints. In Abin and Beigy (2015), the authors proposed a method based on multiple kernels learning. By that way, the limitation of improved possibilistic *C*-Means when dealing with spherical clusters is resolved because of the non-linear separable data is mapped into an appropriate feature space.

In Vu et al. (2012), Vu et al. proposed the ASC method. The key idea of the method is to select the points in the sparse regions to form the query candidates to get the label from users. One measure of the quality of constraints set has been proposed based on the *k*-nearest neighbor graph. The method can collect constraints for all kind of constrained clustering. However, some constraints may be not generated correctly to follow the propagation process of the method.

In Abin and Beigy (2014), the sequential selection of clustering constraints (SSCC) was proposed in which the queries process include three steps as follows: (1) identifying the boundary data based on a data description algorithm, (2) assigning each constraint a quality value based on two assumptions on the quality of constraints, and (3) sequentially querying the best quality constraint and updating the quality value of remaining constraints. The efficiency of SSCC highly depends on data description and is restricted to medium scale problems.

In Xu, desJardins, and Wagstaff (2005), Xu et al. proposed an active constraint selection for spectral clustering. The main idea of this work is based on the spectral theory to find data points locating on the boundaries of clusters. However, the work can only work with the two-clusters problem. Craenendonck et al. proposed a method for active query selection from users (van Craenendonck & Blockeel, 2017). In the first step, some clustering algorithms are used for clustering data. In the second step, the idea is that a pair is more uncertain if more clusterings disagree on whether it should be in the same cluster or not. A random walk approach for constraints selection (RWACS) was proposed in Abin (2017). RWACS takes a random walk on the proximity graph constructed on data points and recursively partitions the graph and looks for useful constraints between partitions. RWACS is somehow sensitive to the graph construction step.

A constraints selection method based on the embedding strategy was proposed by Abin in (Abin, 2019). The key idea of the proposed method is that the behavior of candidate constraints has tracked in various embedding spaces and assigned by a utility measure based on its resistance to destruction. The proposed method includes four steps: (1) a neighbors graph is used for expressing the data; (2) the distance between two data points is measured using the Euclidean commute time (ECT) and a random walk on the neighborhood graph; (3) the ECT distance is embedded into various spaces; and (4) the most useful constraints are chosen using utility value. The proposed method shows good effectiveness when compared with other methods. However, the complexity of the proposed method is $O(n^3)$.

3. Density tracking

In this section, we describe *density tracking* that considers a new definition of the point-point relationship. It is based on the assumption that a data point x_i should be related to its closest neighbor point x_j with a higher density. The relationship between x_i and x_j is called density relationship. This concept comes from the idea that it is enough to compare each data point with its neighbors (Sibson, 1973). There are generally three steps of the density tracking: density estimation, density following, and density group discovery. For each step, a detailed description is given in the following.

3.1. Step 1: Density estimation

In this step, the density of each data point x_i is estimated based on its neighborhood points. Let $\mathsf{N}_k(x_i)$ be set of k nearest neighbors to data point x_i. We use the following simple relation to estimate the density $\mathsf{Density}(x_i)$ for each data point x_i based on its k nearest neighbors.

$$\mathsf{Density}(x_i) = \frac{1}{\max\left(d(x_i, x_j)\right)}, \forall x_j \in \mathsf{N}_k(x_i) \tag{1}$$

where $d(x_i, x_j)$ stands for the Euclidean distance between x_j and x_j. Based on Eq. (1), the density of each data point x_i is inversely related to its distance to the farthest point in its neighborhood (see Fig. 3 (a)). So, it is very likely for points located in a sparse or boundary regions of data to be assigned low-density values (see Fig. 3(b) and (c)). There is no limitation to use other techniques such as counting the number of neighbors for data points or applying kernel functions on all data points to estimate the density of points. Algorithm 1 shows the heuristic used for density estimation.

Algorithm 1 Density estimation algorithm

procedure DENSITYESTIMATION(X, k)
  for each $x_i \in X$ **do** ▷$X = \{x_i\}_{i=1}^{n}$
    $x_{\text{distant}} \leftarrow$ The k^{th} nearest neighbor of x_i
    $\text{Density}(x_i) \leftarrow \frac{1}{d(x_i, x_{\text{distant}})}$ ▷Compute density of x_i
  end for
  return density of points Density
end procedure

3.2. Step 2: Density following

After density estimation, a density following procedure relates each data point x_i to its closest neighboring point with a higher density and extracts density chains based on the density relationships between data points. Starting from each data point x_i, we look for the closest data point $x_j \in \mathrm{N}_k(x_i)$ whose density is greater than that of x_i and record the relation between them as density chain $x_i \rightarrow x_j$. This procedure is continued for data point x_j and the closest data point $x_k \in \mathrm{N}_k(x_j)$ whose density is greater than that of x_j is found and added to the previous density chain as $x_i \rightarrow x_j \rightarrow x_k$. For each data point x_i, the chain grows until there exists no data point whose density is greater than that of the last data point of chain (See Fig. 4). During density following, we assign each data point x_i a *centrality index* $\text{Centrality}(x_i)$ that shows how many times a data points is presented on different density chains. So, for each data point x_i, its centrality index is set to zero in the beginning and is incremented by one for each presence of point x_i on density chains.

3.3. Step 3: Density group discovery

In this step, we extract the existing density groups based on the density relationship of data. To this end, all density chains with common end points are considered as connected density chains and points involved in them are considered to be in the same density group. In fact, each density group shows the tendency of points into a core density point. An illustration of density group discovery is given in Fig. 5. In this figure, different density groups are plotted in different colors.

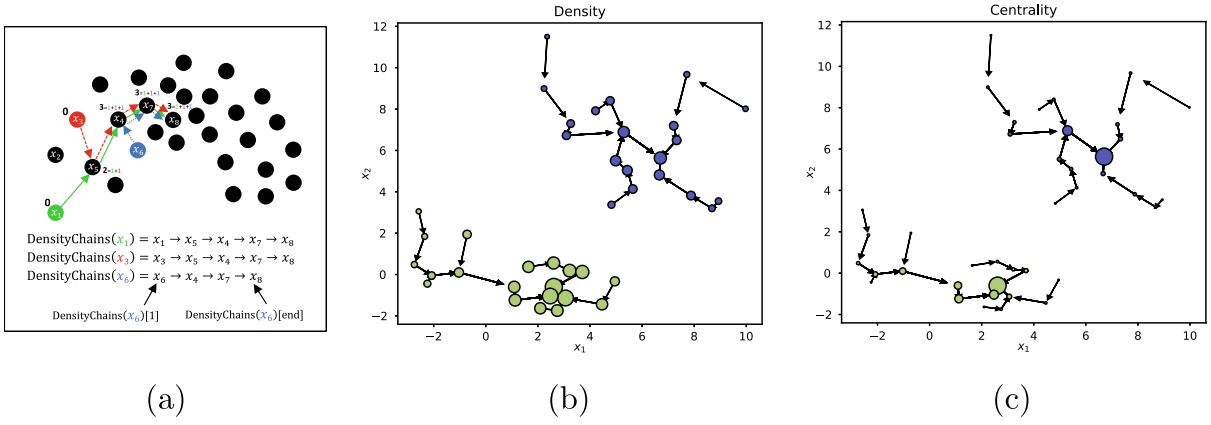


Fig. 4. (a) An illustration of density following on three sample data points x_1, x_3 and x_6. The printed number near each point shows the centrality index of the point, (b) Density following on synthetic data points plotted in Fig. 3(b) and, (c) Centrality of data points. For each data point, both density and centrality are depicted by solid circles. Lower values are depicted by smaller circles.

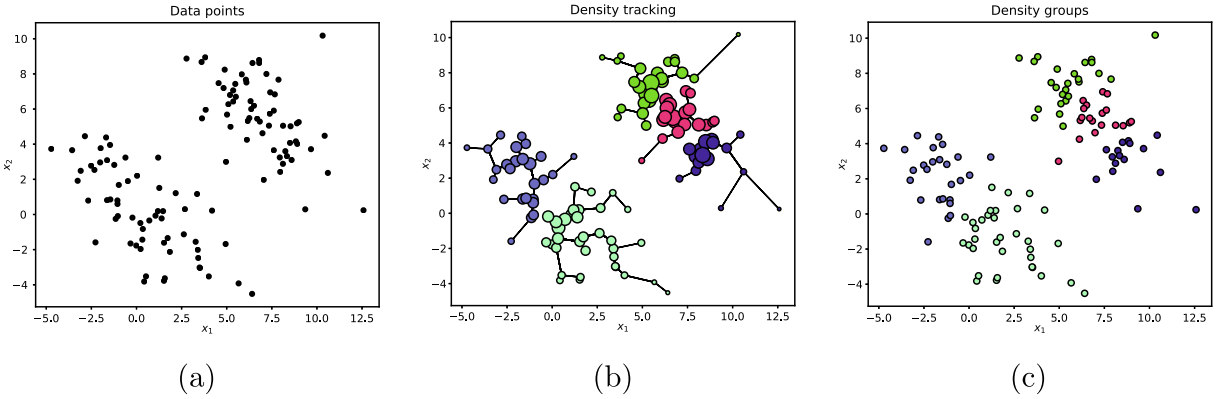


Fig. 5. (a) Synthetic data points, (b) density tracking on the points in (a) and, (c) the discovered density groups.

An illustration of density tracking is given in Algorithm 2. The result of the density track is comprised of the centrality of data points, a list of density chains and a set of density groups.

Algorithm 2 Density tracking algorithm

procedure DENSITYTRACKING (X, Density, k)
  for each $x_i \in X$ **do** ▷$X = \{x_i\}_{i=1}^n$
    DensityChains(x_i) ← Follow density for x_i to extract its density chain
    Centrality(x_i) ← Compute centrality of x_i
  end for
  DensityGroups ← Discover existing density groups
  return Centrality, DensityChains and, DensityGroups
end procedure

4. The proposed method

Let $X = \{x_i\}_{i=1}^n \in \mathbb{R}^d$ be set of data points. In a constraint selection problem, we want to choose λ constraints for clustering that contains as much information as possible. To this end, we use a density-based approach for querying constraints based on the following key assumptions on the quality of constraints. A candidate constraint is useful if at least one of the following conditions is met: (1) It provides helpful information about the boundary points of clusters or, (2) it provides helpful information about the boundary between different clusters or, (3) It provides helpful information about the skeleton of clusters. An illustration of the idea behind the proposed method was depicted in Fig. 6. In the first step, pre-processing is applied to data points to estimate their densities and impurities on different adjacency distances. In the second step, a density tracking is applied to the obtained densities to not only records centrality for each data point but also extract both density chain and group for each point. The obtained densities, impurities, and centralities along with the extracted density chains and density groups are then presented to a constraints selection method for querying beneficial constraints. The selected constraints are finally presented to a clustering algorithm for the next evaluation. In the following sections, each step is technically mentioned with more details.

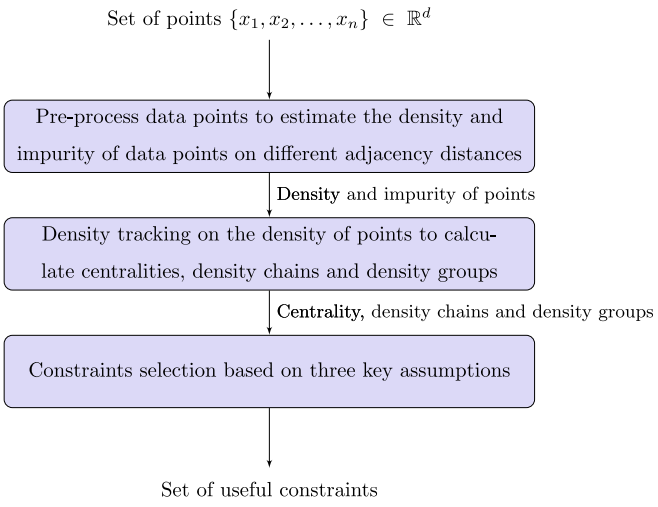


Fig. 6. The illustration of the proposed method.

4.1. Pre-processing of data points

For a better constraint selection, we pre-process the given dataset to estimate the density and impurity for each data point. Description of density estimation was given in Section 3 (see Algorithm 1). Impurity of each data point x_i is defined as the amount of impurity on cluster assignment for that point based on its adjacent points. To calculate the impurity of data points we apply density tracking on X by using Algorithm 2 and record how much a data point is impure by measuring: (1) the extent that the data point and its neighbors differ in density groups id and, (2) the extent that the density of the data point deviates from the density of the last data point of its chain. The first measure introduces a data point as impure if that point and its neighbors come from different density groups. To measure it we propose the following relation.

$$\text{Impurity}^{\{\mathbf{1}\}}(x_i) = \left(1 - \sum_{g=1}^{|\text{DensityGroups}|} Prob_g(\{x_i \cup \text{N}_k(x_i)\})^2\right) \tag{2}$$

where $Prob_g(S) = \frac{\sum_{x_j \in S} \mathbf{1}_{[\text{DensityGroups}(x_j)=g]}}{k+1}$ stands for the proportion of points in S that are from density groups g and $\mathbf{1}_{[\text{condition}]} = 1$ if condition is true.

On the other hand, the second measure records the deviation of density between a data point and the densest data point on its chain that is the last data point of the chain. This measure helps us to find data points whose densities are significantly different from the density at the end of the chain. We use the following relation to record this measure.

$$\text{Impurity}^{\{\mathbf{2}\}}(x_i) = 1 - \frac{\text{Density}(x_i)}{\text{Density}(\text{DensityChains}(x_i)[\text{end}])} \tag{3}$$

where $\text{DensityChains}(x_i) = x_i \to x_j \to \cdots \to x_{\text{end}}$ and $\text{DensityChains}(x_i)[\text{end}] = x_{\text{end}}$. Based on two impurity measures $\text{Impurity}^{\{\mathbf{1}\}}$ and $\text{Impurity}^{\{\mathbf{2}\}}$, we use the following equation to compute the impurity $\text{Impurity}(x_i)$ of data point x_i.

$$\text{Impurity}(x_i) = \text{Impurity}^{\{\mathbf{1}\}}(x_i) \times \text{Impurity}^{\{\mathbf{2}\}}(x_i) \tag{4}$$

Up to now, we described how to compute the impurity of each data points based on its k nearest neighbors. For a better estimation of impurity, we estimate the impurity of each data point x_i on different values of $k \in \{k_1, \ldots, k_m\}$ and returns the weighted sum of the estimated impurities as the total impurity of x_i. Such an idea is also used for the density of x_i in such a way that the weighted sum of the estimated densities is recorded as the density of x_i. But why don't we limit the computation of density and impurity to a fixed neighborhood radius? The reason for this is to examine the status of each data in different neighborhood radius in order to better estimate the density and impurity of the data. In some cases, the density and impurity of data may vary locally, but when viewed from a larger radius, we do not see significant changes. Therefore, in order to have a multiresolution analysis of the density and impurity of the data, we examined the status of each data in several different neighborhood radii. After calculating the density and impurity of each data in different neighborhood radius, their weighted sum is returned as the final result. In computing this weight, larger neighborhood radius contributes more weight to the sum of the data so that data density and impurity are not biased towards lower resolutions. If we give more weight to smaller neighborhood radius we may be biased towards the local state of the data and may not be able to accurately estimate the density and impurity of the data. The pre-processing step is fully described in Algorithm 3. In Fig. 7(b), the result of this step is shown on several synthetic datasets. The impurity of each data point is shown by solid circle in this figure, with the radius proportional to the degree of impurity.

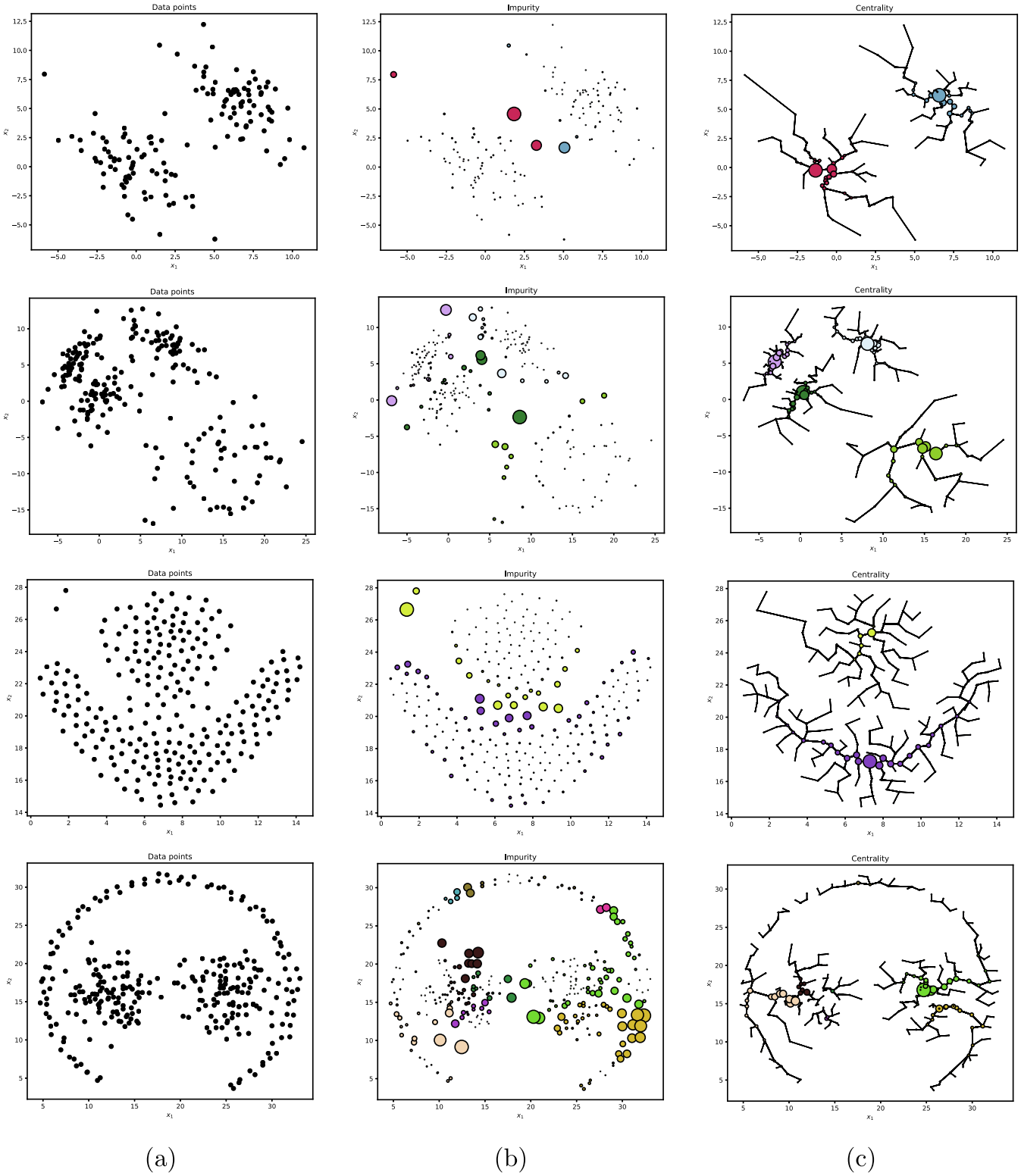
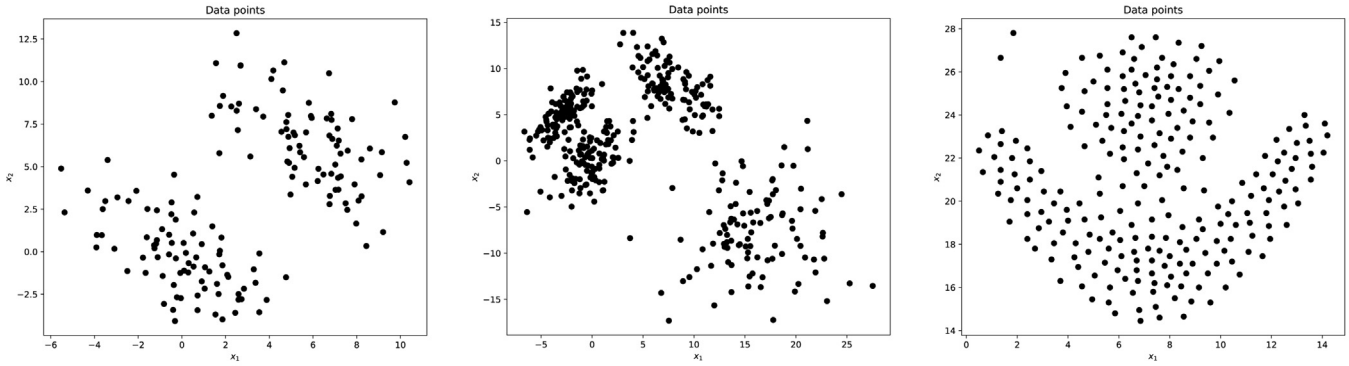
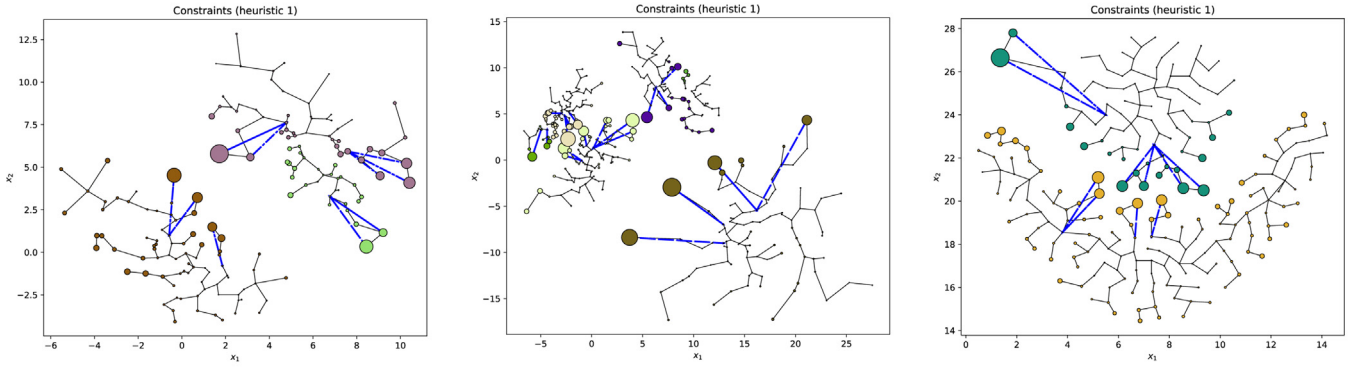


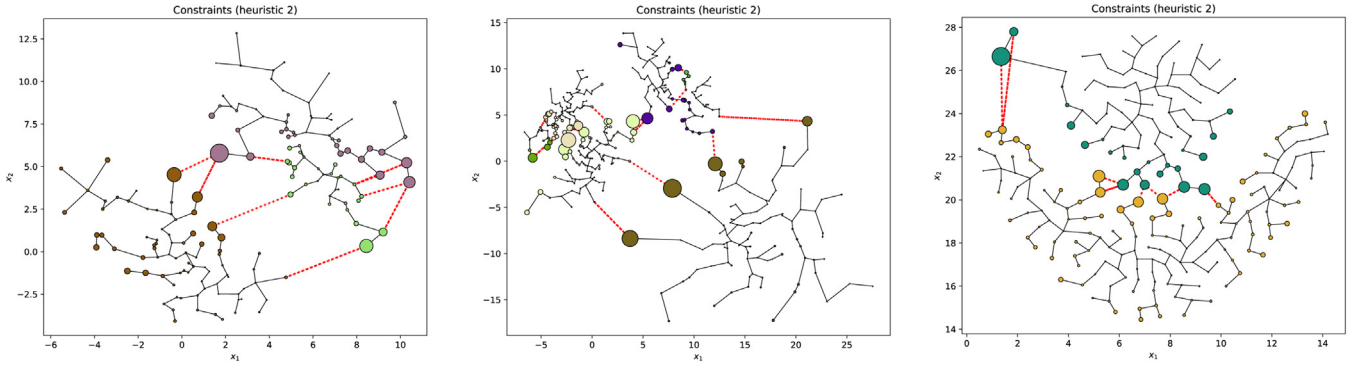
Fig. 7. (a) Synthetic data points, (b) the impurity of points depicted by solid circles around data points, and (c) the centrality of points along with the density chains and groups.



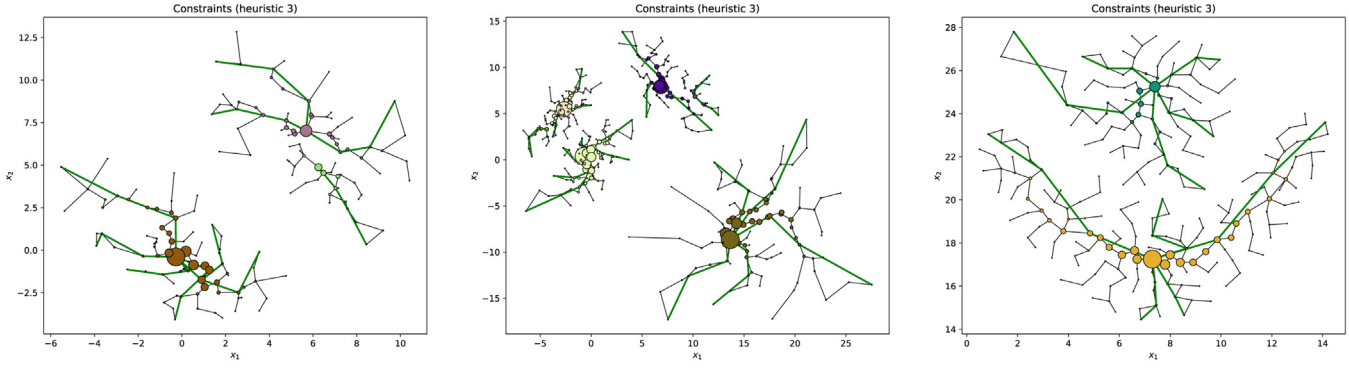
Synthetic data points.



Constraints selected based on assumption 1. The circles show the impurity of points.



Constraints selected based on assumption 2. The circles show the impurity of points.



Constraints selected based on assumption 3. The circles show the centrality of points.

Fig. 8. An illustration of constraints selected by three basic assumptions.

Algorithm 3 Pre-processing algorithm

procedure PRE-PROCESS X, K
  $t = 1$
  γ=0.95 ▷Discount factor
  for each $x_i \in X$ **do** ▷Initialization
    GlobalDensity$(x_i) = 0$
    GlobalImpurity$(x_i) = 0$
  end for
  for each $k \in K$ **do** ▷$K=\{k_1,k_2,...,k_m\}, k_1<k_2<...<k_m$
    Density $\leftarrow$ DENSITYESTIMATION(X, k)
    Centrality, DensityChains, DensityGroups $\leftarrow$ DENSITYTRACKING $(X,$ Density$, k)$
    for each $x_i \in X$ **do**
      Impurity$(x_i) \leftarrow$ Compute impurity of x_i by using (4)
      GlobalImpurity(x_i)=GlobalImpurity$(x_i)+\gamma^{m-t}\times$Impurity(x_i)
      GlobalDensity(x_i)=GlobalDensity$(x_i)+\gamma^{m-t}\times$Density(x_i)
    end for
    $t = t + 1$
  end for
  return GlobalDensity and GlobalImpurity
end procedure

Table 1
Test data from UCI machine learning repository.

Dataset	n	d	C
Iris	150	4	3
Glass Identification	214	9	6
Soybean	47	34	4
Ionosphere	354	31	2
Wine	178	13	3
Sonar	208	60	2
Heart	270	13	2
Balance Scale	625	4	3

4.2. Density tracking

After computing density and impurity for each data point in different neighborhoods, in this step, once again, we perform density tracking on the data with the obtained densities, and for each data point, we compute its centrality and extract its density chain and density group. The reason for using density values obtained from the preprocessing step is to have a more accurate estimation of the densities. So, we did not limit ourselves to a fixed neighboring radius and looked at the relationships in different neighborhoods. Because in some cases the relationships of data points may vary locally, but not globally. In Fig. 7(c), the result of this step is shown on several synthetic datasets. The centrality of each data point in this figure is shown by solid circles, with their radius proportional to the degree of centrality.

4.3. Constraints selection

The last step in the proposed method is to select beneficial constraints based on the three mentioned assumptions. In the following, we will describe how to choose constraints based on each of the assumptions.

4.3.1. Selecting constraints based on assumption 1

According to the first assumption, it is useful to provide information about the boundary points of clusters, or, more precisely, the boundary points of density groups for the clustering algorithm. To do this, we use the impurity of data points along with the information of density chains to select such constraints. How to select a constraint according to this assumption is that we first sort the data points according to their impurities in descending order. Then we start to select data points from the beginning of the ordered list (the most impure point on top) and for each impure point selected, its relation is queried with the first point on its density chain, whose density is greater than the *density_drop_rate* percentage of the density of chain last point (see the second row in Fig. 8).

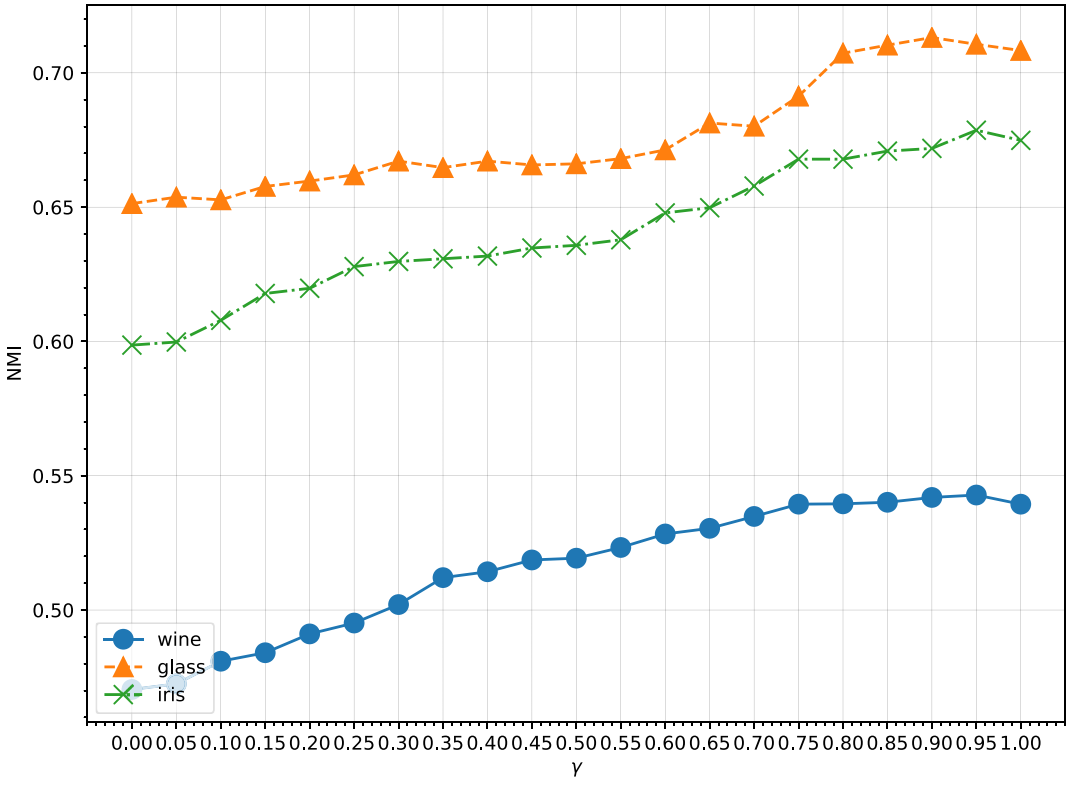


Fig. 9. Grid search in parameter space for γ parameter on three Iris, Wine, and Glass datasets.

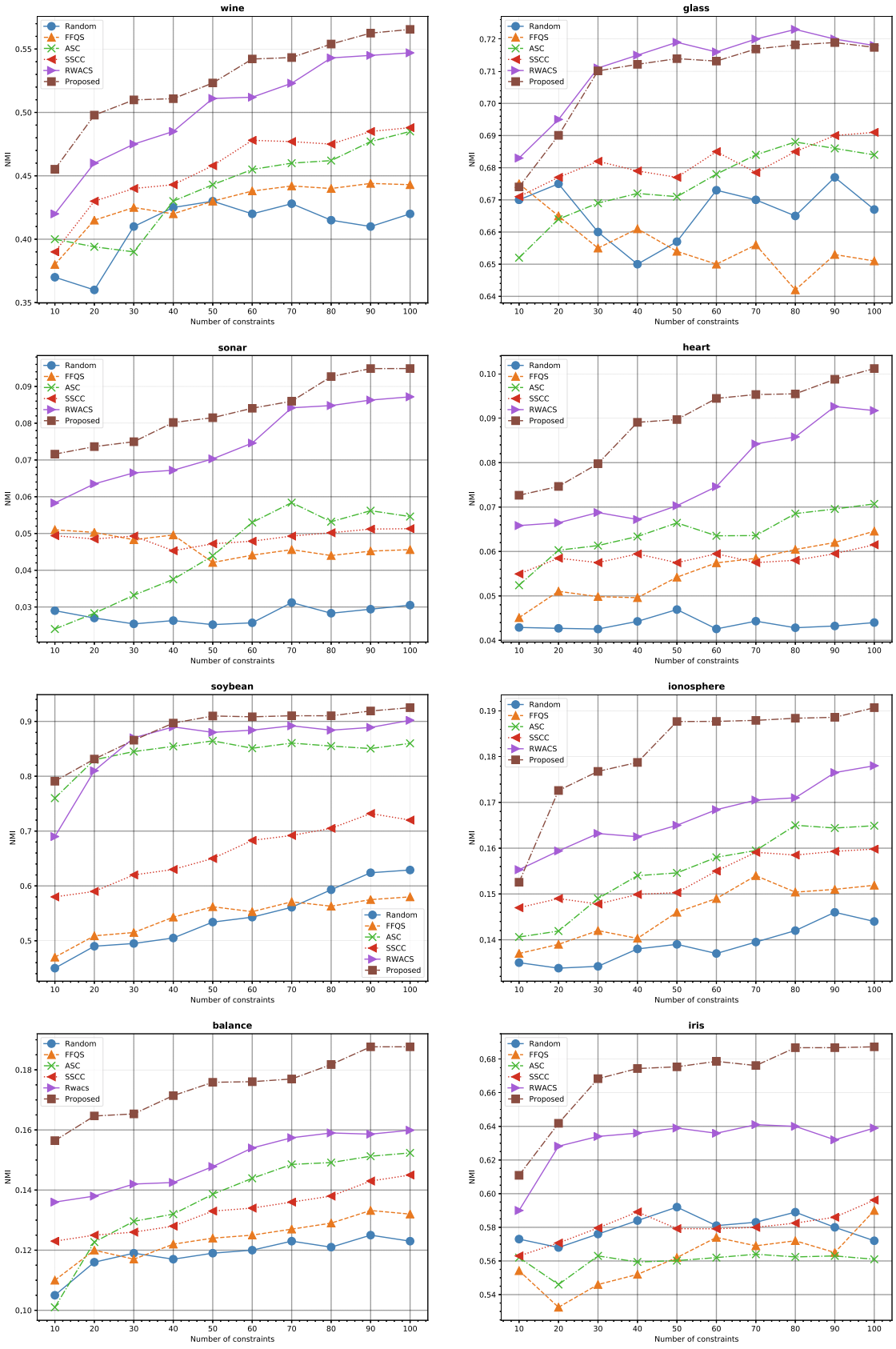


Fig. 10. Evaluation of the quality of constraints by MPC-KMeans (Bilenko et al., 2004).

4.3.2. Selecting constraints based on assumption 2

According to the second assumption, constraints that provide useful information about the boundaries between clusters or density groups are known as useful constraints. More precisely, any constraint that eliminates the ambiguity of clusters boundaries is more beneficial. The proposed method for choosing these types of constraints is to first sort constraints based on their impurity in descending order. Then, from the beginning of the ordered list, we select the most impure point and ask its relation to the nearest point from other density groups (see the third row of Fig. 8).

4.3.3. Selecting constraints based on assumption 3

According to the third assumption, we select constraints that provide useful information on the shape and distribution of data. To this end, we use density chains along with the centrality of data, and try to select useful constraints from the skeleton of clusters. We expect these constraints to be very useful in discovering cluster shapes and learning distance metric. The method of choosing such constraints is that for each density chain, the sum of the centrality of points located on the chain is calculated. We then sort the density chains based on their total centrality values. Then the chain with the highest amount is selected as a candidate for the cluster skeleton. Selecting constraints from the candidate chain is that we start from the beginning of the chain and select points along the chain with an interval, and ask the relation of both consecutive points as a constraint. After choosing constraints from the first candidate chain, we are going to choose the second candidate chain. But before that, the centrality of points appeared on the first candidate chain is set to zero. This avoids the selection of chains that are common in many points and discovers clusters skeleton from different directions. The last row in Fig. 8 shows the constraints selected based on this assumption for several sample data, along with information on the centrality of points and density groups. Algorithm 4 shows how to select constraints based these assumptions.

Algorithm 4 Constraints selection algorithm

procedure SELECTCONSTRAINTS X, Density, Impurity, DensityChains, DensityGroups, λ, $sampling\_rate = 3$, $density\_drop\_rate = 0.8$
  $list = \{\}$ ▷Selecting constraints based on assumptions 1 & 2
  X_I = Sort X based on impurity of points in descending order
  $limit = \frac{\lambda}{3}$
  for $c = 1, \ldots, limit$ **do** ▷Loop to select constraints
    $x_{\text{from}} \leftarrow$ Pop the most impure point form X_I
    $x_{\text{end}} = \text{DensityChains}(x_{\text{from}})[\text{end}]$ ▷1
    for $i = 2, \ldots, |\text{DensityChains}(x_{\text{from}})|$ **do**
      $x_{\text{to}} = \text{DensityChains}(x_{\text{from}})[i]$
      if $\text{Density}(x_{\text{to}}) \geqslant density\_drop\_rate \times \text{Density}(x_{\text{end}})$ **then**
        Break
      end if
    end for
    $list = list \cup \{(x_{\text{from}}, x_{\text{to}})\}$
    $x^*_{\text{to}} \leftarrow \underset{x_{\text{to}} \in \text{DensityGroups} \setminus \text{DensityGroups}(x_{\text{from}})}{\arg\min} \|x_{\text{from}} - x_{\text{to}}\|_2$ ▷2
    $list = list \cup \{(x_{\text{from}}, x^*_{\text{to}})\}$
  end for
  $limit = \frac{\lambda}{3 \times sampling\_rate}$ ▷Selecting constraints based on assumption 3
  for $c = 1, \ldots, limit$ **do**
    $chain^* \leftarrow \text{argmax}_{chain \in \text{DensityChains}} \left(\sum_{x \in chain} \text{Centrality}(x)\right)$
    $step = \lfloor \frac{|chain^*|}{sampling\_rate} \rfloor$
    for $i = 0, \ldots, sampling\_rate - 1$ **do**
      $x_{\text{from}} = chain^*[i \times step]$
      $x_{\text{to}} = chain^*[(i+1) \times step]$
      $list = list \cup \{(x_{\text{from}}, x_{\text{to}})\}$
    end for
    for each $x \in chain^*$ **do**
      $\text{Centrality}(x) = 0$
    end for
  end for
  return *list*
end procedure

Algorithm 5 shows three stages of pre-processing, density tracking, and constraints selection from the proposed method altogether.

Algorithm 5 The proposed method for querying high-quality constraints

procedure PROPOSEDMETHOD $X, K = \{k_1, k_2, \ldots, k_m\}, \lambda$
  Density, Impurity $\leftarrow$ PRE-PROCESS(X, K)
  $k = k_{\lfloor \frac{m}{2} \rfloor}$
  Centrality, DensityChains, DensityGroups $\leftarrow$ DENSITYTRACKING $(X, \text{Density}, k)$
  $list \leftarrow$ SELECTCONSTRAINTS $(X, \text{Density}, \text{Impurity}, \text{DensityChains}, \text{DensityGroups}, \lambda, 3, 0.8)$
  return set of constraints *list*
end procedure

5. Experiments

To evaluate the proposed method, we have experimented on several datasets and compared the quality of the selected constraints with other methods. The default values for $K = \{k_1, k_2, \ldots, k_m\}$, *density_drop_rate* and *sampling_rate* are set to $\{5, 7, 9, 11, 13, 15\}$, 0.8 and 3, respectively. We perform a fixed-step grid search in parameter space to determine the best-fit values of γ in Algorithm 3. The step size of the search is chosen to be 0.05. Descriptions on the compared methods, clustering techniques, test datasets and evaluation metric are given in the following.

5.0.4. Compared methods

The quality of constraints is compared against Random, FFQS (Basu et al., 2004), ASC (Vu et al., 2012), SSCC (Abin & Beigy, 2014) and, RWACS (Abin, 2017) methods. ASC is initialized by $k = 6$ and $\theta = 4$ as suggested by the author. We configure SSCC and RWACS by their suggested default values. The quality of constraints for Random and FFQS methods are averaged over 75 runs.

5.0.5. Clustering methods

The quality of constraints in all experiments are evaluated by two well-known methods in constrained clustering MPC-KMeans (Bilenko, Basu, & Mooney, 2004) and RCA (Bar-Hillel, Hertz, Shental, & Weinshall, 2005). MPCK-Means is built upon metric learning and learns a distance metric for each cluster by using the input constraints. Relevant component analysis (RCA) is another leading method for constrained clustering proposed by Hillel et al. (Bar-Hillel et al., 2005). RCA learns a global distance measure based on the information of constraints and tries to identify and down-scale global unwanted variability within the data by using side information.

5.0.6. Datasets

We evaluate the quality of results on some datasets from UCI machine learning repository[1] (see Table 1). In this table, n, d and, C stand for the Number of objects, number of attributes and number of clusters, respectively.

5.0.7. Evaluation metric

Normalized mutual information (NMI) as a well-known metric in clustering research area is used to evaluate the quality of clustering in all experiments. Let A be the result of clustering and B

[1] http://archive.ics.uci.edu/ml/

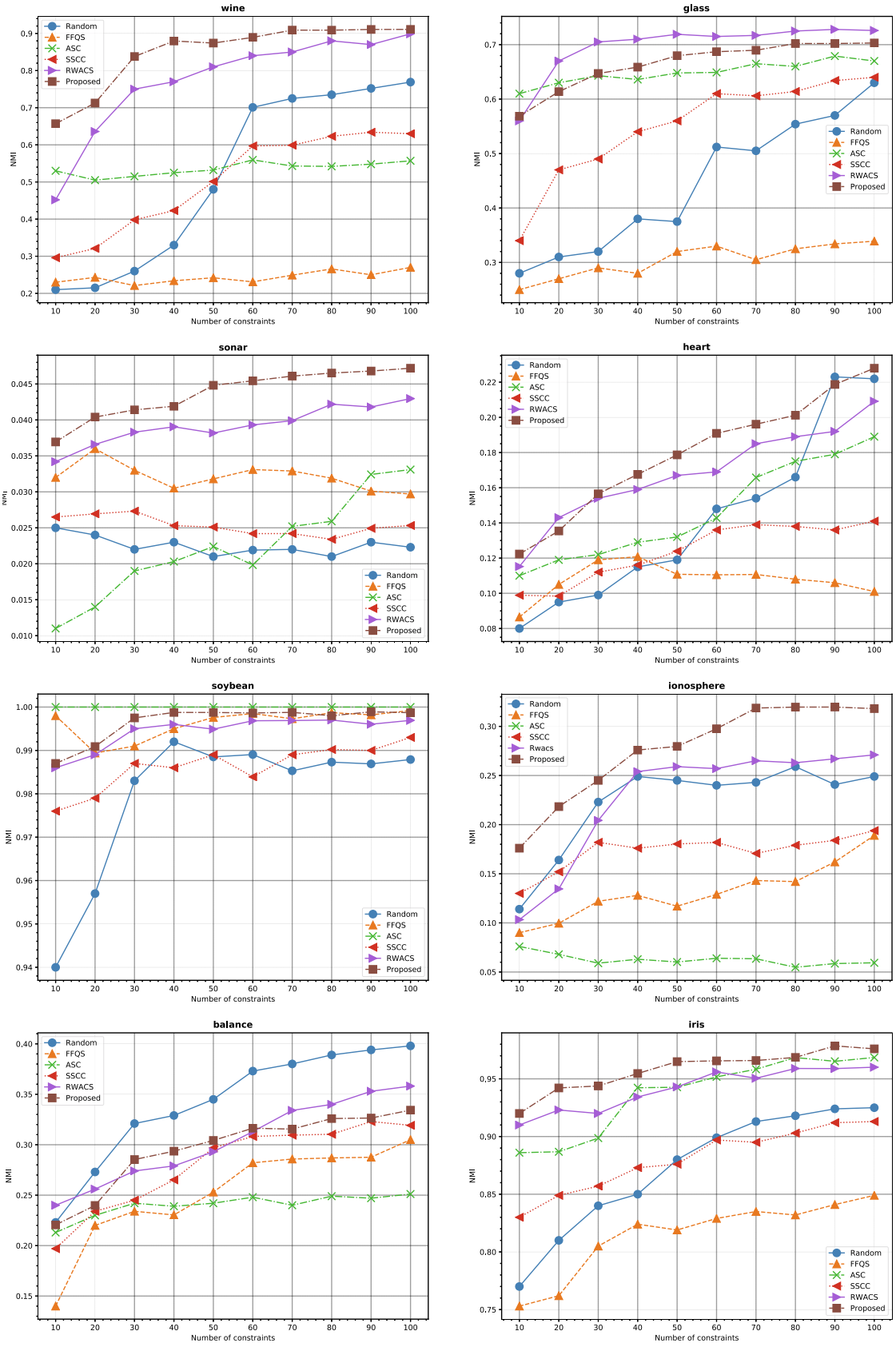


Fig. 11. Evaluation of the quality of constraints by RCA (Bar-Hillel et al., 2005).

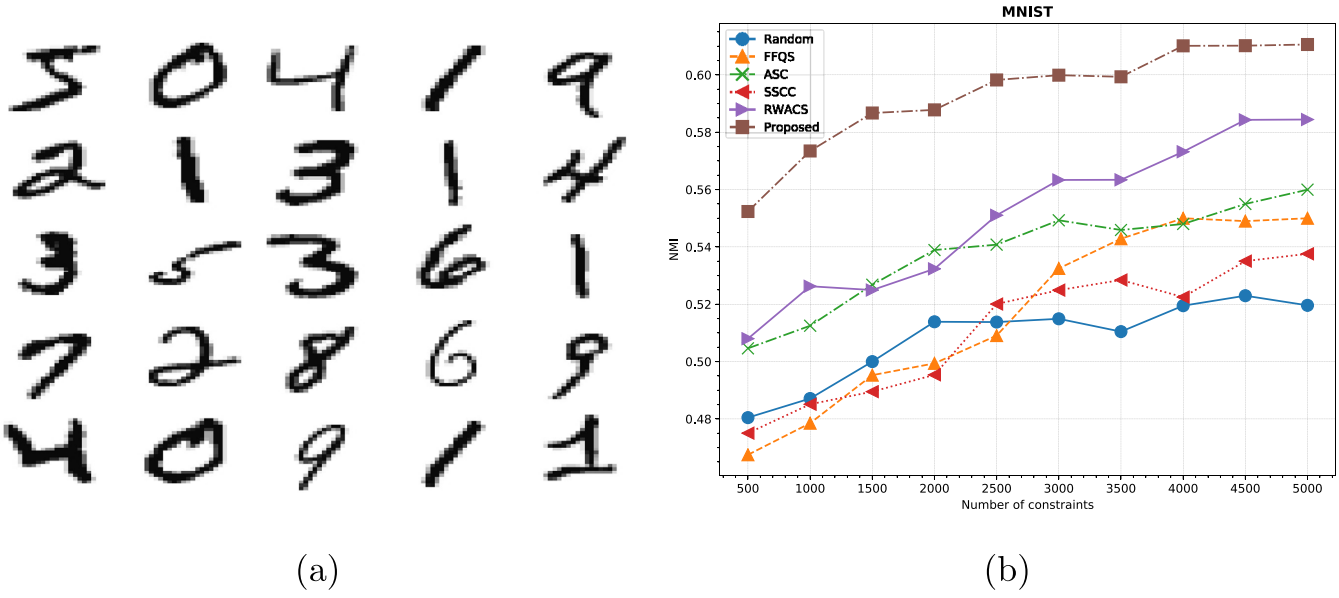


Fig. 12. (a) An illustration of some images from MNIST dataset, and (b) Evaluation of the quality of constraints by MPC-KMeans.

be the true labels. $NMI(A,B)$ is defined as $\frac{I(A,B)}{\sqrt{H(A)H(B)}}$, where $I(A,B)$ is the mutual information between A and B, and $H(A)$ and $H(B)$ are the entropies of A and B, respectively (Strehl & Ghosh, 2003).

To determine the best-fit value of γ in Algorithm 3, we performed a fixed-step grid search in parameter space with step size 0.05 on all datasets given in Table 1 and found that choosing γ in the range of $[0.80, 0.95]$ gives good result. So, we set the value of the γ parameter to 0.95. Fig. 9 shows the grid search results for three Iris, Wine, and Glass datasets when 60constraints are selected by the proposed method and evaluated by MPCK-Means algorithm. As we can see in this figure, the proposed method has given better results by selecting gamma around 0.95.

A comparison of the quality of constraints is plotted in Figs. 10 and 11 by reporting the name, the number of constraints and, the mean NMI. As these figures show, the proposed method outperforms the compared methods generally. It implies that density tracking may be the right direction to select beneficial constraints. An important issue in constraints selection is that a given set of constraints may be useful or not for clustering algorithms (See the quality of randomly selected constraints in Figs. 10 and 11). As the results show, this phenomenon has not occurred in the proposed method, which suggests that assumptions for choosing constraints can be valid assumptions.

If we want to have an analysis of the results, it should be noted that the method based on random behavior cannot select useful constraints for clustering. The reason is that such methods often choose constraints without any assumption on the quality of constraints. This can be seen in the case of two Random and FFQS algorithms (See the results for Glass and Iris datasets in Fig. 10).

Another important issue in the selection of constraints is that in some cases, by increasing the number of constraints, the quality of clustering is reduced. For example, this case is evident from Fig. 10 for Glass data. In this figure, the quality of clustering is reduced by increasing the number of constraints chosen by FFQS method. This is not the case with the proposed method. This indicates that the more constraints we choose, the more ambiguities of clustering will be eliminated.

The proposed method was also tested on MNIST dataset[2]. MNIST dataset contains 60000handwritten digits that is flattened from 28×28 images into $\mathbb{R}^{784}$ vectors. The result is given in Fig. 12.

[2] http://yann.lecun.com/exdb/mnist/

6. Analysis of the proposed method

In this section, we investigate the proposed method from the idea behind, limitations and time complexity. As mentioned in Section 4, the proposed method uses the idea of density tracking to query beneficial constraints based on three key assumptions. Based on these assumptions, the proposed method tries to select constraints that provide helpful information about the boundary and skeleton of clusters. Fig. 13 shows the output of the proposed method on several synthetic data. In this figure, must-link and cannot-link constraints are drawn by solid green and dashed red line colors, respectively. As this figure shows, the constraints are either selected from the boundary or the skeleton of clusters.

A noteworthy point in Fig. 13 is the constraints selected from the skeleton of clusters that provide useful information about the shape of clusters. Another point is the constraints selected from the boundary of clusters that are selected from the right areas and significantly eliminate clustering ambiguities. Using the idea of density tracking in constraints selection gives the proposed method the ability to examine the quality of constraints in various densities based on the impurities of data points. This can be useful when no prior knowledge about the shape and distribution of data exists.

An important point about the proposed method is that the quality of results depends on three key assumptions on constraint selection. We performed some experiments to analyse the clustering performances if the proposed method only use a single assumption for selecting the constrains. As mentioned in Section 4.3, the proposed method selects constrains based on three following assumptions if the constraints provide helpful information about: (1) the boundary points of each cluster or, (2) the boundary between different clusters or, (3) the skeleton of clusters. Fig. 14 shows the quality of clustering when the proposed method uses different assumptions for selecting the constrains. As this figure shows, in most datasets, the third assumption i.e. selecting the constrains from the skeleton of clusters plays a greater role in the quality of clustering.

One of the main limitation of the proposed method is that the density-based analysis is not very reliable for a small number of high-dimensional data. This is true not only for the proposed method, but also for most existing methods that use density-based analysis. This issue can affect the accuracy of density

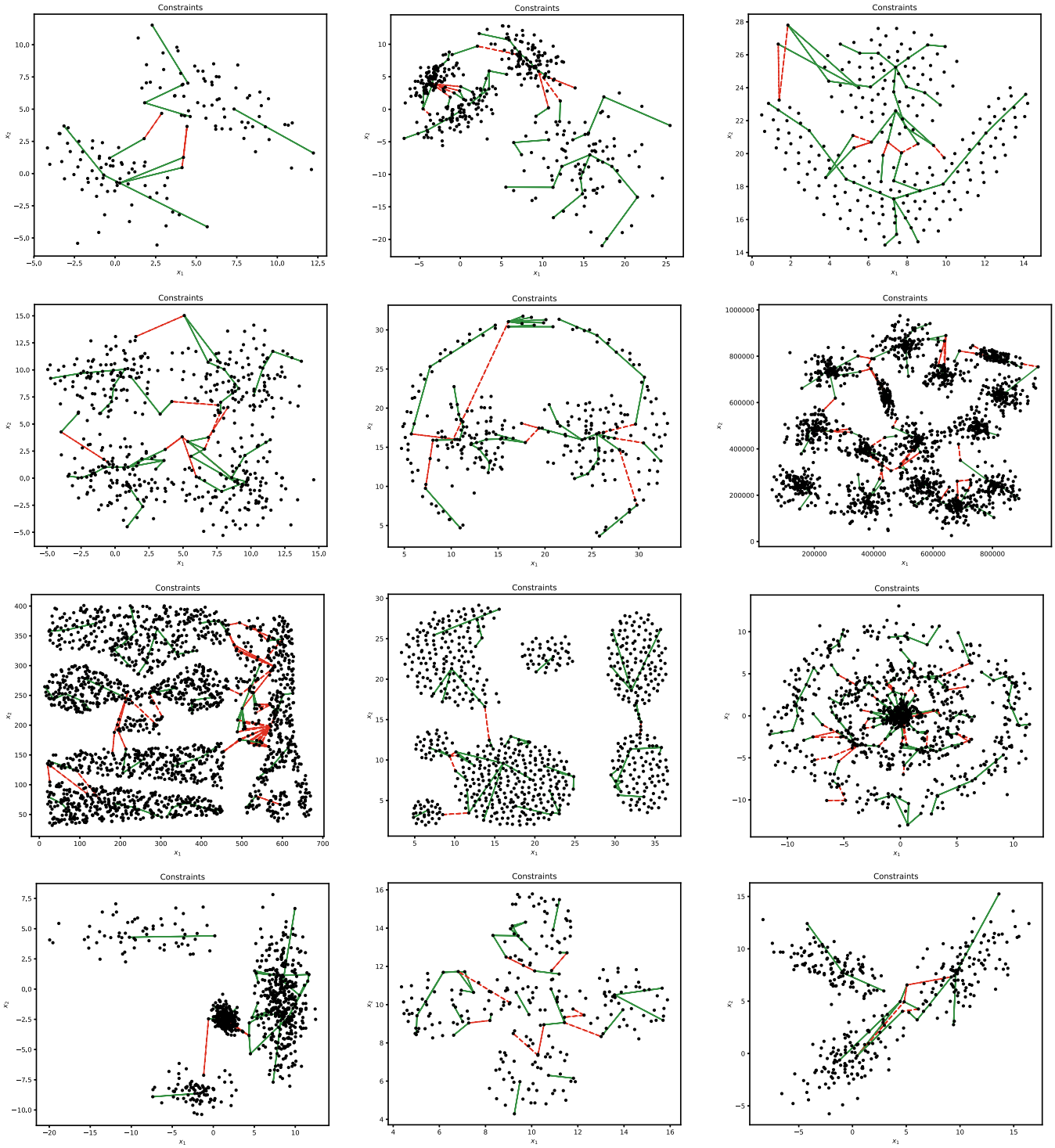


Fig. 13. An illustration of constraints selected by the proposed method on several synthetic datasets.

estimation and consequently the quality of the proposed method. The proposed solution to this problem is to increase the number of data, which in many cases is not possible. Also, in some cases where we have enough data, we may still not have enough samples from some areas of the data space for reasons such as improper sampling and make an incorrect estimate of the local or global density of data. This may affect the quality of the extracted skeletons and reduce the quality of constraints. Although in this article we have tried to cover this issue to some extent by multi-resolution analysis of densities, but still there may be a situation where due to the selection of inappropriate parameters we can not detect the continuity of density and consequently can not choose high quality constraints.

A suggestion for further research in the field of active selection of clustering constraints is to select the constraint that provides new information in discovering clusters that is not present in the previously set of selected constraints. In fact, when selecting constraint at each step, avoid redundant information as much as

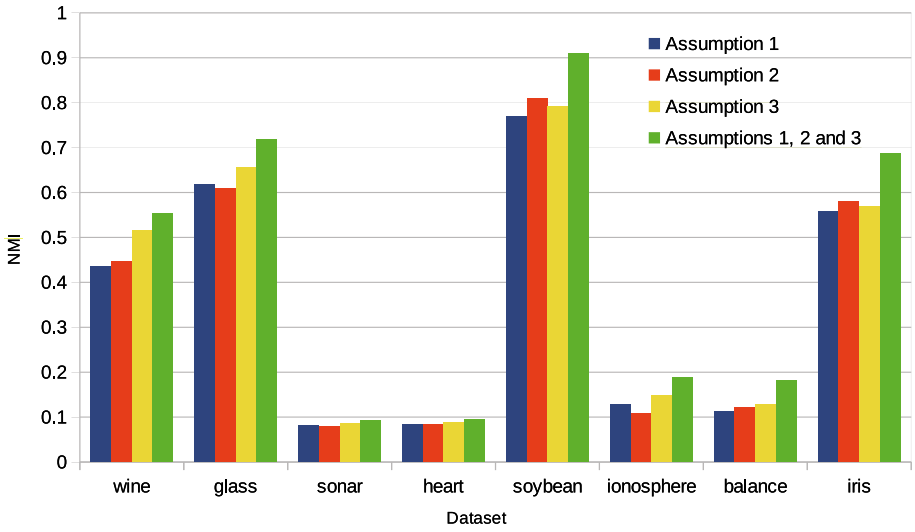


Fig. 14. Quality of clustering when the proposed method uses different assumptions for selecting the constrains.

possible. In this way we can improve the current state of knowledge in this field.

6.1. Complexity analysis of the proposed method

Many operations in the proposed method relied on the computation of k nearest neighbors. The complexity of finding k nearest neighbors of data points $X \in \mathbb{R}^d$ is $O(kdn^2)$. Having k nearest neighbors, density estimation on a given neighborhood k is $O(dn)$ if we use Euclidean distance measure. The complexity of density tracking is $O(kln)$ where l is the maximum length of chains. So, the complexity of preprocessing step in the proposed method on nearest neighbors set $K = \{k_1, k_2, \ldots, k_m\}$ is $O(k_m dn^2 + m(dn) + m(k_m ln))$ for the computation of k_m nearest neighbors, density estimation and density tracking, respectively. Density tracking after preprocessing has the cost of $O(k_m ln)$ at the worst case. In constraints selection based on assumptions 1 and 2, sorting impurities takes $O(n \log n)$ and constraints selection based on assumptions 1 takes $O(\lambda l)$ and constraints selection based on assumptions 2 takes $O(\lambda nd)$. The complexity of constraints selection based on assumptions 3 is $O(\lambda(nl + sampling\_rate + l)$. Because *sampling_rate* and m can be considered as constants and the maximum length of chains l is n, The complexity of the proposed method is $O\left(max\left(\lambda n^2, k_m dn^2\right)\right)$.

7. Conclusions

In this work, the idea of density tracking is used for beneficial constraints selection before clustering. We considered the density relationship between data points and proposed a method for constraints selection that considers three key assumptions on the quality of constraint. Constraints that provide helpful information about the boundary and skeleton of clusters were considered as high-quality constraints in these assumptions. Based on these assumptions, a three steps method was proposed that computes density and impurity of data points in the first step and then applies a density track on the obtained densities to calculate centrality for each data point. Finally, the obtained information is used to select a set of useful constraints. A promising future direction is to extend the proposed method to avoid redundant constraints during sequential selection of clustering constraints. Extending the proposed method to handle a relatively small number of high-dimensional data can also be considered as another direction to the future work.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: This research is funded by Vietnam National University, Hanoi (VNU) under project number QG.18.40.

Acknowledgment

This research is funded by Vietnam National University, Hanoi (VNU) under project number QG.18.40.

References

Abin, A. A. (2017). *A random walk approach to query informative constraints for clustering* (pp. 1–12). PP: IEEE Transactions on Cybernetics.
Abin, A. A. (2019). Querying informative constraints for data clustering: An embedding approach. *Applied Soft Computing, 80*, 31–41.
Abin, A., & Beigy, H. (2014). Active selection of clustering constraints: a sequential approach. *Pattern Recognition, 47*(3), 1443–1458.
Abin, A. A., & Beigy, H. (2014). Active selection of clustering constraints: a sequential approach. *Pattern Recognition, 47*, 1443–1458.
Abin, A., & Beigy, H. (2015). Active constrained fuzzy clustering: A multiple kernels learning approach. *Pattern Recognition, 48*(3), 953–967.
Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. (2005). Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research, 6*, 937–965.
Basu, S., Banerjee, A., & Mooney, R. J. (2004). Active semi-supervision for pairwise constrained clustering. In *Proceedings of the SIAM international conference on data mining, SDM-2004* (pp. 333–344).
Basu, S., Davidson, I., & Wagstaff, K. L. (2008). *Constrained clustering: advances in algorithms, theory, and applications*. Chapman and Hall/CRC Data Mining and Knowledge Discovery Series.
Bilenko, M., Basu, S., & Mooney, R. J. (2004). Integrating constraints and metric learning in semi-supervised clustering. In *Machine learning proceedings of the twenty-first international conference (ICML 2004), Banff, Alberta, Canada, July 4–8, 2004.* . 2004 (pp. 11–18)..
Davidson, I., Wagstaff, K., & Basu, S. (2006). Measuring constraint-set utility for partitional clustering algorithms. In *PKDD* (pp. 115–126).
Faur, S., & Schwenker, F. (2014). Semi-supervised clustering of large data sets with kernel methods. *Pattern Recognition Letters, 37*, 78–84.
Grira, N., Crucianu, M., & Boujemaa, N. (2008). Active semi-supervised fuzzy clustering. *Pattern Recognition, 41*(5), 1834–1844.
Janani, R., & Vijayarani, S. (2019). Text document clustering using spectral clustering algorithm with particle swarm optimization. *Expert Systems with Applications, 134*, 192–200.
Junhua, C., Miao, T., Xingming, Q., Wenxing, W., & Youjun, L. (2019). A solution to reconstruct cross-cut shredded text documents based on constrained seed

k-means algorithm and ant colony algorithm. *Expert Systems with Applications, 127*, 35–46.
Klein, D., Kamvar, S., & Manning, C. (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *Proceedings of the nineteenth international conference* (pp. 307–314).
Mallapragada, P. K., Jin, R., & Jain, A. K. (2008). Active semi-supervision for pairwise constrained clustering. In *Proceedings of the SIAM international conference on data mining, SDM-2004* (pp. 333–344).
Pei, Y., Liu, L., & Fern, X. (2015). Bayesian active clustering with pairwise constraints. *Machine learning and knowledge discovery in databases – european conference*, 235–250.
Saha, S., Alok, A. K., & Ekbal, A. (2016). Brain image segmentation using semi-supervised clustering. *Expert Systems with Applications, 52*, 50–63.
Schwenker, F., & Trentin, E. (2014). Pattern classification and clustering: A review of partially supervised learning approaches. *Pattern Recognition Letters, 37*, 4–14.
Sibson, R. (1973). Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal, 16*, 30–34.
Śmieja, M., Struski, U., & Tabor, J. (2017). Semi-supervised model-based clustering with controlled clusters leakage. *Expert Systems with Applications, 85*, 146–157.
Strehl, A., & Ghosh, J. (2003). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research, 3*, 583–617.
van Craenendonck, T., & Blockeel, H. (2017). Constraint-based clustering selection. *Machine Learning, 106*(9–10), 1497–1521.
Vu, V.-V., Labroche, N., & Bouchon-Meunier, B. (2012). Improving constrained clustering with active query selection. *Pattern Recognition, 45*, 1749–1758.
Xu, Q., desJardins, M., & Wagstaff, K. L. (2005). Active constrained clustering by examining spectral eigenvectors. In *Proceedings of discovery science conference, DS-2005* (pp. 294–307).
Yan, R., Zhang, J., Yang, J., & Hauptmann, A. (2006). A discriminative learning framework with pairwise constraints for video object classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(4), 578–593.
Ye, Y., Liu, R., & Lou, Z. (2015). Incorporating side information into multivariate information bottleneck for generating alternative clusterings. *Pattern Recognition Letters, 51*, 70–78.
Yeung, D.-Y., & Chang, H. (2007). A kernel approach for semisupervised metric learning. *IEEE Transactions on Neural Networks, 18*(1), 141–149.
Yin, X., Chen, S., Hu, E., & Zhang, D. (2010). Semi-supervised clustering with metric learning: An adaptive kernel method. *Pattern Recognition, 43*(4), 1320–1333.