Journal of Classification (2020) 37:789–809
https://doi.org/10.1007/s00357-019-09349-x

A Note on the Formal Implementation of the *K*-means Algorithm with Hard Positive and Negative Constraints

Igor Melnykov[1] · **Volodymyr Melnykov**[2]

Published online: 10 January 2020

Abstract
The paper discusses a new approach for incorporating hard constraints into the K-means algorithm for semi-supervised clustering. An analytic modification of the objective function of K-means is proposed that has not been previously considered in the literature.

Keywords K-means · Semi-supervised clustering · Hard constraints

1 Introduction

The field of cluster analysis comprises many diverse methods that locate groups of similar observations in a dataset. Clustering is usually understood as the grouping of data without any additional considerations or restrictions. We are going to adopt the view taken by the authors that refer to this kind of clustering as *unsupervised*. At the same time, the methods that are implemented in the presence of restrictions on the proposed solution or any additional supplementary information are called *semi-supervised* (Hennig et al. 2015; Yu et al. 2015; Liu and Fu 2015).

The development of methodology for grouping data under constraints goes back to DeSarbo and Mahajan (1984) who developed an algorithm for the formation of clusters with pre-determined sizes and used it to analyze the information on Forbes 500 corporations. With the advances in machine learning theory and expansion of computing capabilities, a variety of restrictions implemented in semi-supervised clustering were explored. One scenario that enjoys much attention in the literature occurs when the classification of a part of the dataset is known and may be used to determine the classification of the rest of the data (Basu et al. 2002; Barbier et al. 2012; Gu and Lu 2012). A more general framework is to consider blocks of points for which the classification may not be necessarily known, but the points are required to be joined together or separated in the clustering solution. When the points are joined together, such a constraint is called a *positive* or "must-link"

✉ Igor Melnykov
imelnyko@d.umn.edu

[1] University of Minnesota - Duluth, Duluth, MN 55812, USA

[2] The University of Alabama, Tuscaloosa, AL 35487, USA

constraint, while when the points are separated, the constraint is called *negative* or "cannot-link" (Basu et al. 2008; Śmieja and Wiercioch 2017; Ruiz et al. 2010). In addition, a variety of approaches may be taken regarding the strictness of such membership rules. If a constraint must be satisfied in the solution, it is called *hard*, while if such a constraint can be avoided (usually, at a penalty), it is called *soft* (Hennig et al. 2015).

A substantial effort has been made to accommodate various semi-supervised scenarios as a part of K-means, one of the most popular clustering algorithms (Basu et al. 2004; Wang et al. 2011; Dinler and Tural 2016). The existing modifications of the K-means algorithm oriented towards hard constraints usually take one of the following approaches. One strategy is to place the conditions consistent with the constraints on the inclusion of points in classes and check the set of restrictions during the assignment of each subsequent point (Wagstaff et al. 2001; Gu and Lu 2012). This methodology can lead to vastly different solutions depending on the order in which the points are assigned to classes as the constraints that come into play at any given time depend on the points that have already been accommodated and assigned to a class.

Some modifications of the K-means algorithm assume that the labels are known for a part of the dataset and use this information to initialize the means as well as recompute them during subsequent iterations while selecting the labels only for the portion of the dataset that has not been classified yet. Two variations of this methodology are referred to as "seeded" K-means algorithm and "constrained" K-means algorithm (Basu et al. 2002). The seeded version uses the provided data with known labeling to initialize the cluster centers at the first iteration. After this, the training data are set aside and the K-means algorithm runs essentially unchanged. The constrained method actively uses the labeled data at each iteration by taking it into account while recomputing the cluster centers. Both of these approaches have been shown to exceed the performance of the method suggested by (Wagstaff et al. 2001). Although the described methods have clear benefits, the algorithms relying on the partial labeling of the data cannot accommodate the scenario where hard constraints join or disconnect certain data points but do not insist on their membership in a particular class. In addition, when the overlap between clusters is substantial, complications of different nature may arise due to the split that occurs in the process of class formation when a part of the dataset is classified instantly, before the classification process has begun. This may lead to unintended misclassifications of large portions of data. We will further discuss this issue in Section 3.

Other suggested adjustments to the semi-supervised K-means algorithm involve labeling of a subset of data by user in multiple stages (Fatehi et al. 2014), inclusion of the information on constraints at the level of classes rather than individual data points (Liu and Fu 2015), and use of adaptive distance measures that learn from the training set (Bilenko and Mooney 2003).

Despite the active research, there are no rigorous theoretically sound approaches to semi-supervised K-means. All methods that have been proposed in literature represent ad hoc empirical algorithms that appeal to intuition but lack methodological rigor. While some of them rely on a labeled set of training data (Basu et al. 2002), the majority of such methods add an external check for possible constraint violations to the core K-means algorithm (Ruiz et al. 2010; Zhigang et al. 2013; Covões et al. 2013). In these circumstances, the algorithm often fails to converge and some authors decide not to pursue all the constraints strictly, but instead resort to minimizing the number of violated constraints (Covões et al. 2013; Davidson and Ravi 2005).

In this paper, we propose a formal modification of K-means that incorporates hard positive and negative constraints directly into the algorithm. While the suggested methodology shares the same goals as the procedures of Basu et al. (2002) or Wagstaff et al. (2001), our approach merges the restrictions completely with the K-means algorithm by modifying its objective function and making the constraints a part of the fabric of the algorithm, which was not done in the past.

We start by describing the methodology of the proposed approach in Section 2. The experimental justification is provided in Section 3 and the discussion of the results concludes the paper in Section 4.

2 Methodology

In this section, we will make a formal statement of our method and show the modifications that occur to the K-means algorithm in the presence of hard constraints.

2.1 Problem Statement and Notation

First, we are going to consider the classical unconstrained K-means algorithm. Suppose that a dataset consists of n independent p-dimensional observations $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n$. Also, suppose that the number of clusters K is pre-determined. Then, the goal of the K-means algorithm is to find the partition of the given dataset that would minimize the amount of variation within the clusters given by

$$S = \sum_{k=1}^{K} \sum_{i=1}^{n} \|\boldsymbol{y}_i - \boldsymbol{\mu}_k\|^2 I(z_i = k),$$

where $\boldsymbol{\mu}_k$ is the mean of the kth cluster, $I(\cdot)$ is the indicator function, and z_i denotes the membership of the ith observation.

Initially, the algorithm is started with a collection of centers $\hat{\boldsymbol{\mu}}_k$ that can be obtained randomly or by means of some specialized initialization techniques. Then, the method iteratively goes through two steps, where on step 1, the clusters are formed by means of assigning each observation to the cluster with the nearest center and during step 2, the new estimated cluster means $\hat{\boldsymbol{\mu}}_k$ are re-calculated according to the rule $\hat{\boldsymbol{\mu}}_k = \frac{\sum_{i=1}^{n} I(z_i=k)\boldsymbol{y}_i}{\sum_{i=1}^{n} I(z_i=k)}$. This process continues until convergence when no changes occur to the partition anymore or until it becomes clear that the convergence will not occur.

Let us now suppose that certain positive and negative restrictions have been placed on the inclusion of points in clusters. The positive constraints specify which points must belong to the same cluster in the final solution. Effectively, positive constraints define the disjoint blocks of points with the sets of indices $\mathcal{B}_b, b = 1, 2, \ldots, B$ such that $\bigcup_{b=1}^{B} \mathcal{B}_b = \{1, 2, \ldots, n\}$. Here, B is the total number of blocks defined. For any two distinct points $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$, $\{i, j \in \mathcal{B}_b\} \Rightarrow \{z_i = z_j\}$. In a trivial case, when no positive constraints are defined, each point can be viewed as a singleton block with $B = n$.

Contrary to the positive restrictions, negative constraints appear when certain points $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ cannot be in the same cluster. It should be noted that such a restriction would involve not only $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$, but the whole blocks associated with these points, i.e., $b(i) = \arg_b\{I(i \in \mathcal{B}_b) = 1\}$ and $b(j) = \arg_b\{I(j \in \mathcal{B}_b) = 1\}$. Thus, in the presence of a negative restriction, $\{r \in \mathcal{B}_{b(i)}, q \in \mathcal{B}_{b(j)}\} \Rightarrow \{z_r \neq z_q\}$.

The accommodation of either of these constraints is not straightforward in the K-means model as it requires the modification of the objective function and a mechanism that would account for more complicated structures. We will address this issue using the connection between the K-means algorithm and a modified expectation-maximization (EM) algorithm (Dempster et al. 1977), but first we proceed to give the relevant background on mixture models.

2.2 *K*-means Algorithm in a Constrained Setting

The Gaussian mixture model is given by $f(\boldsymbol{y};\boldsymbol{\theta}) = \sum_{k=1}^{K} \tau_k \phi(\boldsymbol{y};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$, where $\phi(\cdot;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$ denotes a multivariate Gaussian density with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. $\tau_k \in (0,1], k = 1,2,\ldots,K$ are mixing proportions bound by the constraint $\sum_k \tau_k = 1$. The parameters of this mixture are most often estimated with the use of the EM algorithm which consists of two iteratively repeated steps: expectation and maximization (McLachlan and Peel 2000). During the expectation step, or E-step, the conditional expected value of the complete data log-likelihood is computed, which in the case of a Gaussian mixture model amounts to finding the posterior probabilities

$$\pi_{ik} = \frac{\tau_k \phi(\boldsymbol{y}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} \tau_{k'} \phi(\boldsymbol{y}_i;\boldsymbol{\mu}_{k'},\boldsymbol{\Sigma}_{k'})}$$

that observation i originated from the kth component of the mixture. At the maximization step, or M-step, parameters τ_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$ are estimated by maximizing the conditional expectation of the complete data log-likelihood.

A modification of this method called classification EM (CEM) algorithm uses an additional classification step performed right after the E-step where each observation obtains a hard assignment to a particular cluster based on the highest posterior probability π_{ik} that was observed. Thus, $z_i = \arg\max_k \pi_{ik}$. Subsequently, at the M-step, π_{ik} are replaced by $I(z_i = k)$. This process makes the algorithm closer in spirit to the K-means method, where each data point is also unequivocally assigned to a particular cluster. In fact, it has been shown by Celeux and Govaert (1993) that the use of K-means algorithm with Euclidean distances is equivalent to implementing a CEM algorithm on a Gaussian mixture model with equal spherical covariance matrices as well as identical mixing proportions. Then, due to the restrictions on the model, $\tau_k = \frac{1}{K}$ and $\boldsymbol{\Sigma}_k = \sigma^2 \mathbf{I}$ for all $k = 1,2,\ldots,K$, where $\mathbf{I}$ is the identity matrix and σ^2 is a common variance parameter. The modification of the K-means method that would accommodate positive or negative constraints will occur at the classification stage when the assignment $z_i = \arg\max_k \pi_{ik}$ is carried out. The classification rules implemented by means of z_i will be determined by particular configurations of constraints in each case. We now proceed to obtain the expressions for π_{ik} in several configurations of constraints.

2.3 *K*-means Algorithm with Positive Constraints

Consider B blocks determined by positive constraints. It should be noted that for the set of indices $\mathcal{B}_b$ associated with block b, $\pi_{ik} = \pi_{jk}, \forall i, j \in \mathcal{B}_b$; therefore, we can define $\pi_{bk} \equiv \pi_{ik}, i \in \mathcal{B}_b$. It was shown by Melnykov et al. (2016) that

$$\pi_{bk} = \frac{\tau_k^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} \phi(\boldsymbol{y}_j;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} \tau_{k'}^{|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} \phi(\boldsymbol{y}_j;\boldsymbol{\mu}_{k'},\boldsymbol{\Sigma}_{k'})}.$$

Consequently, we can write the rule for the assignment of block b to cluster k as $\pi_{bk} \geq \pi_{bk'}$, $k' = 1, 2, \ldots, K$, which after taking logarithms and some straightforward manipulations becomes

$$\sum_{j \in \mathcal{B}_b} (\boldsymbol{y}_j - \boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{y}_j - \boldsymbol{\mu}_k) < \sum_{j \in \mathcal{B}_b} (\boldsymbol{y}_j - \boldsymbol{\mu}_{k'})' \boldsymbol{\Sigma}_{k'}^{-1} (\boldsymbol{y}_j - \boldsymbol{\mu}_{k'}) + |\mathcal{B}_b| \left[\log \frac{\tau_k^2 |\boldsymbol{\Sigma}_{k'}|}{\tau_{k'}^2 |\boldsymbol{\Sigma}_k|} \right],$$

where $|\mathcal{B}_b| = \sum_{j=1}^{n} I(j \in \mathcal{B}_b)$. $|\boldsymbol{\Sigma}_k|$ and $|\boldsymbol{\Sigma}_{k'}|$ denote the determinants of $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\Sigma}_{k'}$, respectively.

Since the assignments will now apply to whole blocks of points at once, we can define $z_b \equiv z_i, i \in \mathcal{B}_b$. Taking into account the sphericity of covariance matrices and restrictions on mixing proportions,

$$z_b = \arg\max_k \left[K^{-|\mathcal{B}_b|} \prod_{j \in \mathcal{B}_b} \phi(\boldsymbol{y}_j; \boldsymbol{\mu}_k, \sigma^2 \mathbf{I}) \right],$$

with $k = 1, 2, \ldots, K$. This expression can be further simplified to obtain the following rule for the assignment of block b to a particular class:

$$z_b = \arg\min_k \left[\sum_{j \in \mathcal{B}_b} \| \boldsymbol{y}_j - \boldsymbol{\mu}_k \|^2 \right].$$

We can see that in the presence of positive constraints, the membership of a block of points is decided by the K-means method in an intuitive way. Similarly to picking the smallest distance to a cluster center in the case of a single point, the block is assigned to a center that minimizes the sum of squared deviations from the class mean within the block. However, this membership criterion operates with the squares of Euclidean distances rather than distances themselves.

2.4 *K*-means Algorithm with Negative Constraints

We now turn to the situation where negative constraints are defined along with positive ones. Melnykov et al. (2016) described some difficulties in handling negative restrictions that arise mainly from the fact that the number of possible configurations grows very fast with the number of blocks involved. We will describe some common situations that occur with two and three blocks that can be readily generalized for a larger number of blocks.

First, let a negative constraint be defined between two blocks. Among the B blocks defined in the dataset, without the loss of generality, we can number the blocks in such a way that the negative constraint disconnects blocks 1 and 2, with the sets of indices $\mathcal{B}_1$ and $\mathcal{B}_2$, respectively. Utilizing the results of Melnykov et al. (2016), we observe that positive block relations remain in place even as negative constraints are enforced additionally.

With the kernel $\tau_k^{|\mathcal{B}_b|}\prod_{j\in\mathcal{B}_b}\phi(\boldsymbol{y}_j;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$ describing the structure of block b, the posterior probability that block 1 belongs to cluster k equals

$$\pi_{1k} = \tau_k^{|\mathcal{B}_1|}\prod_{j_1\in\mathcal{B}_1}\phi(\boldsymbol{y}_{j_1};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\sum_{\substack{s=1\\ s\neq k}}^{K}\tau_s^{|\mathcal{B}_2|}\prod_{j_2\in\mathcal{B}_2}\phi(\boldsymbol{y}_{j_2};\boldsymbol{\mu}_s,\boldsymbol{\Sigma}_s)$$
$$\times\left[\sum_{k'=1}^{K}\tau_{k'}^{|\mathcal{B}_1|}\prod_{j_1\in\mathcal{B}_1}\phi(\boldsymbol{y}_{j_1};\boldsymbol{\mu}_{k'},\boldsymbol{\Sigma}_{k'})\sum_{\substack{s'=1\\ s'\neq k'}}^{K}\tau_{s'}^{|\mathcal{B}_2|}\prod_{j_2\in\mathcal{B}_2}\phi(\boldsymbol{y}_{j_2};\boldsymbol{\mu}_{s'},\boldsymbol{\Sigma}_{s'})\right]^{-1}.$$

Then, it follows that the membership of block 1 is determined by

$$z_1 = \arg\max_k\left[\tau_k^{|\mathcal{B}_1|}\prod_{j_1\in\mathcal{B}_1}\phi(\boldsymbol{y}_{j_1};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\times\left(\sum_{\substack{s=1\\ s\neq k}}^{K}\tau_s^{|\mathcal{B}_2|}\prod_{j_2\in\mathcal{B}_2}\phi\left(\boldsymbol{y}_{j_2};\boldsymbol{\mu}_s,\boldsymbol{\Sigma}_s\right)\right)\right],$$

which in turn can be re-written as

$$z_1 = \arg\max_k\left[\sum_{s=1,s\neq k}^{K}\exp\left(-\frac{1}{2\sigma^2}\left(\sum_{j_1\in\mathcal{B}_1}\|\boldsymbol{y}_{j_1}-\boldsymbol{\mu}_k\|^2+\sum_{j_2\in\mathcal{B}_2}\|\boldsymbol{y}_{j_2}-\boldsymbol{\mu}_s\|^2\right)\right)\right],$$

where we once again utilized the sphericity of covariance matrices and equality of mixing proportions. The corresponding expressions for block 2 are completely symmetric.

Similarly, in the case of three blocks represented by $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$, where negative constraints are established for each pair, i.e., all three blocks must belong to different clusters in the solution,

$$z_1 = \arg\max_k\left[\sum_{\substack{s=1,\\ s\neq k}}^{K}\sum_{\substack{h=1,\\ h\neq k,h\neq s}}^{K}\exp\left(-\frac{1}{2\sigma^2}\left(\sum_{j_1\in\mathcal{B}_1}\|\boldsymbol{y}_{j_1}-\boldsymbol{\mu}_k\|^2\right.\right.\right.$$
$$\left.\left.\left.+\sum_{j_2\in\mathcal{B}_2}\|\boldsymbol{y}_{j_2}-\boldsymbol{\mu}_s\|^2+\sum_{j_3\in\mathcal{B}_3}\|\boldsymbol{y}_{j_3}-\boldsymbol{\mu}_h\|^2\right)\right)\right].$$

Thanks to the symmetry, z_2 and z_3 are determined by similar expressions with straightforward index adjustments.

3 Experiments and Illustrations

3.1 Motivating Example

We begin by considering an example that will emphasize the flexibility of our method and show the advantage of defining block restrictions over the direct labeling of points. Suppose that there are two classes of individuals overlapping quite substantially with one another. Thus, the two classes may represent two species of animals that, in principle, could be

identified with high degree of certainty by an expert. An example of such process is detailed in Nimmo et al. (2018) in relation to the identification of insects that belong to family *Chironomidae*. The identification of these insects involves their dissection on a microscopic scale, exposure to specialized chemical solutions, and further slide preparation that takes 4 weeks. For the classification described in Nimmo et al. (2018), the slides had to be mailed to an expert located off-site. Alternatively, the identification can be made by observing morphological characteristics of *Chironomidae* flies such as their length, measurements of antennae, leg segments, and other parts of the body. The latter method is not error-proof, but much less demanding in terms of time and effort. To train the clustering algorithm, a small group of individuals can be first identified by an expert using the classical technique involving the microscopic slide preparation. At this point, the algorithms such as seeded K-means or constrained K-means will assign labels to the points that have been identified. The identification of *Chironomidae* can involve a large number of species; however, for illustrative purposes, we will focus on only two classes in this example.

Figure 1 shows a two-dimensional dataset generated from two such classes of points with $n = 200$. Both clusters here are spherical and have approximately equal proportions, making this dataset favorable for classification by the K-means algorithm. Three points in each class have been identified and this information is used to start the K-means algorithm. In Fig. 1, these points are shown circled and connected by the lines of corresponding color. Note that the points used for training were all picked in the area of overlap between the clusters where misclassifications can easily occur. We implement the constrained K-means approach with these starting conditions and observe that in the estimated classification the classes are essentially flipped. As a result, the majority of animals that belong to species 1 will be classified as species 2, while those from species 2 will be identified as species 1. Being driven to maintain the labels of the points in the training set, this method does so at the expense of the vast majority of the observations.

This scenario is less likely to occur if a large random sample of points is chosen for training. At the same time, the described situation is not far-fetched if one takes it into account that the observations used in the training set are often the ones that are readily available to the researcher and are not selected at random. On the surface, it may also seem advantageous to determine the classification of points in the "gray" area with an intent to equip the algorithm with the tools to classify observations in the most difficult circumstances. However, such a strategy can easily misfire as shown by our example. We will explore this phenomenon in more detail in the simulations with a larger number of clusters.

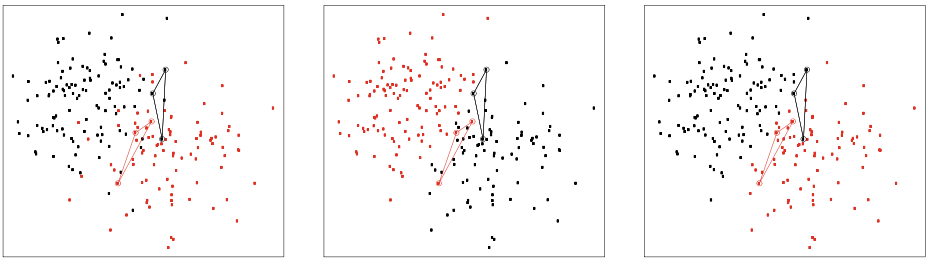


Fig. 1 Differences between the true (**a**) and estimated classifications using the constrained K-means algorithm (**b**) and the proposed method (**c**). The points with known classification are shown via connecting lines

3.2 Experiments

We now proceed to consider a variety of practical situations with different configurations of positive and negative constraints. Where appropriate, we consider a comparison with the constrained K-means and seeded K-means approaches of Basu et al. (2002) that are often used as benchmarks for other algorithms (Gu and Lu 2012; Fatehi et al. 2014). In addition, in the situations where negative constraints are defined among all specified blocks the constrained K-means approach is close in spirit to our proposed method. R package MixSim (Melnykov et al. 2012) was used to simulate datasets with varying degrees of overlap and clustering complexity.

3.2.1 Ten Clusters with Positive and Negative Constraints in Place

We first consider a simulation with $K = 10$ classes where ten blocks were defined in such a manner that each class has exactly one block within it and all blocks are separated by negative constraints. This setup fits well with the constrained K-means and seeded K-means methods of Basu et al. (2002), but such a configuration captures only one in a variety of possible scenarios that our proposed method is capable of handling.

In addition, a "naive" semi-supervised clustering approach was realized where a randomly chosen training set was utilized for finding the initial mean in each of the ten clusters. Then, each of the remaining points was assigned to the cluster with the nearest mean, but no repeated recalculation of means and reassignment of points took place. This approach amounts to one iteration of the K-means algorithm.

To generate the datasets of varying complexity, the number of dimensions p was taken at 5, 10, and 20 and the average pairwise overlap $\bar{\omega}$ was set at 0.01 and 0.10. Here, the pairwise overlap is defined as the sum of misclassification probabilities when two clusters are considered at a time (Maitra and Melnykov 2010). For each combination of p and $\bar{\omega}$, 100 datasets of $n = 10,000$ points each were generated with approximately equal allocations of points to the ten clusters. The training set was obtained by randomly extracting a fraction of points in each class.

The results of the simulation study are summarized in Tables 1 and 2 where the proposed method is labeled as "ss$K$-means" (for "semi-supervised K-means"). In Table 1, training percentages varied between 1 and 7% in 2% increments, while in Table 2 larger percentages between 10 and 70% were considered.

It can be seen that all four methods performed roughly the same with the "naive" method falling behind somewhat at lower training percentages. This effect is maintained for both lower and larger values of the average pairwise overlap and across different dimensionality values. Thus, the proposed method has been shown to be quite viable in the simulated scenario with the training sets picked at random. It is worth noting that in practice the training sets are often formed not at random, but based on other considerations such as availability of points or their perceived usefulness in classification of future observations.

3.2.2 Selecting a Training Set in the Overlap Area

We now proceed to evaluate some less straightforward clustering situations that will showcase the advantages of our proposed method. First, let us extend the scenario described in Section 3.1 to a larger number of clusters and consider how the choice of points for training affects the proportion of correct classifications. We consider $K = 4$ two-dimensional clusters with the average pairwise overlap $\bar{\omega}$ equal to 0.2. Each dataset consists of the total of

Table 1 Smaller training percentages

			$p = 5$		$p = 10$		$p = 20$	
			$\bar{\omega} = 0.01$	$\bar{\omega} = 0.1$	$\bar{\omega} = 0.01$	$\bar{\omega} = 0.1$	$\bar{\omega} = 0.01$	$\bar{\omega} = 0.1$
ss$K$-means	1%	$\mathcal{M}$	0.9583	0.7234	0.9614	0.7446	0.9635	0.7555
		IQR	0.0054	0.0132	0.0034	0.0084	0.0023	0.0068
	3%	$\mathcal{M}$	0.9596	0.7296	0.9621	0.7504	0.9643	0.7603
		IQR	0.0039	0.0121	0.0029	0.0085	0.0021	0.0071
	5%	$\mathcal{M}$	0.9604	0.7354	0.9630	0.7550	0.9649	0.7650
		IQR	0.0036	0.0122	0.0027	0.0086	0.0025	0.0065
	7%	$\mathcal{M}$	0.9613	0.7420	0.9638	0.7602	0.9657	0.7700
		IQR	0.0032	0.0121	0.0028	0.0089	0.0024	0.0061
Constrained K-means	1%	$\mathcal{M}$	0.9588	0.7219	0.9615	0.7451	0.9635	0.7553
		IQR	0.0039	0.0132	0.0030	0.0090	0.0023	0.0068
	3%	$\mathcal{M}$	0.9595	0.7282	0.9623	0.7498	0.9643	0.7599
		IQR	0.0036	0.0126	0.0029	0.0082	0.0022	0.0067
	5%	$\mathcal{M}$	0.9602	0.7350	0.9629	0.7551	0.9651	0.7648
		IQR	0.0033	0.0122	0.0029	0.0086	0.0022	0.0062
	7%	$\mathcal{M}$	0.9612	0.7408	0.9639	0.7598	0.9658	0.7692
		IQR	0.0032	0.0122	0.0028	0.0088	0.0023	0.0063
Seeded K-means	1%	$\mathcal{M}$	0.9588	0.7226	0.9615	0.7449	0.9635	0.7552
		IQR	0.0039	0.0130	0.0030	0.0090	0.0023	0.0070
	3%	$\mathcal{M}$	0.9595	0.7282	0.9623	0.7500	0.9643	0.7597
		IQR	0.0036	0.0127	0.0029	0.0088	0.0022	0.0068
	5%	$\mathcal{M}$	0.9602	0.7350	0.9629	0.7551	0.9650	0.7644
		IQR	0.0033	0.0123	0.0028	0.0086	0.0022	0.0062
	7%	$\mathcal{M}$	0.9612	0.7409	0.9639	0.7599	0.9658	0.7693
		IQR	0.0032	0.0118	0.0028	0.0085	0.0023	0.0061
Naive K-means	1%	$\mathcal{M}$	0.9496	0.6866	0.9459	0.6856	0.9384	0.6500
		IQR	0.0056	0.0228	0.0056	0.0165	0.0062	0.0176
	3%	$\mathcal{M}$	0.9568	0.7215	0.9575	0.7315	0.9565	0.7242
		IQR	0.0037	0.0161	0.0037	0.0099	0.0028	0.0083
	5%	$\mathcal{M}$	0.9588	0.7312	0.9602	0.7449	0.9605	0.7455
		IQR	0.0031	0.0143	0.0034	0.0084	0.0024	0.0066
	7%	$\mathcal{M}$	0.9599	0.7379	0.9618	0.7541	0.9630	0.7571
		IQR	0.0030	0.0124	0.0030	0.0081	0.0025	0.0056

Results of the simulation study for $K = 10$. $\mathcal{M}$ and IQR represent the median and interquartile range of the proportion of correct classifications, p is the dimensionality, and $\bar{\omega}$ is the average pairwise overlap between mixture components

$n = 500$ points with equal proportions among the four classes that were generated. We have seen in Section 3.1 that the choice of a training set in the "gray" area carries a substantial misclassification risk. Let $f_k(\boldsymbol{x})$ and $f_{k'}(\boldsymbol{x})$, where $k, k' = 1, ..., K, k \neq k'$, be two of the Gaussian pdfs used by MixSim to generate the classes of points and let the "gray" area be

Table 2 Larger training percentages

			$p = 5$		$p = 10$		$p = 20$	
			$\bar{\omega} = 0.01$	$\bar{\omega} = 0.1$	$\bar{\omega} = 0.01$	$\bar{\omega} = 0.1$	$\bar{\omega} = 0.01$	$\bar{\omega} = 0.1$
ss$K$-means	10%	$\mathcal{M}$	0.9624	0.7495	0.9650	0.7687	0.9668	0.7773
		IQR	0.0030	0.0130	0.0029	0.0084	0.0021	0.0062
	30%	$\mathcal{M}$	0.9707	0.8064	0.9729	0.8206	0.9740	0.8271
		IQR	0.0027	0.0104	0.0022	0.0062	0.0021	0.0053
	50%	$\mathcal{M}$	0.9790	0.8620	0.9807	0.8727	0.9814	0.8771
		IQR	0.0023	0.0071	0.0017	0.0050	0.0016	0.0034
	70%	$\mathcal{M}$	0.9873	0.9173	0.9883	0.9233	0.9889	0.9265
		IQR	0.0019	0.0050	0.0013	0.0041	0.0012	0.0033
Constrained K-means	10%	$\mathcal{M}$	0.9624	0.7485	0.9650	0.7678	0.9669	0.7767
		IQR	0.0031	0.0131	0.0030	0.0084	0.0021	0.0058
	30%	$\mathcal{M}$	0.9706	0.8045	0.9728	0.8193	0.9739	0.8260
		IQR	0.0029	0.0113	0.0022	0.0063	0.0021	0.0051
	50%	$\mathcal{M}$	0.9788	0.8587	0.9806	0.8710	0.9812	0.8752
		IQR	0.0023	0.0071	0.0018	0.0061	0.0017	0.0039
	70%	$\mathcal{M}$	0.9872	0.9149	0.9883	0.9213	0.9887	0.9240
		IQR	0.0018	0.0052	0.0013	0.0038	0.0012	0.0031
Seeded K-means	10%	$\mathcal{M}$	0.9624	0.7484	0.9650	0.7677	0.9669	0.7767
		IQR	0.0030	0.0130	0.0030	0.0085	0.0021	0.0061
	30%	$\mathcal{M}$	0.9706	0.8045	0.9728	0.8194	0.9739	0.8260
		IQR	0.0029	0.0109	0.0022	0.0063	0.0021	0.0049
	50%	$\mathcal{M}$	0.9788	0.8589	0.9806	0.8709	0.9812	0.8751
		IQR	0.0023	0.0070	0.0018	0.0059	0.0017	0.0040
	70%	$\mathcal{M}$	0.9872	0.9147	0.9882	0.9213	0.9887	0.9240
		IQR	0.0018	0.0051	0.0013	0.0045	0.0012	0.0031
Naive K-means	10%	$\mathcal{M}$	0.9618	0.7482	0.9636	0.7653	0.9649	0.7696
		IQR	0.0030	0.0130	0.0029	0.0081	0.0026	0.0053
	30%	$\mathcal{M}$	0.9705	0.8067	0.9727	0.8202	0.9738	0.8262
		IQR	0.0024	0.0097	0.0023	0.0063	0.0018	0.0052
	50%	$\mathcal{M}$	0.9788	0.8623	0.9807	0.8726	0.9813	0.8768
		IQR	0.0022	0.0073	0.0018	0.0053	0.0016	0.0042
	70%	$\mathcal{M}$	0.9873	0.9171	0.9883	0.9231	0.9888	0.9263
		IQR	0.0019	0.0049	0.0014	0.0039	0.0014	0.0030

Results of the simulation study for $K = 10$. $\mathcal{M}$ and IQR represent the median and interquartile range of the proportion of correct classifications, p is the dimensionality, and $\bar{\omega}$ is the average pairwise overlap between mixture components

defined by the choice of a constant $C > 1$ to have $C^{-1} \leq \frac{f_k(\boldsymbol{x})}{f_{k'}(\boldsymbol{x})} \leq C$. Figure 2 shows the proportion of correct classifications for the proposed method (green dots) and constrained K-means (blue triangles) as a function of C. The average proportions were determined over 10,000 runs for 3, 5, 7, and 9 points in the training set.

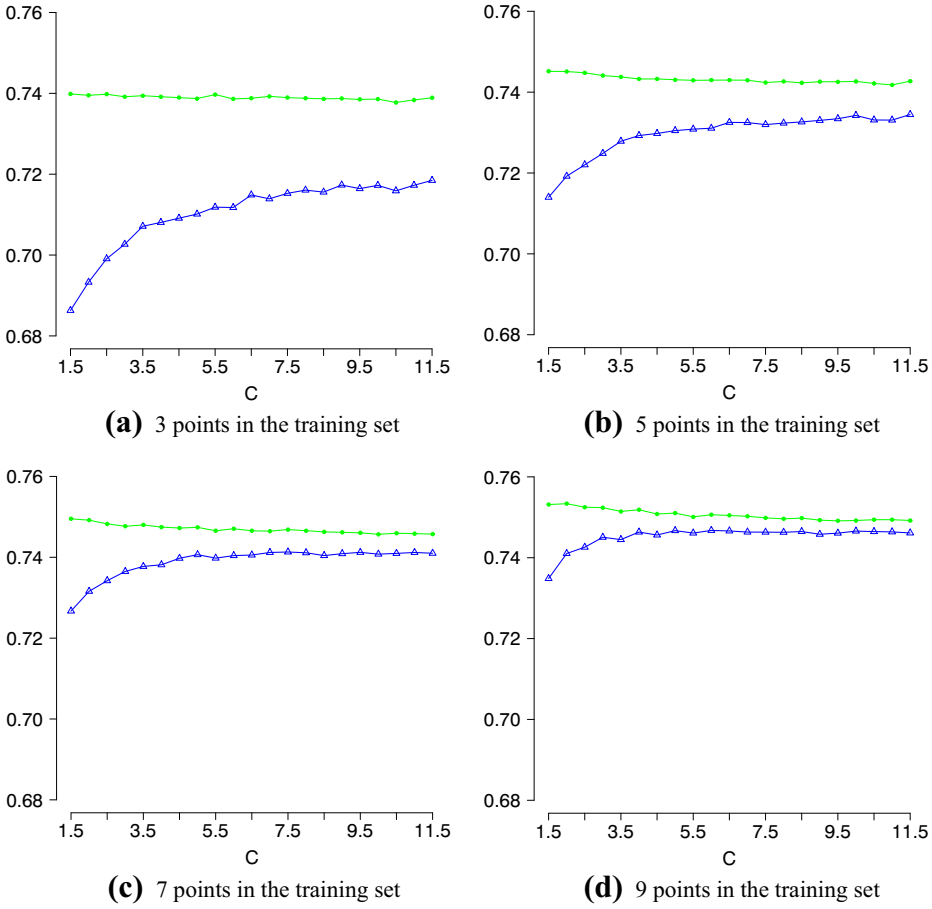


Fig. 2 **a–d** Proportion of correct classifications versus the ratio of densities (C) defining the "gray" area for the proposed method (dots) and constrained K-means (triangles)

While the proposed method shows a stable performance for different training set sizes and even leads to slightly better outcomes for lower values of C, the constrained K-means algorithm experiences problems if the size of the training set is small and especially when the points for training are picked in the area of high overlap. For $C < 3$, it has lower classification proportions even when the training set is obtained from 9 randomly selected points that represent about 7% of the given class and are classified without error.

It is worth noting that a similar effect is observed if one or more of the points in the training set are misclassified. When the labels are determined for the points in the training set, it is typically assumed that these labels are 100% error-free. However, in practice, there is a possibility that a point may be misidentified even by an expert or some other error may occur. We have simulated $n = 10,000$ points in equal proportions between $K = 2$ two-dimensional clusters with the pairwise overlap of 0.2 and generated at random two blocks of three points with known labels in each class. In the absence of errors, both the proposed method and constrained K-means yield 0.900 rate of correct classifications over 1000 simulations.

Now to determine the effect that an incorrect label may have, we kept one of the blocks error-free, while in the second one we included one point that was misidentified and in reality came from the opposite class. Since our method does not rely on direct labeling, its proportion of correct classifications stands unaffected at 0.900 due to the negligible effect that one point has on a set of 10,000 data. At the same time, the constrained K-means method sees its rate go down to 0.882.

To summarize, if the assignment of labels to observations is done in two stages, caution needs to be exercised due to a real possibility of misclassifying substantial portions of data, especially, if the majority of training points are located in the area of cluster overlap. Consequently, it is beneficial to include some points with the easy straightforward classification in the training set. The proposed algorithm would overcome this obstacle by holding off the assignment of labels to the points comprising each block until the stage in the algorithm when all observations receive their labels.

3.2.3 Positive and Negative Constraint Configurations

As was pointed out in the beginning of Section 3.2, the proposed method can be used to model a variety of situations involving both positive and negative constraints not accessible to methods that rely on the direct labeling of points involved in constraints. For the following illustrations, we consider the simulations on the two-dimensional datasets generated from $K = 4$ classes with $n = 500$ and $\bar{\omega} = 0.1$. A typical dataset of this nature is shown in Fig. 3.

A configuration of blocks consistent with that of Section 3.2.1 would involve a block from each class defined by means of positive constraints and a set of negative constraints disconnecting all blocks from one another. This configuration is shown in Fig. 4a with three points per block. The symbols defining each block, such as circles, squares, triangles and diamonds, are different to indicate the negative constraints in place. Below, we will adhere

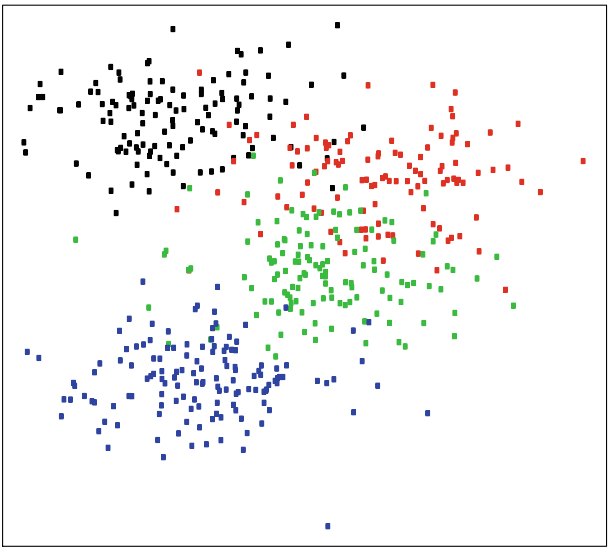
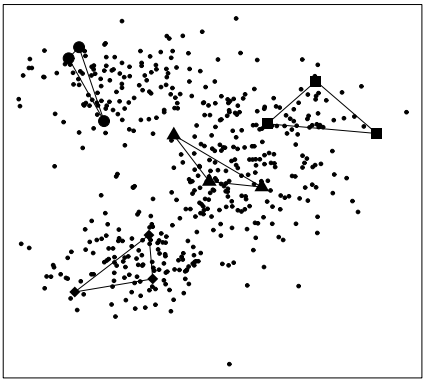
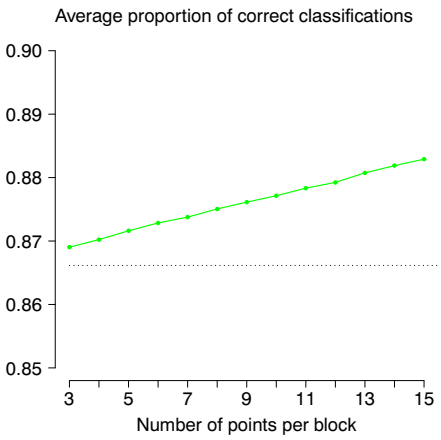
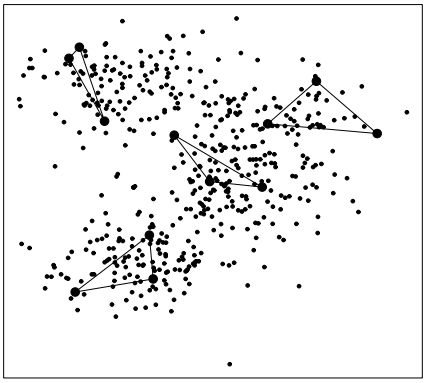


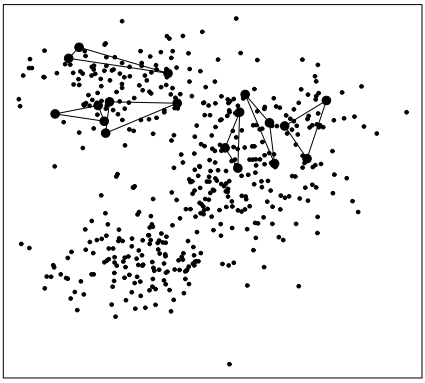
Fig. 3 True classification in a dataset with $K = 4$, $p = 2$, $\bar{\omega} = 0.10$



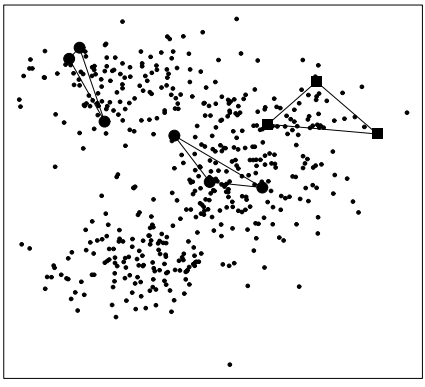


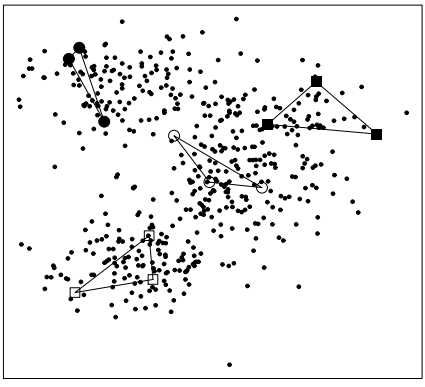
(a) **(b)**





(c) **(d)**





(e) **(f)**

Fig. 4 **a**–**f** Blocks with various configurations of constraints defined over four classes

to the same conventions by showing positive constraints as connecting lines and indicating negative constraints by using different symbols in the respective blocks. Figure 4b shows the proportion of correct classifications as a function of the number of points in each block. For the subsequent configurations of blocks, we omit the graphs that show correct classification proportions as they are very similar to (b).

Figure 4c shows a similar situation where only positive constraints are in place and no negative constraints are defined. Thus, in principle, the blocks that were defined can be allocated to the same class in the solution. The points participating in the four blocks are all shown by circles to indicate the lack of negative constraints.

Another possibility is for multiple blocks to be defined within the same class. Thus, in Fig. 4d, there were three blocks defined in the first and second classes but none were defined in the remaining two. Also, no negative constraints were imposed.

Figure 4e shows three blocks that were set up in three different classes in such a way that one of them, shown with squares, is separated from the other two by negative constraints. However, those two remaining blocks, both shown by circles, do not have a negative constraint defined between them. We use this opportunity to illustrate the construction of the membership function for multiple blocks. Let $\mathcal{B}_1$ and $\mathcal{B}_3$ represent the blocks marked with circles and $\mathcal{B}_2$ represent the block marked with squares in Fig. 4e. Then, for $\mathcal{B}_1$,

$$z_1 = \arg\max_k \left[\sum_{\substack{s=1,\\ s\neq k}}^{K} \sum_{\substack{h=1,\\ h\neq s}}^{K} \exp\left(-\frac{1}{2\sigma^2} \left(\sum_{j_1 \in \mathcal{B}_1} \|\boldsymbol{y}_{j_1} - \boldsymbol{\mu}_k\|^2 + \sum_{j_2 \in \mathcal{B}_2} \|\boldsymbol{y}_{j_2} - \boldsymbol{\mu}_s\|^2 + \sum_{j_3 \in \mathcal{B}_3} \|\boldsymbol{y}_{j_3} - \boldsymbol{\mu}_h\|^2 \right) \right) \right].$$

The key to implementing negative constraints here is the careful treatment of the sums over s and h. The sum over s is matched with $\mathcal{B}_2$ and thus needs an exclusion for k, the summation index matched with $\mathcal{B}_1$. At the same time, the sum over h is matched with $\mathcal{B}_3$ and requires an exclusion for s, the index matched with $\mathcal{B}_2$.

The membership function for block $\mathcal{B}_2$ also needs to reflect the fact that $\mathcal{B}_2$ is separated from the other two blocks, while $\mathcal{B}_1$ and $\mathcal{B}_3$ do not have a negative constraint between them:

$$z_2 = \arg\max_k \left[\sum_{\substack{s=1,\\ s\neq k}}^{K} \sum_{\substack{h=1,\\ h\neq k}}^{K} \exp\left(-\frac{1}{2\sigma^2} \left(\sum_{j_1 \in \mathcal{B}_1} \|\boldsymbol{y}_{j_1} - \boldsymbol{\mu}_s\|^2 + \sum_{j_2 \in \mathcal{B}_2} \|\boldsymbol{y}_{j_2} - \boldsymbol{\mu}_k\|^2 + \sum_{j_3 \in \mathcal{B}_3} \|\boldsymbol{y}_{j_3} - \boldsymbol{\mu}_h\|^2 \right) \right) \right].$$

The membership function for $\mathcal{B}_3$ is analogous to that of $\mathcal{B}_1$ with appropriate index adjustments.

Finally, Fig. 4f shows a configuration of four blocks where only two negative constraints were defined. The blocks shown with solid circles ($\mathcal{B}_1$) and solid squares ($\mathcal{B}_2$) will be kept separate, but either of them may end up in the same class with the blocks shown by the blank symbols. Similarly, the blocks shown with blank circles ($\mathcal{B}_3$) and blank squares ($\mathcal{B}_4$) have a negative constraint only between the two of them. We present the membership function for $\mathcal{B}_3$, while other membership functions will be similar due to the fact that each block has only one negative constraint in place separating it from one other block:

$$z_3 = \arg\max_k \left[\sum_{s=1}^{K} \sum_{\substack{h=1,\\ h\neq s}}^{K} \sum_{\substack{t=1,\\ t\neq k}}^{K} \exp\left(-\frac{1}{2\sigma^2} \left(\sum_{j_1\in\mathcal{B}_1} \|\boldsymbol{y}_{j_1} - \boldsymbol{\mu}_s\|^2 + \sum_{j_2\in\mathcal{B}_2} \|\boldsymbol{y}_{j_2} - \boldsymbol{\mu}_h\|^2 + \sum_{j_3\in\mathcal{B}_3} \|\boldsymbol{y}_{j_3} - \boldsymbol{\mu}_k\|^2 + \sum_{j_4\in\mathcal{B}_4} \|\boldsymbol{y}_{j_4} - \boldsymbol{\mu}_t\|^2 \right) \right) \right].$$

Overall, the proposed method offers much needed flexibility in modeling positive or negative constraints. We once again emphasize that the configurations illustrated in Fig. 4c–f could not be accommodated by the methods that require hard class assignments of the points involved in the constraints.

3.2.4 Comparison with a Model-Based Semi-supervised Method

The proposed method enjoys the advantages associated with the K-means algorithm, such as the relative simplicity and ease of practical implementation if compared to model-based approaches. At the same time, it is worth exploring how ss$K$-means compares to potentially more flexible models, for example, those involving the use of finite Gaussian mixtures. In this simulation study, we focus on the comparison of our algorithm with the semi-supervised method of Melnykov et al. (2016).

The datasets were generated from mixture models with $K = 4$ and $K = 10$ and the sample size was chosen to be $100K$. The number of dimensions p was taken equal to 5, 10, and 20, while the maximum overlap $\breve{\omega}$ was set at 0.01 and 0.1. In each class, one block of points was defined at random. As discussed in previous sections, the observations within each block are bound together by positive constraints and, in principle, the blocks drawn from different classes could be clustered together. The sizes of blocks $|B|$ equal to 1, 10, and 25 were evaluated. Here, $|B| = 1$ represents a trivial case of unsupervised clustering with each block being a singleton. Finally, to evaluate the effect that cluster shapes have on the outcomes, all simulations were performed for clusters of elliptical as well as spherical shapes. Of course, the Gaussian mixtures are expected to outperform ss$K$-means in the case of elliptical clusters. On the contrary, for the second scenario when spherical clusters are simulated under the assumptions of equal covariance matrices, it is our expectation that ss$K$-means will be highly competitive with semi-supervised Gaussian mixtures. Each simulation scenario was repeated for 250 mixtures generated by MixSim.

As expected, the model-based method performed noticeably better on elliptical clusters, especially, in the presence of a larger overlap ($\breve{\omega} = 0.1$) (Table 3). At the same time, the ss$K$-means algorithm was not far behind, especially, in the situations when larger blocks were used. This emphasizes the positive effect that the introduction of extra information in the form of constraints has on successful classification.

Table 3 Results of the simulation study for elliptical clusters with $K = 4$ and $K = 10$

				$p = 5$		$p = 10$		$p = 20$	
				$\breve{\omega} = 0.01$	$\breve{\omega} = 0.1$	$\breve{\omega} = 0.01$	$\breve{\omega} = 0.1$	$\breve{\omega} = 0.01$	$\breve{\omega} = 0.1$
ss$K$-means	$K = 4$	$\lvert B\rvert = 1$	$\mathcal{M}$	0.9900	0.9150	0.9875	0.8625	0.9800	0.7225
			IQR	0.0075	0.0500	0.0100	0.0600	0.0150	0.1044
		$\lvert B\rvert = 10$	$\mathcal{M}$	0.9925	0.9275	0.9900	0.8800	0.9825	0.7663
			IQR	0.0075	0.0444	0.0075	0.0550	0.0125	0.0825
		$\lvert B\rvert = 25$	$\mathcal{M}$	0.9950	0.9425	0.9925	0.9025	0.9850	0.8200
			IQR	0.0075	0.0350	0.0075	0.0419	0.0100	0.0550
	$K = 10$	$\lvert B\rvert = 1$	$\mathcal{M}$	0.9950	0.9395	0.9930	0.8805	0.9815	0.6495
			IQR	0.0050	0.0520	0.0070	0.0753	0.0130	0.1320
		$\lvert B\rvert = 10$	$\mathcal{M}$	0.9950	0.9470	0.9940	0.8950	0.9840	0.7155
			IQR	0.0048	0.0425	0.0068	0.0633	0.0120	0.1060
		$\lvert B\rvert = 25$	$\mathcal{M}$	0.9960	0.9575	0.9950	0.9135	0.9860	0.7745
			IQR	0.0040	0.0358	0.0050	0.0530	0.0100	0.0680

Table 3 (continued)

				$p = 5$		$p = 10$		$p = 20$	
				$\breve{\omega} = 0.01$	$\breve{\omega} = 0.1$	$\breve{\omega} = 0.01$	$\breve{\omega} = 0.1$	$\breve{\omega} = 0.01$	$\breve{\omega} = 0.1$
Semi-supervised model-based method	$K = 4$	$\lvert B\rvert = 1$	$\mathcal{M}$	0.9950	0.9475	0.9930	0.9050	0.9815	0.7600
			IQR	0.0050	0.0300	0.0070	0.0475	0.0130	0.1119
		$\lvert B\rvert = 10$	$\mathcal{M}$	0.9975	0.9550	0.9950	0.9175	0.9875	0.7925
			IQR	0.0025	0.0275	0.0075	0.0375	0.0100	0.0919
		$\lvert B\rvert = 25$	$\mathcal{M}$	0.9975	0.9625	0.9950	0.9338	0.9900	0.8450
			IQR	0.0050	0.0219	0.0050	0.0319	0.0100	0.0675
	$K = 10$	$\lvert B\rvert = 1$	$\mathcal{M}$	0.9980	0.9645	0.9970	0.9115	0.9880	0.6815
			IQR	0.0020	0.0300	0.0030	0.0638	0.0090	0.1418
		$\lvert B\rvert = 10$	$\mathcal{M}$	0.9980	0.9690	0.9970	0.9235	0.9890	0.7230
			IQR	0.0020	0.0268	0.0040	0.0535	0.0080	0.1255
		$\lvert B\rvert = 25$	$\mathcal{M}$	0.9990	0.9750	0.9980	0.9400	0.9910	0.7835
			IQR	0.0020	0.0200	0.0030	0.0408	0.0060	0.1053

$\mathcal{M}$ and IQR represent the median and interquartile range of the proportion of correct classifications, p is the dimensionality, and $\breve{\omega}$ is the maximum pairwise overlap between mixture components. $|B|$ denotes the number of points in a block drawn from each class

Table 4 Results of the simulation study for spherical clusters with $K = 4$ and $K = 10$

				$p = 5$		$p = 10$		$p = 20$	
				$\breve{\omega} = 0.01$	$\breve{\omega} = 0.1$	$\breve{\omega} = 0.01$	$\breve{\omega} = 0.1$	$\breve{\omega} = 0.01$	$\breve{\omega} = 0.1$
ss$K$-means	$K = 4$	$\lvert B\rvert = 1$	$\mathcal{M}$	0.9975	0.9550	0.9975	0.9425	0.9950	0.9250
			IQR	0.0050	0.0300	0.0044	0.0325	0.0050	0.0319
		$\lvert B\rvert = 10$	$\mathcal{M}$	0.9975	0.9600	0.9975	0.9500	0.9950	0.9325
			IQR	0.0050	0.0250	0.0025	0.0300	0.0050	0.0275
		$\lvert B\rvert = 25$	$\mathcal{M}$	0.9975	0.9675	0.9975	0.9575	0.9963	0.9438
			IQR	0.0050	0.0225	0.0050	0.0250	0.0050	0.0250
	$K = 10$	$\lvert B\rvert = 1$	$\mathcal{M}$	0.9990	0.9740	0.9980	0.9515	0.9960	0.9080
			IQR	0.0010	0.0210	0.0030	0.0320	0.0040	0.0428
		$\lvert B\rvert = 10$	$\mathcal{M}$	0.9990	0.9770	0.9980	0.9570	0.9960	0.9170
			IQR	0.0010	0.0188	0.0020	0.0295	0.0040	0.0370
		$\lvert B\rvert = 25$	$\mathcal{M}$	0.9990	0.9800	0.9980	0.9640	0.9970	0.9310
			IQR	0.0020	0.0160	0.0020	0.0240	0.0030	0.0318

Table 4 (continued)

				$p = 5$		$p = 10$		$p = 20$	
				$\check{\omega} = 0.01$	$\check{\omega} = 0.1$	$\check{\omega} = 0.01$	$\check{\omega} = 0.1$	$\check{\omega} = 0.01$	$\check{\omega} = 0.1$
Semi-supervised model-based method	$K = 4$	$\lvert B\rvert = 1$	$\mathcal{M}$	0.9975	0.9500	0.9975	0.9300	0.9950	0.9075
			IQR	0.0050	0.0300	0.0050	0.0400	0.0075	0.0375
		$\lvert B\rvert = 10$	$\mathcal{M}$	0.9975	0.9550	0.9950	0.9375	0.9950	0.9175
			IQR	0.0025	0.0300	0.0050	0.0350	0.0075	0.0325
		$\lvert B\rvert = 25$	$\mathcal{M}$	0.9975	0.9625	0.9975	0.9475	0.9950	0.9300
			IQR	0.0050	0.0250	0.0025	0.0300	0.0050	0.0300
	$K = 10$	$\lvert B\rvert = 1$	$\mathcal{M}$	0.9980	0.9710	0.9970	0.9430	0.9950	0.8920
			IQR	0.0020	0.0250	0.0028	0.0370	0.0050	0.0510
		$\lvert B\rvert = 10$	$\mathcal{M}$	0.9980	0.9760	0.9970	0.9500	0.9950	0.9040
			IQR	0.0020	0.0218	0.0030	0.0340	0.0040	0.0438
		$\lvert B\rvert = 25$	$\mathcal{M}$	0.9990	0.9785	0.9980	0.9580	0.9960	0.9210
			IQR	0.0010	0.0170	0.0030	0.0260	0.0040	0.0390

The composition of the table is identical to that of Table 3

On spherical clusters with equal covariance matrices, the K-means algorithm was expectedly very competitive. In fact, it was even more successful than the model-based clustering method in the vast majority of situations as shown in Table 4. A similar phenomenon was pointed out by Steinley and Brusco (2011) in their study of unsupervised clustering. In our semi-supervised setting, this effect is observed across different degrees of overlap, number of dimensions, and block sizes in positive constraints. Even though our developed K-means-based method is expected to perform well under such circumstances, the fact that the proposed method is capable of showing better performance than its model-based counterpart is quite noteworthy.

4 Discussion

The methodology described in the paper allows to accommodate both positive and negative constraints in semi-supervised clustering by making them a part of the K-means algorithm framework. The proposed method performs well in the situations that are generally favorable for the use of the K-means algorithm, in particular, when the clusters are roughy spherical and have approximately equal representations.

The novelty of the proposed approach is in making both kinds of hard constraints an organic part of this clustering algorithm, while the methods proposed to date either verify the restrictions concurrently with executing the algorithm or use the labeled data at the training stage of the algorithm. Another advantage of the proposed method is the fact that it does not rely on the training data to be labeled as belonging to a certain class before the algorithm starts, since the placement of blocks into different classes is ensured by means of negative constraints. Thus, the process does not prohibit the training data from being labeled in advance but does not rely on such labeling. As a result, a larger variety of clustering situations can be accommodated compared to the methods that rely on direct labeling.

References

Barbier, G., Zafarani, R., Gao, H., Fung, G., Liu, H. (2012). Maximizing benefits from crowdsourced data. *Computational and Mathematical Organization Theory*, *18*, 257–279.

Basu, S., Banerjee, A., Mooney, R. (2002). Semi-supervised clustering by seeding. In *Proceedings of the 19th international conference on machine learning* (pp. 19–26).

Basu, S., Banerjee, A., Mooney, R. (2004). Active semi-supervision for pairwise constrained clustering. In *Proceedings of the SIAM international conference on data mining*.

Basu, S., Davidson, I., Wagstaff, K. (2008). *Constrained clustering: advances in algorithms, theory, and applications*. Boca Raton: CRC Press.

Bilenko, M., & Mooney, J.R. (2003). Adaptive duplicate detection using learnable string similarity measures. In *International conference on knowledge discovery and data mining* (pp. 39–48).

Celeux, G., & Govaert, G. (1993). Comparison of the mixture and the classification maximum likelihood in cluster analysis. *Journal of Statistical Computation and Simulation*, *47*, 127–146.

Covões, T.F., Hruschka, E.R., Ghosh, J. (2013). A study of k-means-based algorithms for constrained clustering. *Intelligent Data Analysis*, *17*, 485–505.

Davidson, I., & Ravi, S. (2005). Clustering with constraints: feasibility issues and the k-means algorithm. In *Proceedings of the 2005 SIAM international conference on data mining* (pp. 138–149): SIAM.

Dempster, A.P., Laird, N.M., Rubin, D.B. (1977). Maximum likelihood for incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, *39*, 1–38.

DeSarbo, W.S., & Mahajan, V. (1984). Constrained classification: the use of a priori information in cluster analysis. *Psychometrika*, *49*, 187–215.

Dinler, D., & Tural, M.K. (2016). A survey of constrained clustering. In *Unsupervised learning algorithms* (pp. 207–235): Springer.

Fatehi, K., Bozorgi, A., Zahedi, M.S., Asgarian, E. (2014). Improving semi-supervised constrained k-means clustering method using user feedback. *Journal of Computing and Security*, *1*, 273–261.

Gu, L., & Lu, X. (2012). Semi-supervised subtractive clustering by seeding. In *2012 9th international conference on fuzzy systems and knowledge discovery* (pp. 738–741): IEEE.

Hennig, C., Meila, M., Murtagh, F., Rocci, R. (2015). *Handbook of cluster analysis*. Boca Raton: CRC Press.

Liu, H., & Fu, Y. (2015). Clustering with partition level side information. In *2015 IEEE international conference on data mining* (pp. 877–882): IEEE.

Maitra, R., & Melnykov, V. (2010). Simulating data to study performance of finite mixture modeling and clustering algorithms. *Journal of Computational and Graphical Statistics*, *19*, 354–376.

McLachlan, G., & Peel, D. (2000). *Finite mixture models*. New York: Wiley.

Melnykov, V., Chen, W.-C., Maitra, R. (2012). Mixsim: an R package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software*, *51*, 1–25.

Melnykov, V., Melnykov, I., Michael, S. (2016). Semi-supervised model-based clustering with positive and negative constraints. *Advances in data analysis and classification*, *10*, 327–349.

Nimmo, D.W.R., Herrmann, S.J., Sublette, J.E., Melnykov, I.V., Helland, L.K., Romine, J.A., Carsella, J.S., Herrmann-Hoesing, L.M., Turner, J.A., Vanden Heuvel, B.D. (2018). Occurrence of Chironomid species (Diptera: Chironomidae) in the high Se-78 concentrations and high pH of Fountain Creek Watershed, Colorado, USA. *Western North American Naturalist*, *78*, 39–64–26.

Ruiz, C., Spiliopoulou, M., Menasalvas, E. (2010). Density-based semi-supervised clustering. *Data Mining and Knowledge Discovery*, *21*, 345–370.

Śmieja, M., & Wiercioch, M. (2017). Constrained clustering with a complex cluster structure. *Advances in Data Analysis and Classification*, *11*, 493–518.

Steinley, D., & Brusco, M.J. (2011). Evaluating mixture modeling for clustering: recommendations and cautions. *Psychological Methods*, *16*, 63.

Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S. (2001). Constrained K-means clustering with background knowledge. In *Proceedings of the eighteenth international conference on machine learning (ICML-2001)* (pp. 577–584).

Wang, X., Wang, C., Shen, J. (2011). Semi–supervised K-means clustering by optimizing initial cluster centers. In *International conference on web information systems and mining* (pp. 178–187): Springer.

Yu, Z., Luo, P., You, J., Wong, H.-S., Leung, H., Wu, S., Zhang, J., Han, G. (2015). Incremental semi-supervised clustering ensemble for high dimensional data clustering. *IEEE Transactions on Knowledge and Data Engineering*, *28*, 701–714.

Zhigang, C., Xuan, L., Fan, Y. (2013). Constrained k-means with external information. In *2013 8th International conference on computer science & education* (pp. 490–493): IEEE.