Neural Computing and Applications (2024) 36:5981–6003
https://doi.org/10.1007/s00521-023-09388-x

ORIGINAL ARTICLE

Robust and compact maximum margin clustering for high-dimensional data

Hakan Cevikalp[1] · Edward Chome[2]

Received: 13 March 2023 / Accepted: 13 December 2023 / Published online: 17 January 2024

Abstract
In the field of machine learning, clustering has become an increasingly popular research topic due to its critical importance. Many clustering algorithms have been proposed utilizing a variety of approaches. This study focuses on clustering of high-dimensional data using the maximum margin clustering approach. In this paper, two methods are introduced: The first method employs the classical maximum margin clustering approach, which separates data into two clusters with the greatest margin between them. The second method takes cluster compactness into account and searches for two parallel hyperplanes that best fit to the cluster samples while also being as far apart from each other as possible. Additionally, robust variants of these clustering methods are introduced to handle outliers and noise within the data samples. The stochastic gradient algorithm is used to solve the resulting optimization problems, enabling all proposed clustering methods to scale well with large-scale data. Experimental results demonstrate that the proposed methods are more effective than existing maximum margin clustering methods, particularly in high-dimensional clustering problems, highlighting the efficacy of the proposed methods.

Keywords Maximum margin clustering · Subspace clustering · Hyperplane fitting · Large margin · Robust clustering

1 Introduction

Technology has advanced rapidly, and this has enabled data to be collected in vast amounts with various forms. The data available range from images, videos, text to web documents and many other forms, and most of it is in high dimension in real-world scenarios. The collected data are usually unlabeled due to the prohibitive costs related to the human labor that is required to label data manually [1]. To overcome this limitation, clustering deals with partitioning data into related groups without any prior knowledge on the underlying relationships of the groups (clusters) [2]. More precisely, it is used to discover the unknown hidden groups of data samples and to assign labels to them. Clustering is widely applied in many applications including computer vision, bioinformatics, information retrieval, web analysis, marketing, data analysis and many more. For example, in marketing research, it can be used to identify distinct customer bases for marketing research purposes. In computer vision, it helps in the automatic segmentation of images (in biological or medical imaging it can be used to identify instance tumors or cancerous cells). Clustering is also applied in knowledge discovery and feature extraction. Despite originating as an unsupervised method, clustering has been increasingly used in both semi-supervised and supervised learning applications. Semi-supervised learning involves a combination of unlabeled data with a limited amount of labeled data or side-information provided in the form of similarity/dissimilarity constraints [3, 4]. On the other hand, supervised clustering methods rely solely on labeled data to arrange data for further processing, such as classification [5].

In practical scenarios, high-dimensional data present a more challenging clustering task compared to low-

✉ Hakan Cevikalp
hcevikalp@ogu.edu.tr

Edward Chome
edwardchome@eskisehir.edu.tr

[1] Machine Learning and Computer Vision Laboratory, Electrical and Electronics Engineering Department, Eskisehir Osmangazi University, Meselik 26480, Eskisehir, Turkey

[2] Computer Engineering Department, Eskisehir Technical University, Eskisehir, Turkey

dimensional data. This is mainly due to the dependence of most clustering algorithms on pairwise Euclidean distances among data samples. However, in high-dimensional spaces, Euclidean distances tend to be less reliable, as demonstrated in existing literature [6–8]. The unreliability of Euclidean distances in high-dimensional spaces can be attributed to the sparsity and irregularity of data distributions. These characteristics lead to erratic Euclidean distances between samples, which in turn degrade the clustering performance. The theoretical analysis in [8] further indicates that as the dimensionality increases, the distance to the nearest data point approaches that of the farthest data point, rendering Euclidean distance meaningless in high-dimensional spaces when clustering pairs of samples.

In this paper, we focus on high-dimensional clustering problem and propose new binary clustering methods that maximize the margin between the two clusters. This type of clustering methods is called as maximum margin clustering methods, and it can be considered as the unsupervised version of the well-known large-margin classifiers such as the support vector machines [9].

The rest of the paper is organized as follows: Related methods and our main contributions are given in Sect. 2. We briefly introduce the maximum margin clustering in Sect. 3. This is followed by the introduction of our proposed methods in Sect. 4. We present our experimental results in Sect. 5, and finally our conclusions and future research directions are given in the last section.

2 Related work

The most successful high-dimensional data clustering methods can be roughly divided into two groups: the subspace clustering methods and the maximum margin clustering methods. Motivation behind these clustering methods and related studies is explained below.

2.1 Subspace clustering methods

Recently, subspace clustering has gained significant attention and popularity due to its superior performance in high-dimensional spaces. The objective of subspace clustering is to divide data samples into groups, where each group consists of data samples that lie in the same low-dimensional subspace within the high-dimensional feature space. This problem has received considerable attention, particularly in computer vision, as many commonly used datasets for motion segmentation, hand-written recognition, and face clustering in different illumination conditions can be modeled by a mixture of linear/affine subspaces. Various subspace algorithms have been proposed and can be broadly classified into iterative, statistical, algebraic, and spectral techniques [10, 11]. Iterative approaches, such as k-subspaces [12], k-means Projective Clustering [13], Median k-Flats [14], and recent local affine/convex hull-based methods [6], alternate between assigning points to linear/affine subspaces and updating subspace parameters based on the newly assigned data points to each subspace. RANSAC (RAndom SAmpling Consensus) method [10, 15] tries to fit an hyperplane to a given data. It alternates between randomly selecting some small subset of points from the dataset and computing an hyperplane that best fits to the selected data points.

Methods like Mixtures of Probabilistic Principal Component Analysis (MPPCA) [16] and Multi-Stage Learning (MSL) [17] use statistical approaches that involve approximating each subspace with a Gaussian distribution and updating cluster memberships and Gaussian distribution parameters through the Expectation Maximization (EM) algorithm. Algebraic methods, such as Generalized PCA [18] and its robust variant [19], tackle the subspace clustering problem by formulating it as a high-order polynomial fitting problem. Spectral clustering is the most widely used technique in subspace clustering, and the subspace clustering methods using it differ in how they construct the affinity matrix. For example, some methods use sparse combination coefficients, while others use similarities between local linear subspaces or low-rank representation for constructing the affinity matrix [20–23]. In contrast, a recent greedy selection algorithm creates numerous local best-fit affine subspaces and selects the best ones for the given data [24]. Wang et al. [25] extended the low-rank-based subspace clustering method for the multi-view data clustering problem in which data samples are represented more than one feature set. Passalis and Tefas [26] proposed a discriminative subspace clustering method that is able to provide regularized low-dimensional representations that are optimized toward clustering tasks. In the proposed methodology, the intra-cluster and the inter-cluster distances are transformed into similarities and then manipulated in an appropriate way that ensures that the representation will not collapse or overfit to the supplied labels. For more information on subspace clustering, interested readers are referred to a comprehensive survey of [27].

Compared to maximum margin clustering, subspace clustering is more challenging because the dimensions are not known a priori, unlike hyperplane clustering, where they are known beforehand [10].

2.2 Maximum margin clustering methods

Large or maximum margin techniques have gained extensive usage in supervised learning and have a proven track

record of success. The primary approach in this domain is the support vector machine (SVM), which identifies a linear separating hyperplane in the feature space that maximizes the distance between two class samples. In this context, the margin refers to the Euclidean distance from the closest samples to the separating hyperplane [1].

Maximum margin clustering techniques are an extension of the large margin concept to unsupervised learning, and they are built based on the cluster assumption of [28], which states that the decision boundary should not cross high-density regions, but instead lie in low-density regions. These methods identify the maximum margin hyperplane that separates the data from different clusters. In contrast to supervised maximum margin learning problems, the optimization problem in unsupervised clustering tasks becomes non-convex.

The first maximum margin clustering (MMC) method was proposed by Xu et al. [29]. The authors proposed the clustering problem as a non-convex integer programming problem. After some relaxations, the clustering problem is solved by using semi-definite programming (SDP). However this approach is computationally intensive and only suitable for small datasets including several hundreds of samples. Furthermore, it does not allow to use the bias term, which constraints the clustering boundaries to pass through the origin. Valizdegan and Jin [30] attempted to reduce the computational costs and the missing bias term problem by proposing the generalized maximum margin clustering (GMMC). This method reduced the costs significantly as it reduces the number of parameters in the SDP formulation from n^2 to n, where n is the number of samples. However, this method also requires to solve a SDP problem and therefore it is suitable only for small datasets. Zhang et al. [31] proposed alternating optimization directly to solve the maximum margin clustering. The proposed method iteratively solves a series of support vector regression (SVR) problem that uses the Laplacian loss rather than the common hinge loss. Using the Laplacian loss avoids being stuck in local optimal solutions. A similar method using support vector regression cost for MMC is also used in [32].

More recently, Zhao et al [33] proposed an efficient cutting plane maximum margin clustering algorithm (CPMMC). They use constrained concave–convex procedure to solve each optimization problem after constructing a nested sequence of successively tighter relaxations on the original maximum margin problem. Li et al. [34] proposed to solve the maximum margin clustering problem via Label-Generating (LG-MMC). This method maximizes the margin by generating the most violated label vectors iteratively and then combines them via efficient multiple kernel learning. The authors formulated the problem as a relaxed convex optimization problem avoiding semi-definite programming (SDP) which is very expensive. Their approach scales better than other convex relaxation approaches. Wang et al. [35] proposed the Manifold Regularized Maximum Margin Clustering (MRMMC) method which combines the maximum margin data discrimination and data manifold in a unified clustering objective function. To this end, the authors added another loss term including the graph Laplacian obtained from the adjacency matrix to the existing MMC objective function to ensure that the locally similar samples are assigned to the same clusters. However, this makes the problem more complicated and restricts to apply it to large-scale datasets since one has to create a $n \times n$ Laplacian matrix, where n is the number of samples.

Hu et al. [36] and Zeng et al. [37] proposed a semi-supervised maximum margin clustering method that utilizes similarity and dissimilarity constraints between data samples. Hoai and De la Torre [38] applied the maximum margin concept for temporal clustering in time series. Chen at al. [39] proposed Bayesian max-margin clustering (BMC) which allows maximum margin constraints to be included in a Bayesian clustering model. In [40], a variant of the maximum margin clustering which uses latent representation of data samples is proposed. More recently, Li et al. [41] proposed a new MMC method using bundle method which is called as Bundle Maximum Margin Clustering (BMMC) method. In this method, the non-convex clustering problem is first decomposed into a series of convex sub-problems, and then, the bundle method is utilized to solve each sub-problem. Another MMC method using incremental learning is introduced in [42]. Xue et al. [43] introduced the indefinite kernel MMC method, which approximates the original indefinite kernel by seeking a proxy positive definite kernel and incorporates an F-norm regularizer into the learning problem. The proposed method firstly transforms the clustering problem into a classification one solved by indefinite kernel support vector machine (IKSVM) with an extra class balance constraint, and then the obtained prediction labels are used as the new input class labels at next iteration until the error rate of prediction is smaller than a pre-specified tolerance. Xiaoa et al. [44] applied the maximum margin clustering method to multi-view data learning problem in which data samples are represented more than one feature set. The main idea is to apply complementarity principle by considering one view as the main learning information and the other views as the privileged information, so that multiple views can provide information to complement each other. The resulting clustering method is non-convex optimization problem, and it is solved by applying the constrained concave–convex procedure and cutting plane techniques. Zhang and Zhu [45] proposed optimal margin distribution

clustering method which characterizes the margin distribution by the first- and second-order statistics, i.e., the margin mean and variance. A stochastic mirror descent method is used to solve the resultant minimax problem. A hierarchical margin clustering method that performs clustering recursively in a top–down manner to extend the binary clustering to multiple clusters is introduced in [46]. A deep transductive semi-supervised maximum margin clustering approach utilizing pairwise constraints is proposed in [47]. The proposed method unifies transductive learning, feature learning and maximum margin techniques in the semi-supervised clustering framework. To this end, a deep network structure with restricted Boltzmann machines (RBMs) is learned greedily by using the most violated constraints as in Sequential Minimization Optimization (SMO) algorithm, and the objective function is optimized by using gradient descent. A clustering method which is similar to maximum margin clustering has been proposed in [48]. The main goal is to find low-density hyperplanes for binary clustering. Low-density hyperplanes avoid intersecting high-density regions and typically pass between high-density clusters, which keep the individual clusters intact. The proposed method is built based on a modified stochastic gradient descent applied on a convolution of the empirical distribution function with a smoothing kernel function.

The maximum margin clustering methods are successfully used in different domains including computer vision [31, 49–51], time series analysis [38] and medical applications [52]. For example, [31] used MMC for image segmentation, Farhadi and Tabrizi [49] used MMC for finding different view points of human activities, [51] used MMC to discover geographical clusters of beach images, whereas Hoai and Zisserman [50] used it to improve the performance for visual object classification in computer vision. Similarly, Zhu et al. [52] used MMC and immune evolutionary method for diagnosis of electrocardiogram arrhythmia.

Lastly, we would like to point out that there are close ties with the maximum margin clustering methods and spectral clustering methods. This issue is first explained in [53]. In this study, the authors show that the Normalized Cuts (NCuts) clustering method of Shi and Malik [54] lifts the dataset into an infinite-dimensional feature space and cuts the data by passing a hyperplane through a margin in the projected space. It then labels data points that fall on the same side of the hyperplane as belonging to the same cluster. Then, Valizadegan and Jin [30] showed the formal connection between the maximum margin clustering and the spectral clustering.

2.2.1 Our contributions

In this paper, we propose new methods for the maximum margin clustering. The first proposed method uses the classical maximum margin clustering objective function, and it tries to split the data into two clusters with the largest margin between them. As opposed to the other existing methods, we solve the primal problem by using Stochastic Gradient (SG) algorithm. The second proposed clustering method searches for two compact clusters with the largest margin between them. It should be noted that the classical MMC objective function does not attempt to minimize the intra-cluster variances; thus, there is no guarantee that the returned clusters are compact. In contrast, our proposed method searches for two parallel hyperplanes that best fit to the two clusters and are far from each other as much as possible. Therefore, this clustering method can be seen as a clustering method that unifies the maximum margin clustering and subspace clustering. In the proposed clustering method, the variations of the samples in the same clusters are minimized, whereas the inter-cluster distances are maximized.

In this study, we focus on binary clustering methods. However the proposed methods can always be used for multi-class clustering by hierarchically splitting the clusters until we reach the desired number of the clusters as in NCuts clustering method. We also do not use kernel functions since we are interested in clustering of high-dimensional data samples. Our proposed methods are simpler and faster than the majority of other maximum margin clustering methods, and they scale well with large datasets.

Briefly, our contributions can be summarized as follows:

- We propose an efficient algorithm using stochastic gradient to solve the classical maximum margin clustering problem more efficiently. Moreover, we introduce a novel maximum margin clustering method utilizing robust ramp losses to handle the outliers and noise within the data.
- We propose a completely novel maximum margin clustering method that returns compact clusters by minimizing the intra-cluster variances. This issue is ignored by the existing maximum margin clustering methods in the literature. The proposed method finds two parallel hyperplanes that best fit to two cluster samples with the maximum margin between them. In addition, we introduce the robust version of this clustering method to cope with the outliers and noise within data.
- Our proposed methods significantly outperform other maximum margin and subspace clustering methods in the majority of the tested datasets. Moreover, the proposed clustering methods are more effective in terms

of running speed and they scale well with the large-scale data as demonstrated in the experiments.

3 Maximum margin clustering

MMC follows the maximum margin principle used in supervised SVM (support vector machine) learning. It aims to identify hyperplanes that divide the data into two separate clusters with the largest margin between them out of all potential labelings. SVM, a successful method in supervised learning, has been employed in this capacity. Consider that we are given a dataset in the form $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$ is the example data vector and $y_i \in \{-1, +1\}$ is the corresponding labels. The SVM method finds a hyperplane characterized by $(\mathbf{w}, b)$ which results in a large margin separating the two classes (in a binary case) by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},\mathbf{b},\xi_\mathbf{i}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \end{aligned} \tag{1}$$

where ξ_i's are slack variables for errors, and C is a positive regularization parameter used to tune errors and the separation margin.

Maximum margin clustering is an extension of SVM to the unsupervised learning. Here, our main interest is to find the hyperplanes that partition the data into two different clusters with the largest margin between them over all the possible labelings. Large margin clustering problem can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{w},\mathbf{b},\mathbf{y},\xi_\mathbf{i}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \end{aligned} \tag{2}$$

Note that the maximum margin clustering follows the same formulation as SVM, the only difference is now being that in this case *the labels are unknown*. This is a much difficult optimization problem since it is non-convex. The above optimization problem has a trivially optimal solution that can be obtained by assigning all data samples to the same cluster. In this case, the resulting margin can be infinite. To avoid this problem, we need to put a constraint on the cluster balance. This also alleviates other undesired solutions which separate a single outlier or a very small groups of samples from the remaining data. To this end, Xu et al. [29] introduced the following class balance constraint on the labels $\mathbf{y}$,

$$-l \leq \mathbf{e}^\top \mathbf{y} \leq l \tag{3}$$

where $l \geq 0$ is a constant controlling the class imbalance and $\mathbf{e}$ is the vector whose all entries are set to 1.

Zhao et al. [33] proved that the optimization problem given in (2) can also be formulated as,

$$\begin{aligned} \min_{\mathbf{w},\mathbf{b},\xi_\mathbf{i}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & \left|\mathbf{w}^\top \mathbf{x}_i + b\right| \geq 1 - \xi_i, \quad \xi_i \geq 0, \\ & -l \leq \mathbf{e}^\top \mathbf{y} \leq l, \end{aligned} \tag{4}$$

where the labeling vector $\mathbf{y}$ is calculated as $y_i = \text{sign}(\mathbf{w}^\top \mathbf{x}_i + b)$. It should be noted that this is much easier to solve since the variable $\mathbf{y}$ does not exist in the new formulation. By setting,

$$\xi_i = \max\left\{0, 1 - \left|\mathbf{w}^\top \mathbf{x}_i + b\right|\right\}, \; i = 1, ..., n, \tag{5}$$

the optimization problem becomes,

$$\begin{aligned} \min_{\mathbf{w},\mathbf{b},\xi_\mathbf{i}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} H_1\left(\left|\mathbf{w}^\top \mathbf{x}_i + b\right|\right) \\ \text{s.t.} \quad & -l \leq \mathbf{e}^\top \mathbf{y} \leq l, \end{aligned} \tag{6}$$

where the function $H_1(t) = \max(0, 1 - t)$ is the classical hinge loss. The loss term, $H_1(|\mathbf{w}^\top \mathbf{x}_i + b|) = \max\{0, 1 - |\mathbf{w}^\top \mathbf{x}_i + b|\}$ used in (6), is plotted in Fig. 1, and it is called the symmetric hinge loss. This loss term is widely used in transductive SVMs [1]. Almost all of the maximum margin clustering methods take the dual of the optimization problems given in (2) or (6) and solve the dual problem by SDP or cutting plane algorithms. For example, Xu et al. [29] used the following dual problem,

$$\begin{aligned} \min_{\mathbf{M},\delta,\eta,\mu} \quad & \delta \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{M} \circ \mathbf{K} & \mathbf{e} + \mu - \eta \\ (\mathbf{e} + \mu - \eta)^\top & \delta - 2C\eta^\top \mathbf{e} \end{bmatrix} \geq \mathbf{0} \\ & \eta \geq \mathbf{0}, \mu \geq \mathbf{0}, \mathbf{M} \geq \mathbf{0} \end{aligned} \tag{7}$$

where $\circ$ operation denotes the elementwise product between two matrices, and $\mathbf{K}$ denotes the $n \times n$ kernel matrix formed from the inner products of feature vectors. This requires a solving complicated semi-definite

programming with complexity $O(n^2)$. Furthermore, the maximum margin clustering algorithm formulated here requires clustering boundaries to pass through the origins of data, which is unsuitable for clustering data with unbalanced clusters. Valizadegan and Jin [30] introduced the Generalized Maximum Margin Clustering (GMMC) method which uses the following alternative optimization that reduced the cost from $O(n^2)$ to $O(n)$

$$\begin{aligned}\min_{\delta,\eta,\mathbf{y},\lambda}\ & \frac{1}{2}(\mathbf{e}+\eta-\delta+\lambda\mathbf{y})^{\top}\mathrm{diag}(\mathbf{y})\mathbf{K}^{-1}\mathrm{diag}(\mathbf{y})(\mathbf{e}+\eta \\ & -\delta+\lambda\mathbf{y})+C_{\delta}\sum_{i=1}^{n}\delta_i^2 \\ \text{s.t. } & \eta\geq\mathbf{0},\delta\geq\mathbf{0},\mathbf{y}\in\{+1,-1\}^n.\end{aligned} \tag{8}$$

However, GMMC cannot handle medium datasets with more than one thousand instances as stated in [34].

4 The proposed methods

We have proposed two different maximum margin clustering methods along with their robust versions. The first proposed method uses the classical MMC objective function whose main goal is to split the data into two clusters by using a hyperplane with the maximum margin between clusters. The second one uses a different objective function which targets both the cluster compactness and the maximum margin. To this end, the proposed method searches for two parallel hyperplanes that best fit to the cluster samples with the maximum margin between these two hyperplanes. For robust variations of the proposed methods, we utilize the ramp loss functions that are more robust against to the noise and outliers within data samples. Using ramp losses also allows us to employ more stable concave–convex procedure that solves a convex optimization problem at each iteration. As stated earlier, we focus on binary clustering methods here, and the proposed methods can always be used for multi-class clustering by hierarchically splitting the clusters as in NCuts clustering. The proposed methods are explained in the following subsections.

4.1 Robust maximum margin clustering

4.1.1 Maximum margin clustering by using stochastic gradient (SG) algorithm

Our first proposed method implements the maximum margin clustering method whose objective function is given below,

$$\begin{aligned}\min_{\mathbf{w},\mathbf{b},\xi_i}\ & \frac{1}{2}\|\mathbf{w}\|^2+\frac{C}{n}\sum_{i=1}^{n}H_1\left(\left|\mathbf{w}^{\top}\mathbf{x}_i+b\right|\right) \\ \text{s.t. } & -l\leq\sum_{i=1}^{n}\left(\mathbf{w}^{\top}\mathbf{x}_i+b\right)\leq l.\end{aligned} \tag{9}$$

This optimization problem differs from the one given in (6) in the way the balance constraint, $-l\leq\mathbf{e}^{\top}\mathbf{y}\leq l$ is relaxed and transformed into $-l\leq\sum_{i=1}^{n}\left(\mathbf{w}^{\top}\mathbf{x}_i+b\right)\leq l$. This relaxation is adopted in many MMC methods implementing both SDP and cutting plane methods [33, 35, 37]. As opposed to the other maximum margin methods in the literature, we solve the primal problem by using SG algorithm instead of the dual problem. To enforce the balance constraint, we project the returned hyperplane parameters to the feasible set imposed by the constraint. To this end, we formulate it as the following sub-problem,

$$\mathcal{P}(\mathbf{v})=\underset{\mathbf{v}'\in\mathbb{R}^{d+1}}{\mathrm{argmin}}\ \frac{1}{2}\|\mathbf{v}-\mathbf{v}'\|^2,\quad s.t.\quad -l\leq\mathbf{v}'^{\top}\bar{\boldsymbol{x}}\leq l \tag{10}$$

which has a closed-form solution

$$\mathcal{P}(\mathbf{v})=\begin{cases}\mathbf{v}-\bar{\mathbf{x}}\dfrac{(\mathbf{v}^{\top}\bar{\mathbf{x}}-l)}{\|\bar{\mathbf{x}}\|^2}, & \text{if}\quad \mathbf{v}^{\top}\bar{\mathbf{x}}>l \\ \mathbf{v}-\bar{\mathbf{x}}\dfrac{(\mathbf{v}^{\top}\bar{x}+l)}{\|\bar{\mathbf{x}}\|^2}, & \text{if}\quad \mathbf{v}^{T}\bar{\mathbf{x}}<-l \\ \mathbf{v}, & \text{otherwise.}\end{cases} \tag{11}$$

Here, the vector $\mathbf{v}$ must be set to $\mathbf{v}=[\mathbf{w}\ \ b]$ and $\bar{\mathbf{x}}=\left[\sum_i^n\mathbf{x}_i\ \ n\right]$ in our clustering problem.

In contrast to the other MMC methods, we solve the primal optimization problem by using SG algorithm instead of the dual problem. Therefore, our proposed method is very efficient and it scales well with training set size since the complexity of SG algorithms solving SVM-type problems does not depend on the size of the training set as proved in [55]. More precisely, our algorithm finds an ϵ-accurate solution using only $O(1/(C\epsilon))$ iterations, while each iteration involves a single inner product between $\mathbf{w}$ and $\mathbf{x}$. To put it another way, the total time it takes to obtain a precise solution can be expressed as $O(d/(C\epsilon))$, where d is the dimensionality of $\mathbf{w}$ and $\mathbf{x}$. We call this method as MMC-SG, and it can be summarized as in Algorithm 1.

Algorithm 1 Stochastic Gradient-Based Solver with Projection for MMC clustering (MMC-SG)

Inputs: *data* – $\mathbf{x}_i,\ (i = 1, ..., n)$, *initial hyperplane parameters* – $\mathbf{w}^1, b^1$

Initialize

$\widetilde{\mathbf{w}}_1 = [\mathbf{w}_1\ \ b_1], T > 0, \lambda_0 > 0, \epsilon > 0$

Description:

for $t \in 1, ..., T$ **do**

$\lambda_t \leftarrow \lambda_0/t;$

for $i \in$ randperm(n) **do**

$\widetilde{\mathbf{x}}_i = [\mathbf{x}_i\ \ 1]$

– Compute sub-gradients

$$\mathbf{g}_t = \begin{cases} -C \cdot \text{sign}(\widetilde{\mathbf{w}}_t^\top \widetilde{\mathbf{x}}_i) \cdot \widetilde{\mathbf{x}}_i, & \text{if } 1 - \left|\widetilde{\mathbf{w}}_t^\top \widetilde{\mathbf{x}}_i\right| \geq 0, \\ 0, & \text{otherwise.} \end{cases}$$

– Update hyperplane parameters

$\widetilde{\mathbf{w}}_t \leftarrow \widetilde{\mathbf{w}}_t - \frac{\lambda_t}{n}(\widetilde{\mathbf{w}}_t + \mathbf{g}_t)$

– Project parameters onto the feasible set imposed by the balance constraint

$(\widetilde{\mathbf{w}}_t) = \mathcal{P}(\widetilde{\mathbf{w}}_t)$

end for

if $(t \geq 2)\ \&\ (\|\widetilde{\mathbf{w}}_t - \widetilde{\mathbf{w}}_{t-1}\| < \epsilon)$, **break**

end for

Outputs: *labels* – $y_i,\ (i = 1, ..., n)$, *separating hyperplane* – $\mathbf{w}_t, b_t$.

4.1.2 Robust maximum margin clustering by using stochastic gradient (SG) algorithm

We can use a more robust loss function by interchanging the symmetric hinge loss with the more robust version, the symmetric ramp loss. The symmetric ramp loss function is plotted in Fig. 1b. Using symmetric ramp loss function avoids the effects of data samples which are too close to the separating hyperplane that are harder to cluster. Using ramp loss also allows us to solve the optimization problem by using concave–convex procedure (CCP) [56], which has a theoretical convergence proof. Another advantage of using CCP is its stability since it solves a convex optimization problem iteratively. In this case, the new clustering objective function becomes:

$$\begin{aligned} &\min_{\mathbf{w},\mathbf{b},\xi_\mathbf{i}} \ \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n} SR_s\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \\ &\text{s.t.} \ -l \leq \sum_{i=1}^{n}\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \leq l. \end{aligned} \tag{12}$$

In this equation, the symmetric ramp loss function $SR_s(t)$ is given as,

$$SR_s(t) \ = \ R_s(t) + R_s(-t), \tag{13}$$

where $R_s(t) = \min(1-s, \max(0, 1-t))$ is the "ramp loss." It is shown in Fig. 2. The ramp loss includes setting a parameter, $-1 < s \leq 0$, by the user. This loss function can be represented as the sum of the convex hinge loss and a concave loss function (or as the difference between two convex hinge losses), i.e., $R_s(t) = H_1(t) - H_s(t)$. The ramp loss function is essentially a "clipped" form of the hinge loss, with the location of the clipping determined by the parameter s. In the case of the symmetric ramp loss function, the s parameter dictates the width of the flat section of the symmetric component shown in Fig. 1b.

To train the proposed clustering method with the symmetric ramp loss function defined on unlabeled samples, each unlabeled sample must appear as two examples labeled with both possible classes [1, 57]. We express this more formally as,

$$\begin{aligned} y_i &= 1, \quad i \in [1, \ldots, n]; \\ y_i &= -1, \quad i \in [n+1, \ldots, 2n]; \\ \mathbf{x}_i &= \mathbf{x}_{i-n}, i \in [n+1, \ldots, 2n]. \end{aligned} \tag{14}$$

Then, by using the equations $R_s(t) = H_1(t) - H_s(t)$ and $SR_s(t) = R_s(t) + R_s(-t)$, the above cost function without the constraint can be decomposed into convex and concave parts as,

$$J(\theta) = J_{\text{convex}}(\theta) + J_{\text{concave}}(\theta), \tag{15}$$

where

$$J_{\text{convex}}(\theta) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{2n} H_1\left(y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right)\right), \tag{16}$$

and

$$J_{\text{concave}}(\theta) = -C\sum_{i=1}^{2n} H_s\left(y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right)\right). \tag{17}$$

Due to the decomposability of the cost function presented in (12) into a convex and concave component, the optimization problem can be effectively solved by utilizing the concave–convex procedure (CCCP) as proposed in the literature [56]. By leveraging the CCCP algorithm, the objective of minimizing $J(\theta)$ with regard to the parameter set $\theta = (\mathbf{w}, b)$ can be accomplished through an iterative parameter update scheme governed by the following rule,

$$\theta^{t+1} = \arg\min_{\theta} \left(J_{\text{convex}}(\theta) + J'_{\text{concave}}(\theta^t)\theta\right), \tag{18}$$

subject to the constraint $-l \leq \sum_{i=1}^{n}\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \leq l$. To optimize this loss function, the derivative of the concave part with respect to θ must be found first,

$$\frac{\partial J_{\text{concave}}(\theta)}{\partial \theta} = -C\sum_{i=1}^{2n} \frac{\partial H_s(\theta)}{\partial f_\theta(\mathbf{x}_i)} \frac{\partial f_\theta(\mathbf{x}_i)}{\partial \theta}.$$

To simplify this process, let us define

$$\begin{aligned} \beta_i &= \frac{\partial J_{\text{concave}}(\theta)}{\partial f_\theta(\mathbf{x}_i)} \\ &= \begin{cases} C, & \text{if } y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) < s \text{ and } i = 1 \leq i \leq 2n \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{19}$$

The utilization of the function definition $f_\theta(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ and its derivative $\partial f_\theta(\mathbf{x}_i)/\partial\theta = (\mathbf{x}_i, 1)$ facilitates the application of CCCP updates to the minimization problem, wherein each update necessitates the minimization of the ensuing cost:

$$\begin{aligned} J(\theta) &= J_{\text{convex}}(\theta) + \frac{\partial J_{\text{concave}}(\theta)}{\partial \theta} \\ &= J_{\text{convex}}(\theta) + \left(\sum_{i=1}^{2n} \beta_i \frac{\partial f_\theta(\mathbf{x}_i)}{\partial \theta}\right)\theta \\ &= J_{convex}(\theta) + \sum_{i=1}^{2n} \beta_i y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right). \end{aligned}$$

subject to $-l \leq \sum_{i=1}^{n}\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \leq l$. The inclusion of the hinge losses within the convex cost function leads to the transformation of the entire optimization problem into the following form,

$$\begin{aligned} \arg\min_{\mathbf{w},b} \ & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{2n} H_1\left(y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right)\right) \\ & + \sum_{i=1}^{2n} \beta_i y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \\ \text{s.t.} \ & -l \leq \sum_{i=1}^{n}\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \leq l. \end{aligned} \tag{20}$$

As the cost function mentioned above comprises a summation of convex functions, the resulting function can be classified as convex. This methodology is termed as Robust Maximum Margin Clustering using SG (RMMC-SG) and is presented in Algorithm 2. To solve the central convex minimization problem of the CCCP algorithm defined in Algorithm 2, we employ the SG algorithm provided in Algorithm 3.

Algorithm 2 The Robust Maximum Margin Clustering by Using CCP (RMMC-SG)

Inputs: *data* – $\mathbf{x}_i,\ (i = 1, ..., n)$, *initial hyperplane parameters* – $\mathbf{w}^0, b^0$

Initialize $\theta^0 = (\mathbf{w}^0, b^0)$, $t = 0$, $\epsilon_1 > 0$, $\epsilon_2 > 0$

Compute

$$\beta_i^0 = \frac{\partial J_{concave}(\theta)}{\partial f_\theta(\mathbf{x}_i)} = \begin{cases} C, & \text{if } y_i((\mathbf{w}^0)^\top \mathbf{x}_i + b^0) < s \\ 0, & \text{otherwise.} \end{cases}$$

while $||\mathbf{w}_{t+1} - \mathbf{w}_t|| \geq \epsilon_1$ or $||\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t|| \geq \epsilon_2$ **do**

– Solve the following convex minimization problem by using SG algorithm given in Algorithm 3

$$\arg\min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{2n} H_1(y_i(\mathbf{w}^\top \mathbf{x}_i + b)) + \sum_{i=1}^{2n} \beta_i y_i(\mathbf{w}^\top \mathbf{x}_i + b)$$
$$\text{such that } -l \leq \sum_{i=1}^{n}(\mathbf{w}^\top \mathbf{x}_i + b) \leq l. ;$$

– Set $\mathbf{w}^{t+1} = \mathbf{w},\ \ b^{t+1} = b$;

– Compute

$$\beta_i^{t+1} = \begin{cases} C, & \text{if } y_i((\mathbf{w}^{t+1})^\top \mathbf{x}_i + b^{t+1}) < s \\ 0, & \text{otherwise.} \end{cases}$$

– Set $t = t + 1$;

end while

Outputs: *labels* – $y_i,\ (i = 1, ..., n)$, *separating hyperplane* – $\mathbf{w}, b$.

Algorithm 3 Stochastic Gradient-Based Solver for Robust MMC (RMMC-SG)

Inputs: *data* – $\mathbf{x}_i,\ (i = 1, ..., n)$, *initial hyperplane parameters* – $\mathbf{w}^1, b^1$

Initialize

$\mathbf{w}_1, b_1, T > 0, \lambda_0 > 0, \epsilon > 0$

Description:

for $t \in 1, ..., T$ **do**

$\lambda_t \leftarrow \lambda_0/t;$

for $i \in \text{randperm}(L + 2U)$ **do**

– Compute sub-gradients

$$\mathbf{g}_t = \begin{cases} -y_i C\mathbf{x}_i + \beta_i y_i \mathbf{x}_i, & \text{if } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 1 \\ \beta_i y_i \mathbf{x}_i, & y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 1. \end{cases}$$

$$h_t = \begin{cases} -y_i C + \beta_i y_i, & \text{if } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 1 \\ \beta_i y_i, & y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 1. \end{cases}$$

– Update hyperplane parameters

$\tilde{\mathbf{w}_t} \leftarrow \mathbf{w}_t - \frac{\lambda_t}{L+2U}(\mathbf{w}_t + \mathbf{g}_t)$

$\tilde{b_t} \leftarrow b_t - \frac{\lambda_t}{L+2U} h_t$

– Project parameters onto the feasible set imposed by the balance constraint

$(\mathbf{w}_t, b_t) = \mathcal{P}(\tilde{\mathbf{w}_t}, \tilde{b_t})$

end for

if $(t \geq 2)\ \&\ (\|\mathbf{w}_t - \mathbf{w}_{t-1}\| < \epsilon)$, **break**

end for

Outputs: *separating hyperplane* – $\mathbf{w}_t, b_t$.

4.2 Robust and compact maximum margin clustering

4.2.1 Compact maximum margin clustering (CMMC)

In our second proposed method, we enforce not only the maximum margin between clusters but also the cluster compactness. To this end, we follow the same strategy as in large margin classifiers using affine hulls [58] and search for two parallel hyperplanes that best fit to the cluster samples but at the same time as far as possible from each other (from this point of view, the proposed clustering method can be regarded as the unsupervised version of the large margin classifier using affine hulls). Therefore, the proposed clustering method can be seen as a hybrid method that combines the maximum margin clustering and subspace clustering methods. The difference between the subspace clustering methods and our proposed method is that we are fitting the cluster samples to parallel hyperplanes that can be regarded as $d - 1$ dimensional affine spaces, whereas general subspace clustering methods do not require parallel subspaces and the dimensions of the subspaces are not fixed. This idea is visualized in Fig. 3,

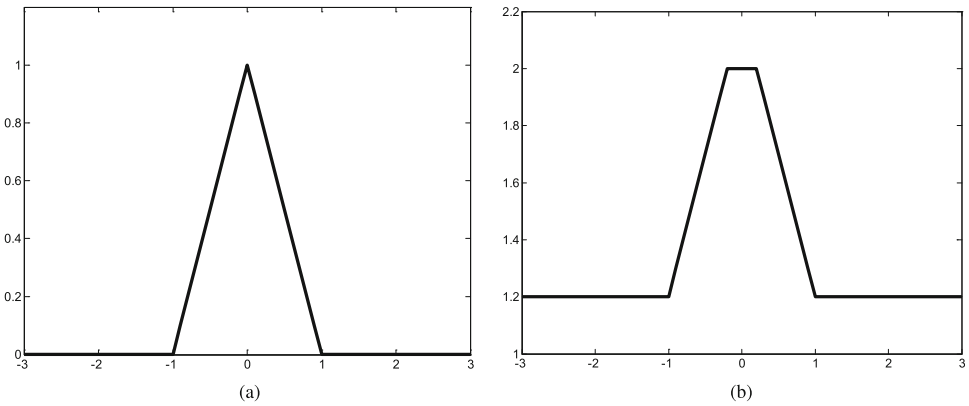

Fig. 1 Loss functions used for the maximum margin clustering. **a** The symmetric hinge loss, $H_1(|t|) = \max(0, 1 - |t|)$, **b** The symmetric ramp loss, $SR_s(t) = R_s(t) + R_s(-t)$. Here, we set $s = -0.20$

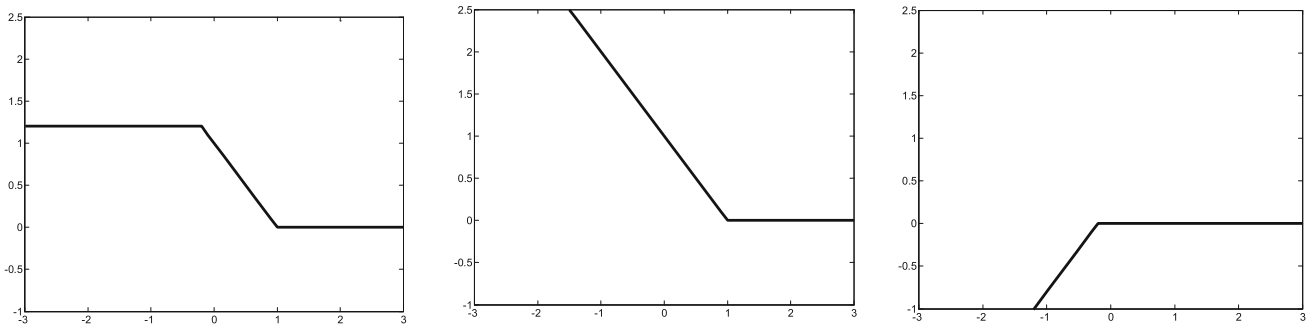

Fig. 2 The illustration of the ramp loss function, $R_s(t) = H_1(t) - H_s(t)$, where $H_a(t) = \max(0, a - t)$ is the classical hinge loss. Here, we set $s = -0.20$

where two clusters lying on two parallel hyperplanes are separated with the maximum margin. In this case, each cluster can be approximated with any fitting affine subspace that lies inside the supporting hyperplanes.

The clustering problem that searches for two parallel hyperplanes that best fit to the cluster samples and separated with a large margin can be formulated as,

$$\begin{aligned} \min_{\mathbf{w},\mathbf{b},\boldsymbol{\xi}_\mathbf{i}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n}\left|\left|\mathbf{w}^\top \mathbf{x}_i + b\right| - 1\right| \\ \text{s.t.} \quad & -l \leq \sum_{i=1}^{n}\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \leq l. \end{aligned} \tag{21}$$

The second loss term is visualized in Fig. 4, and it is similar to the Laplacian loss used in support vector regression. It should be noted that the Laplacian loss is also used for the maximum margin clustering in [31]. However, they start with assigned labels and they iteratively solve the convex supervised learning problem using Laplacian loss in each iteration. Based on the returned hyperplane, they update the labels and continue with solving new supervised learning problem. In contrast, our problem is non-convex, we do not have label information, and we solve only one optimization problem. It should be noted the second term penalizes all the samples that do not lie exactly on the supporting hyperplanes, which is difficult to hold in practice. To avoid this problem, we can define a positive constant, $0 \leq \xi < 1$ (similar to the slack variables in support vector regression problem), that allows not to punish the samples lying in the intervals between $\mathbf{w}^\top \mathbf{x}_i + b = 1 + \xi$ and $\mathbf{w}^\top \mathbf{x}_i + b = 1 - \xi$ for positive cluster samples and between $\mathbf{w}^\top \mathbf{x}_i + b = -1 + \xi$ and $\mathbf{w}^\top \mathbf{x}_i + b = -1 - \xi$ for the negative samples as illustrated in Fig. 5. We implemented our method with this user-defined ξ parameter. The resulting method is called the Compact Maximum Margin Clustering by using SG (CMMC-SG), and it is given in Algorithm 4.

Algorithm 4 Stochastic Gradient-Based Solver with Projection for Compact MMC clustering (CMMC-SG)

Inputs: *data* – $\mathbf{x}_i,\ (i = 1, ..., n)$, *initial hyperplane parameters* – $\mathbf{w}^1, b^1$

Initialize

$\widetilde{\mathbf{w}}_1 = [\mathbf{w}_1\ \ b_1], \xi, T > 0, \lambda_0 > 0, \epsilon > 0$

Description:

for $t \in 1, ..., T$ **do**

$\lambda_t \leftarrow \lambda_0/t$;

for $i \in$ randperm(n) **do**

$\widetilde{\mathbf{x}}_i = [\mathbf{x}_i\ \ 1]$

– Compute sub-gradients

$$\mathbf{g}_t = \begin{cases} C\widetilde{\mathbf{x}}_i, & \text{if } (\widetilde{\mathbf{w}}_t^\top \widetilde{\mathbf{x}}_i) > 1 + \xi, \\ -C\widetilde{\mathbf{x}}_i, & \text{if } 0 \le (\widetilde{\mathbf{w}}_t^\top \widetilde{\mathbf{x}}_i) < 1 - \xi, \\ C\widetilde{\mathbf{x}}_i, & \text{if } -1 + \xi < (\widetilde{\mathbf{w}}_t^\top \widetilde{\mathbf{x}}_i) < 0, \\ -C\widetilde{\mathbf{x}}_i, & \text{if } (\widetilde{\mathbf{w}}_t^\top \widetilde{\mathbf{x}}_i) < -1 - \xi, \\ 0, & \text{otherwise.} \end{cases}$$

– Update hyperplane parameters

$\widetilde{\mathbf{w}}_t \leftarrow \widetilde{\mathbf{w}}_t - \frac{\lambda_t}{n}(\widetilde{\mathbf{w}}_t + \mathbf{g}_t)$

– Project parameters onto the feasible set imposed by the balance constraint

$(\widetilde{\mathbf{w}}_t) = \mathcal{P}(\widetilde{\mathbf{w}}_t)$

end for

if $(t \ge 2)\ \&\ (\|\widetilde{\mathbf{w}}_t - \widetilde{\mathbf{w}}_{t-1}\| < \epsilon)$, **break**

end for

Outputs: *labels* – $y_i,\ (i = 1, ..., n)$, *separating hyperplane* – $\mathbf{w}, b$.

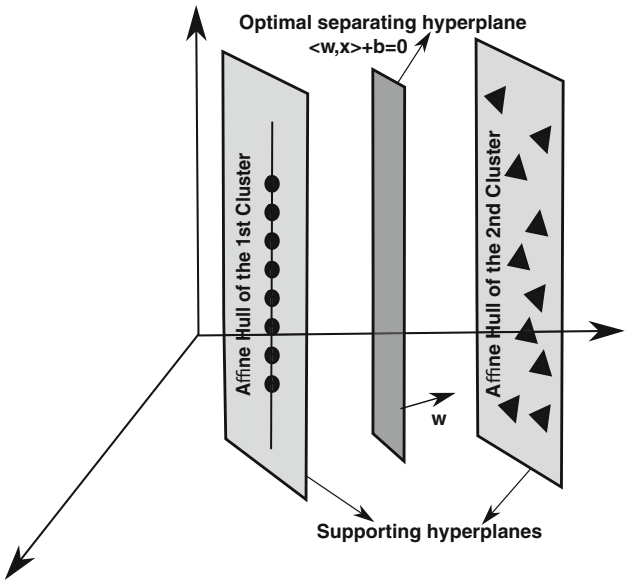

Fig. 3 The proposed method searches for two parallel hyperplanes (supporting hyperplanes) that best fit to the cluster samples and separated with a large margin. This corresponds to the approximating cluster samples with affine hulls (affine subspaces) that lie on the supporting hyperplanes

4.2.2 Robust compact maximum margin clustering (RCMMC)

One of the problems with the loss function given in Eq. (21) is that it is affected by the samples that are far from the supporting hyperplanes that are used for approximating clusters. There may be outliers or noisy data that may be far from the fitting hyperplanes, and the proposed CMMC-SG method may not return good clusters in such cases. To avoid this problem and to make the method more robust to outliers and noisy samples, we can suppress the costs coming from the samples that lie very far from the fitting hyperplanes. To this end, we use the following robust clustering cost in the new clustering method,

$$\begin{aligned} &\min_{\mathbf{w},\mathbf{b},\xi_\mathbf{i}} \ \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n}\big(G_1\big(\mathbf{w}^\top\mathbf{x}_i+b\big)+G_{-1}\big(\mathbf{w}^\top\mathbf{x}_i+b\big)\big), \\ &\text{s.t.} \ -l \le \sum_{i=1}^{n}\big(\mathbf{w}^\top\mathbf{x}_i+b\big) \le l. \end{aligned} \tag{22}$$

where $G_1(x)$ and $G_{-1}(x)$ are mixture of hinge losses and they are defined as,

$$\begin{aligned} G_1(x) &= \max(-1,t-x)-\max(-1,s-x) \\ &\quad +\max(1,t+x)-\max(1,s+x), \\ G_{-1}(x) &= \max(1,t-x)-\max(1,s-x) \\ &\quad +\max(-1,t+x)-\max(-1,s+x). \end{aligned} \tag{23}$$

Here, t is a user-defined constant that can take values between 0 and 0.5, and $s = t + 0.8$. The parameter, t, is similar to ξ term, and it controls the flat region around -1 and 1. These loss terms are plotted in Fig. 6. The resulting method can be solved directly or by using CCP procedure as before since the loss terms can be decomposed into convex and concave parts. The algorithm solving the optimization problem directly is summarized in Algorithm 5.

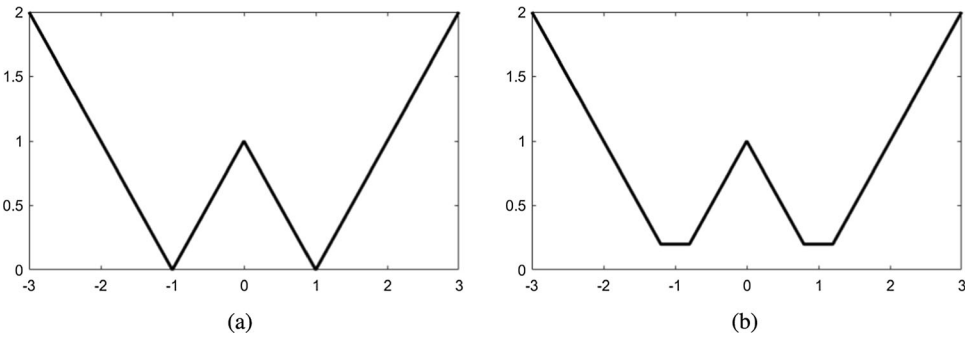

Fig. 4 The visualization of the loss terms: **a** the loss term $\sum_{i=1}^{n}||\mathbf{w}^\top\mathbf{x}_i+b|-1|$, **b** the loss term which also uses ξ term (here it is set to $\xi = 0.20$

Algorithm 5 Stochastic Gradient-Based Solver with Projection for Robust Compact Maximum Margin Clustering (RCMMC)

Inputs: *data* – $\mathbf{x}_i,\ (i = 1, ..., n)$, *initial hyperplane parameters* – $\mathbf{w}^1, b^1$

Initialize

$\widetilde{\mathbf{w}}_1 = [\mathbf{w}_1\ \ b_1], \xi, T > 0, \lambda_0 > 0, \epsilon > 0$

Description:

for $p \in 1, ..., T$ **do**

$\lambda_p \leftarrow \lambda_0/p$;

for $i \in$ randperm(n) **do**

$\widetilde{\mathbf{x}}_i = [\mathbf{x}_i\ \ 1]$

– Compute sub-gradients

$\mathbf{g}_p = zeros(d + 1, 1)$

$\mathbf{g}_p = \mathbf{g}_p - C\widetilde{\mathbf{x}}_i, \quad \text{if } (t - (\widetilde{\mathbf{w}}_p^\top \widetilde{\mathbf{x}}_i)) > -1,$

$\mathbf{g}_p = \mathbf{g}_p + C\widetilde{\mathbf{x}}_i, \quad \text{if } (s - (\widetilde{\mathbf{w}}_p^\top \widetilde{\mathbf{x}}_i)) > -1,$

$\mathbf{g}_p = \mathbf{g}_p + C\widetilde{\mathbf{x}}_i, \quad \text{if } (t + (\widetilde{\mathbf{w}}_p^\top \widetilde{\mathbf{x}}_i)) > 1,$

$\mathbf{g}_p = \mathbf{g}_p - C\widetilde{\mathbf{x}}_i, \quad \text{if } (s + (\widetilde{\mathbf{w}}_p^\top \widetilde{\mathbf{x}}_i)) > 1,$

$\mathbf{g}_p = \mathbf{g}_p - C\widetilde{\mathbf{x}}_i, \quad \text{if } (t - (\widetilde{\mathbf{w}}_p^\top \widetilde{\mathbf{x}}_i)) > 1,$

$\mathbf{g}_p = \mathbf{g}_p + C\widetilde{\mathbf{x}}_i, \quad \text{if } (s - (\widetilde{\mathbf{w}}_p^\top \widetilde{\mathbf{x}}_i)) > 1,$

$\mathbf{g}_p = \mathbf{g}_p + C\widetilde{\mathbf{x}}_i, \quad \text{if } (t + (\widetilde{\mathbf{w}}_p^\top \widetilde{\mathbf{x}}_i)) > -1,$

$\mathbf{g}_p = \mathbf{g}_p - C\widetilde{\mathbf{x}}_i, \quad \text{if } (s + (\widetilde{\mathbf{w}}_p^\top \widetilde{\mathbf{x}}_i)) > -1,$

– Update hyperplane parameters

$\widetilde{\mathbf{w}}_p \leftarrow \widetilde{\mathbf{w}}_p - \frac{\lambda_p}{n}(\widetilde{\mathbf{w}}_p + \mathbf{g}_p)$

– Project parameters onto the feasible set imposed by the balance constraint

$(\widetilde{\mathbf{w}}_p) = \mathcal{P}(\widetilde{\mathbf{w}}_p)$

end for

if $(p \geq 2)\ \ \&\ \ (\|\widetilde{\mathbf{w}}_p - \widetilde{\mathbf{w}}_{p-1}\| < \epsilon)$, **break**

end for

Outputs: *labels* – $y_i,\ (i = 1, ..., n)$, *separating hyperplane* – $\mathbf{w}_t, b_t$.

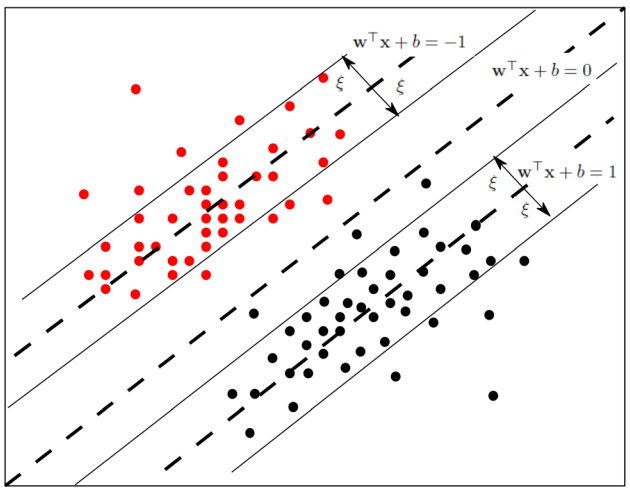

Fig. 5 The visualization of the compact maximum margin clustering that does not punish the samples closer to the best fitting hyperplanes

5 Experiments

In this section we present the test results of our proposed clustering methods, MMC-SG, RMMC-SG, CMMC-SG and RCMMC-SG. We compare our algorithms against each other and also with the existing maximum margin clustering and subspace clustering methods including CPMMC [33], IterSVR [31], LG-MMC [34], Projective k-means [13], SMC [21], SSC [11], LRR [23] and OSC [59].

Our methods have several parameters that must be set by the user, and we would like to point out that initialization of the proposed clustering methods is crucial for good accuracies. Therefore, we initialized the proposed clustering methods by using either CPMMC or k-means clustering outputs. The user must set C parameter and balance ratio, l. We used a small part of the datasets to determine the best parameter values, and then we run the proposed clustering methods on the entire dataset. Typically, setting the balance parameter very small values $l < 0.1$ produced better accuracies. Also, setting the step size of the SG method is important and we decreased the initial value at each iteration by dividing the initial value with the iteration number.

5.1 Experiments on low-dimensional datasets

We used the following low-dimensional datasets that are commonly used for comparison of the maximum margin clustering methods: Wine [60], UCI-Digits [61], Letter [62], Satellite [63], Usps [64] and Ionosphere [65]. We present the detailed description of these datasets in Table 1, where n is the number of data samples, d is the dimension, and c is the number of classes. In these clustering problems, the number of samples, n, is much higher than the dimensionality of the feature space, d. In each dataset, we take only the first 2 classes, with the exception of when we explicitly indicate which classes we take, for example UCI-digit dataset where the two combinations of classes are indicated explicitly. For the UCI-digits dataset, we chose the (1 &7, 2 &7, 3 &8, and 8 &9) pair combinations which are the most difficult to differentiate as noted by [33].

In our assessment of clustering accuracy, we follow the same setting as in [29], where we do not include the labels in all samples when we run our clustering algorithms. Then, we compare the true class labels and the cluster memberships returned by the tested clustering methods. Finally, to measure the clustering accuracy, we used the classification accuracy which is adopted by other maximum margin clustering and subspace clustering methods. The results are averaged over 10 independent runs for all the clustering algorithms. We initialized the proposed clustering methods by using CPMMC. For the proposed MMC-SG and RMMC-SG methods, we used the separating hyperplane returned by CPMMC for initialization. However, for CMMC-SG and RCMMC-SG methods, we applied supervised affine hull margin classifier to find the supporting hyperplanes that best fit to the clusters returned by CPMMC. Then, we initialized our compact maximum margin clustering methods with the resulting hyperplane.

The accuracies of the tested clustering methods are given in Table 2. Among the maximum margin-clustering methods, IterSVR [31] achieves the best accuracy for two datasets, CPMMC [33] obtains the best accuracy for one dataset, and our proposed clustering method, RMMC-SG,

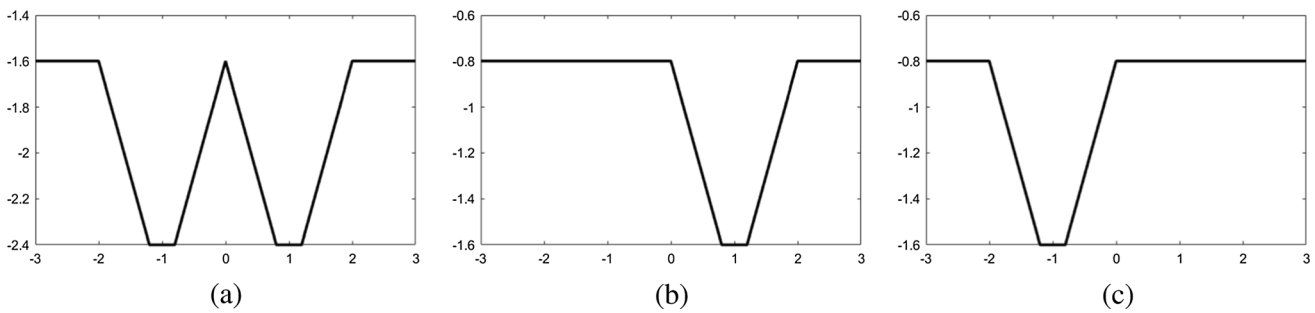

Fig. 6 The visualization of the loss terms: **a** the plot of $G_1(x) + G_{-1}(x)$, **b** the plot for $G_1(x)$, **c** the plot for $G_{-1}(x)$. Here, t is set to 0.2

Table 1 Information on the datasets used for comparison of the proposed binary clustering methods

Dataset	Instance (n)	Dimension (d)	Classes (c)
Wine	130	13	2
UCI-Digits 1 &7	1137	64	2
UCI-Digits 2 &7	1123	64	2
UCI-Digits 3 &8	1126	64	2
UCI-Digits 8 &9	1116	64	2
Letter	1555	16	2
Satellite	4435	36	2
Ionosphere	351	34	2
Usps	7291	256	2

yields the best accuracy for one dataset. It should be noted that although CPMMC method obtains the best accuracy for one dataset, it significantly fails for Letter A &B, Satellite and Wine datasets. (The authors report better accuracies on some of the tested datasets. But, we never obtained these results although we tried many hyper-parameters. The authors of [34] also report the same accuracies that we reported in Table 2). SMC method is the best performing method among subspace clustering methods, and this method also seems the best performing method among all tested clustering methods. Regarding our proposed clustering methods, MMC-SG, RMMC-SG and RCMMC-SG perform well and yield comparable results to the best performing clustering methods. But, CMMC-SG significantly fails for the most of the tested datasets. This shows that exact fitting of the data samples to supporting hyperplanes does not work well for low-dimensional datasets. This is expected since the data samples span the whole feature space, and it is impossible to fit the data to the lower-dimensional parallel hyperplanes. This is also clear from the successful results of RCMMC-SG, which is the robust counterpart of CMMC-SG. It does not punish the samples very far from the supporting hyperplanes and obtains the best accuracies among the proposed clustering methods in general.

5.2 Experiments on high-dimensional datasets

We also compared our proposed clustering methods on high-dimensional datasets: Coil-100 [66], FaceScrub [67], Cifar 10, Gender [1], 20 Newsgroups [68] and Duke Breast Cancer MRI (DBC-MRI) [69] datasets.

The Coil-100 database [66] includes 100 different objects and 72 views of each object taken at 5-degree-apart orientations. All images are converted to gray-scale images, and their original dimensionality 128×128 was reduced to $d = 64 \times 64 = 4096$ by down-sampling. We applied k-means clustering to determine the most difficult classes for clustering and chose 5 difficult binary-class pairs.

The FaceScrub dataset [67] includes face images of 530 celebrities. It has been created by detecting faces based on automated search of public figures on the internet followed by manually checking and cleaning the results. We extracted 4096-dimensional CNN features of face images by using a network pre-trained on a large-scale face

Table 2 Clustering accuracies (%) of tested methods on low-dimensional datasets

Methods	UCI Dig. 1 &7	UCI Dig. 2 &7	UCI Dig. 3 &8	UCI Dig. 8 &9	Ionosphere	Letter A &B	Satellite 1 &2	Usps	Wine
MMC-SG	98.6 ± 3.4	99.8 ± 0.3	96.9 ± 0.6	94.5 ± 1.6	70.1 ± 0.0	93.8 ± 0.03	97.2 ± 1.2	99.0 ± 0.04	94.9 ± 0.4
RMMC-SG	99.5 ± 0.1	99.9 ± 0.2	$\mathbf{97.2 \pm 0.0}$	95.1 ± 0.3	66.0 ± 0.1	93.8 ± 0.03	98.1 ± 0.02	96.1 ± 2.2	95.0 ± 0.5
CMMC-SG	60.0 ± 6.0	75.1 ± 15.0	60.6 ± 3.4	60.1 ± 6.2	71.5 ± 0.0	92.3 ± 0.06	68.4 ± 7.0	99.7 ± 0.0	93.9 ± 0.0
RCMMC-SG	99.6 ± 0.2	99.2 ± 2.1	96.8 ± 0.7	96.0 ± 0.8	71.5 ± 0.0	94.0 ± 0.0	98.3 ± 0.02	99.8 ± 0.0	95.0 ± 0.4
CPMMC	94.2 ± 0.0	$\mathbf{100 \pm 0.0}$	96.9 ± 0.0	94.9 ± 0.0	64.1 ± 0.0	70.0 ± 0.0	70.0 ± 0.0	98.2 ± 0.0	60.0 ± 0.0
IterSVR	99.7 ± 0.0	$\mathbf{100 \pm 0.0}$	96.7 ± 0.1	$\mathbf{96.6 \pm 0.0}$	69.1 ± 0.2	91.6 ± 0.0	83.4 ± 0.0	97.7 ± 0.02	93.9 ± 0.0
LG-MMC	96.7 ± 0.0	88.4 ± 0.0	78.2 ± 0.0	96.3 ± 0.0	62.4 ± 0.0	92.7 ± 0.0	97.0 ± 0.0	75.9 ± 0.0	93.1 ± 0.0
P-k-means	60.4 ± 0.0	98.8 ± 3.7	96.7 ± 1.6	89.0 ± 0.0	$\mathbf{74.7 \pm 0.0}$	88.1 ± 0.0	85.2 ± 0.3	$\mathbf{100 \pm 0.0}$	83.9 ± 0.2
SMC	$\mathbf{100 \pm 0.0}$	99.7 ± 0.0	96.6 ± 0.0	90.7 ± 0.0	70.4 ± 0.0	$\mathbf{94.3 \pm 0.0}$	99.3 ± 0.0	$\mathbf{100 \pm 0.0}$	96.9 ± 0.0
SSC	96.1 ± 0.0	98.3 ± 0.0	89.6 ± 0.0	71.7 ± 6.4	65.0 ± 0.0	51.2 ± 0.0	96.5 ± 0.05	97.4 ± 0.0	89.2 ± 0.0
LRR	95.5 ± 0.6	95.2 ± 0.0	91.0 ± 0.0	89.3 ± 0.0	70.1 ± 0.1	91.0 ± 0.0	$\mathbf{99.4 \pm 0.0}$	98.9 ± 0.0	96.2 ± 0.0
OSC	70.6 ± 0.0	99.2 ± 0.0	95.2 ± 0.0	88.0 ± 0.2	65.1 ± 8.0	92.2 ± 0.0	97.1 ± 0.0	99.9 ± 0.0	$\mathbf{97.7 \pm 0.0}$

The bold fonts represent the best accuracies

Fig. 7 Samples of male (left) and female (right) images from "in-the-wild" dataset used for gender estimation experiments

dataset. For a fair evaluation, we did not apply any fine-tuning to the pre-trained deep neural network since it uses label information. We again determined 5 difficult binary-class pairs based on k-means clustering. The number of samples belonging to the selected classes changes between 203 and 283.

The Cifar 10 dataset consists of 60K, 32×32 color images of 10 classes, with 6K images per class. There are 50K training and 10K test samples. We extracted 4096-dimensional CNN features of images by using a network pre-trained on ILSVRC 2015 dataset. We did not apply any fine-tuning as before.

For Gender dataset, we adopted the same dataset we used in [1]. This dataset was created by using three publicly available datasets, namely, Labelled Faces in the Wild [70], PubFig [71], and PAL [72]. From the Internet, we also downloaded a total of 14,000 face images. Multiple independent individuals annotated these images. We then created a subset of approximately 34,000 near-frontal images, characterized by a yaw angle within the range of $-30°$ to 30°. This dataset presents a considerable challenge as it comprises "in-the-wild" images that exhibit significant variations in illumination, race, resolution and background clutter. In Fig. 7, a random selection of face images from this dataset is displayed. We initially used a commercial face detector to locate and identify the images, followed by the utilization of a landmark detector [73]. These detected landmarks were instrumental in transforming the faces into a standardized pose of dimensions 60×40 pixels. We further trained a 2048-dimensional feature descriptor using a Convolutional Neural Network (CNN) with 7 convolutional and 2 fully connected layers. As opposed to the our

proposed clustering methods, most of the tested clustering methods had memory or convergence problems when we used full 34K data; therefore, we randomly selected 1000 samples from each gender 5 times and used these samples for clustering tests.

The 20 Newsgroups dataset contains about 20,000 newsgroup documents. They are distributed across almost equitably among 20 different newsgroups categories, each matching a different topic. Some of the newsgroups are very closely related to each other for instance comp.sys.ibm.pc.hardware and comp.sys.mac.hardware, while others are strongly unrelated such as misc.forsale and soc.religion.christian for instance. The data are represented with 61,188 high-dimensional bag of words features. We determined 5 difficult binary-class pairs based on k-means clustering as before.

Duke Breast Cancer MRI dataset has 922 patients collected in Duke Hospital from January 1, 2000, to March 23, 2014, with invasive breast cancer and available pre-operative MRI. In total, 529 dimensional features are extracted from MRI data by using a wide range of imaging characteristics including size, shape, texture, and enhancement of both the tumor and the surrounding tissue.

It should be noted that the dimension of the sample space, d, is much larger than the number of samples in each class in all tested datasets. CPMMC significantly failed on these datasets; thus, we initialized the proposed clustering methods by using the clusters returned by the k-means clustering (we also omitted the results of CPMMC since they are too low). To this end, we applied the classical SVM and affine hull margin classifiers to find the hyperplanes separating the clusters returned by k-means clustering method. Then, the separating hyperplane returned by SVM is used to initialize the proposed MMC-SG and RMMC-SG clustering method, and the separating hyperplane returned by the affine hull margin classifier is used to initialize CMMC-SG and RCMMC-SG clustering methods.

The results are given in Table 3. Missing results in the table indicate that the corresponding clustering methods had a memory or convergence problem due to the size of the dataset. As opposed to the low-dimensional clustering results, our proposed method, RCMMC-SG, typically yields the best clustering accuracies for high-dimensional datasets achieving the best performance on five datasets among all tested six datasets. OSC achieves the best clustering accuracy on the FaceScrub dataset followed by our proposed clustering method CMMC-SG. The proposed CMMC-SG, which significantly failed for low-dimensional clustering problems, also performs well due to the high dimensionality of the feature spaces. Among the proposed methods, the clustering methods returning compact clusters perform better compared to the ones that aim maximum margin clustering only. This clearly shows that fitting of the data samples to supporting hyperplanes significantly improves accuracies for high-dimensional clustering problems. This is also illustrated in Fig. 8. We plotted the decision scores, $\left(\mathbf{w}^\top \mathbf{x}_i + b\right)$, obtained by the proposed clustering methods in this figure. The samples are sorted based on true class memberships (e.g., the first 72 scores belong to the first class samples and the remaining 72 scores belong to the second class samples in the first row). As seen in the figure, the proposed clustering methods typically return correct labels with a few exceptions. Also, the scores of the samples belonging to Coil dataset returned by the compact maximum margin clustering methods, CMMC-SG and RCMMC-SG, are compactly clustered around $+1$ and -1, which shows that the majority of the samples exactly lie on the parallel supporting hyperplanes. Our proposed CMMC-SG and RCMMC-SG clustering

Table 3 Clustering accuracies (%) of tested clustering methods on high-dimensional datasets

Methods	Coil	FaceScrub	Cifar 10	Gender	20 Newsgroups	DBC-MRI
MMC-SG	94.3 ± 4.8	97.5 ± 3.3	80.0 ± 7.3	$\mathbf{97.8 \pm 0.0}$	84.7 ± 16.5	67.3 ± 5.0
RMMC-SG	93.3 ± 4.5	98.3 ± 1.8	79.4 ± 9.0	$\mathbf{97.8 \pm 0.0}$	85.9 ± 13.1	62.7 ± 4.1
CMMC-SG	90.7 ± 0.5	99.4 ± 0.4	83.5 ± 3.7	$\mathbf{97.8 \pm 0.1}$	84.9 ± 16.8	75.9 ± 7.6
RCMMC-SG	$\mathbf{95.6 \pm 1.8}$	98.6 ± 0.9	$\mathbf{87.1 \pm 6.4}$	$\mathbf{97.8 \pm 0.0}$	$\mathbf{86.4 \pm 13.7}$	$\mathbf{79.6 \pm 1.1}$
IterSVR	88.5 ± 6.4	96.5 ± 5.4	82.8 ± 7.9	$\mathbf{97.8 \pm 0.0}$	85.0 ± 18.2	59.6 ± 8.3
LG-MMC	91.9 ± 6.5	91.0 ± 4.5	65.9 ± 3.6	94.3 ± 0.0	84.8 ± 8.8	$\mathbf{79.6 \pm 0.3}$
P-k -means	76.9 ± 16.2	60.2 ± 3.5	62.3 ± 8.9	80.2 ± 0.2	68.1 ± 6.6	59.1 ± 7.2
SMC	90.6 ± 5.6	95.5 ± 1.7	51.2 ± 0.8	84.2 ± 5.9	64.5 ± 0.6	$\mathbf{79.6 \pm 1.2}$
SSC	93.9 ± 6.5	98.9 ± 2.2	83.1 ± 4.4	91.7 ± 0.0	–	75.0 ± 1.7
LRR	87.9 ± 13.0	98.2 ± 1.2	83.3 ± 4.2	50.1 ± 0.0	–	–
OSC	93.9 ± 5.9	$\mathbf{99.8 \pm 0.2}$	86.7 ± 4.0	97.4 ± 0.0	–	59.1 ± 2.7
k-means	80.6 ± 8.0	87.5 ± 7.2	77.5 ± 10.0	$\mathbf{97.8 \pm 0.0}$	84.0 ± 16.2	57.7 ± 4.7

The bold fonts represent the best accuracies

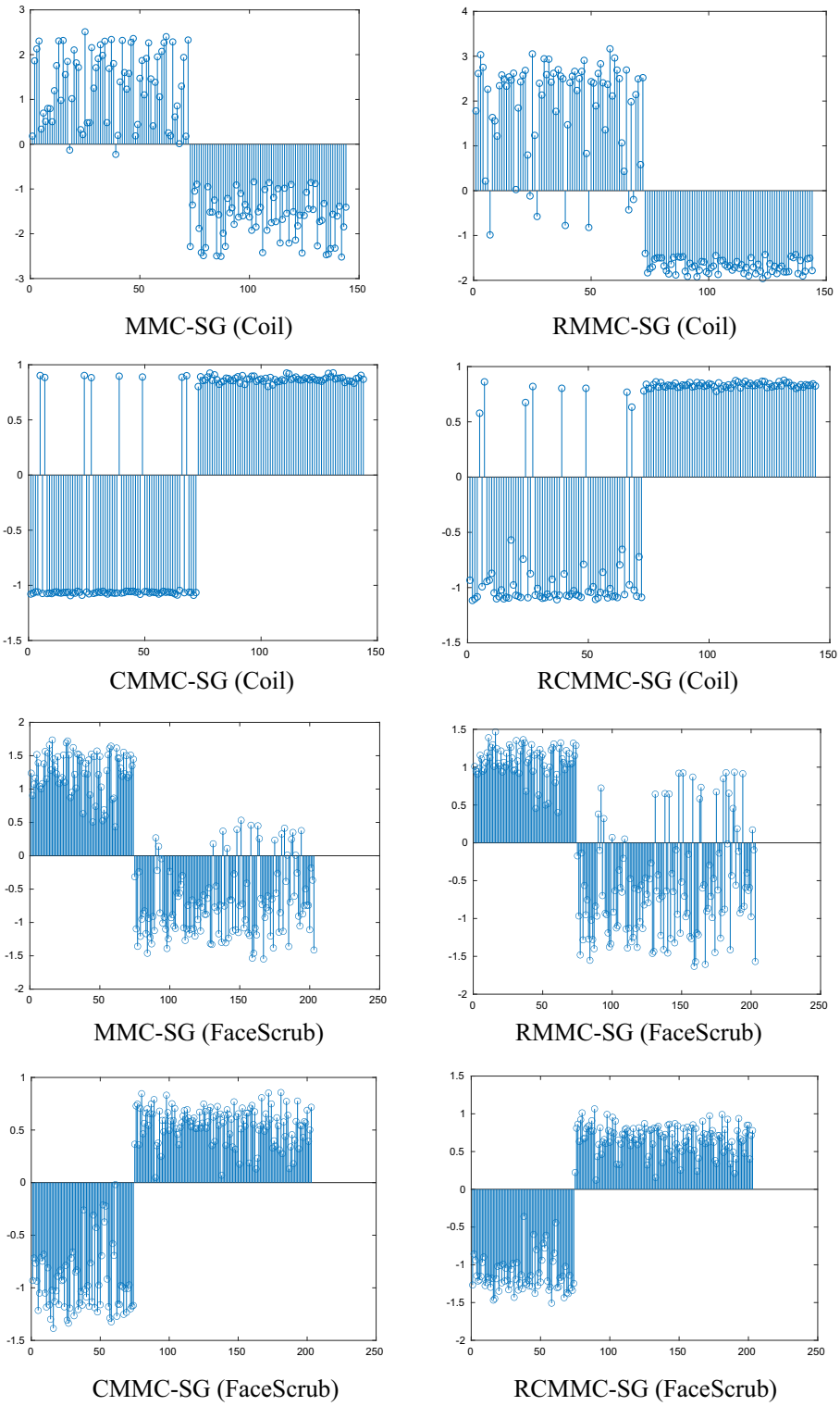

Fig. 8 The visualization of the scores of the proposed methods tested on the Coil and FaceScrub datasets. The first two rows show the scores obtained on the Coil dataset, and the last two rows show the scores obtained on the FaceScrub dataset. When the number of samples is very small compared to the dimensionality as in Coil dataset, the cluster samples fit to the supporting hyperplanes better

methods significantly outperform other maximum margin clustering methods. Among subspace clustering methods, OSC is the best performing one, and it also achieves the best accuracy on the FaceScrub dataset. CPMMC failed to converge on all tested datasets and yielded very low accuracies. Therefore, we omitted its results in the table.

5.3 Comparison of testing times

One of the main problems associated with traditional maximum margin approaches is that they are computationally expensive and only suitable for a dataset with a few hundred samples. In contrast, our proposed clustering methods are very efficient and they scale well with training set size since we utilize Stochastic Gradient (SA) algorithm. As indicated in [55], the complexity of SG algorithms solving SVM-type problems does not depend on the size of the training set and they are fast since we have to make simple dot products between samples, $\mathbf{x}_i$ and hyperplane normal, $\mathbf{w}$. To verify these facts, we conducted experiments to compare testing times of all the clustering methods used in this study.

We compared the testing times of the clustering methods on 5 datasets: UCI Digits 1 &7, Ionosphere, Coil, FaceScrub and Gender 34K datasets. Please note that we used all 34K samples in the Gender dataset rather than 2K samples used in high-dimensional clustering experiments given above. The tests are conducted on a computer having Intel Xeon CPU E5-2609 v3@1.90 GHz and 128 GB RAM memory. The testing time of LG-MMC is omitted since it could be run only on another computer with a different operating system. The testing times only show the time spent for the operations in the main clustering algorithms and the time spent for the initialization of the clustering algorithms are not reported. The results are given in Table 4 and the most efficient times are indicated with bold fonts. As seen in the table, our clustering methods are typically the most fast clustering algorithms. For low-dimensional datasets, UCI-Digits 1 &7 and Ionosphere all methods finish the clustering in reasonable times with the exception of LRR on Ionosphere dataset. For high-dimensional feature spaces with the small data sizes (Coil and FaceScrub datasets), all tested methods are quite fast with the exception of P–*k*-means clustering method. This is reasonable since this clustering method solves computationally expensive eigen-decomposition of large matrices in each iteration because of high-dimensional feature sizes. Other large-margin and subspace clustering methods are no affected badly since the dataset size is small. But, when the dataset size increases to 34 K samples, all margin and subspace clustering methods significantly fail and they cannot converge in 24 h with the exception of SMC method which converged in approximately 4.11 h. Because, large margin clustering methods solving the problem in the dual space use kernel matrices with size 34,000 $\times$ 34,000 which is even hard to fit to the memory. In a similar manner, the subspace methods create affinity matrices with the same size and operate on them. Therefore, almost all margin clustering and subspace clustering methods are not feasible even for moderate size datasets including more than 10 K samples. In contrast, our proposed clustering methods are still very efficient and they scale well with the large dataset sizes. For example, our slowest clustering algorithm RCMMC-SG is still 493 times faster than SMC on the Gender 34 K dataset.

6 Discussions ad conclusion

This paper presents new robust clustering methods that aim to maximize the margin between two clusters. The key idea is to split the data into two clusters with the greatest possible margin between them. Additionally, we introduce two novel clustering methods that build upon this approach by combining maximum margin and subspace clustering to produce more condensed clusters. To achieve this, our proposed methods search for two parallel hyperplanes that best fit the cluster samples while maintaining maximum distance from each other. In order to handle noisy or outlier samples, we also introduce more robust loss terms that minimize the influence of these samples. Our experimental results demonstrate that our proposed robust compact clustering method, RCMMC-SG, significantly improves

Table 4 Testing times of the clustering methods

Methods	UCI-Digits 1 &7	Ionosphere	Coil	FaceScrub	Gender 34K
MMC-SG	**0.14 s**	0.14 s	**0.15 s**	**0.18 s**	37.83 s
RMMC-SG	**0.14 s**	**0.13 s**	0.17 s	0.21 s	37.94 s
CMMC-SG	**0.14 s**	0.36 s	0.18 s	0.22 s	140.00 s
RCMMC-SG	0.15 s	0.38 s	1.85 s	2.85 s	300.35 s
IterSVR	0.24 s	0.17 s	2.10 s	4.64 s	$>$ 24 hours
P– k -means	1.07 s	0.40 s	697.50 s	208.73 s	$>$ 24 hours
SMC	1.56 s	1.72 s	1.28 s	2.81 s	148000 s
LRR	1.32 s	9.32 s	0.92 s	2.52 s	$>$ 24 hours
OSC	2.31 s	0.25 s	1.63 s	2.81 s	$>$ 24 hours
SSC	1.64 s	1.57 s	1.45 s	3.22 s	$>$ 24 hours
k-means	0.91 s	0.87 s	0.22 s	0.32 s	**14.64 s**

clustering accuracy, particularly in high-dimensional clustering scenarios.

In our proposed clustering methods, we solve the optimization problem in the primal space. This enables us to use fast and efficient stochastic gradient algorithm, yet we cannot use the kernel functions since the problem is not solved in the dual space. Therefore, the proposed methods will not work well for low-dimensional spaces where the classes have nonlinear decision boundaries. Moreover, the proposed CMMC-SG clustering method which enforces the cluster samples to lie on the hyperplanes only work well when the dimension of the feature space is higher than the number of data samples. If this condition is not met, the data samples typically span the whole feature space and it becomes impossible to fit to the lower-dimensional parallel hyperplanes. This is also the main reason why CMMC-SG performed badly in low-dimensional datasets. Its robust version does not have these limitations since it does not punish samples far from the hyperplanes severely. In general, our proposed methods are mostly suitable for high-dimensional clustering problems where the traditional clustering methods perform poorly. That is why we restrict our focus on high-dimensional clustering problems in this study.

As a potential future research topic, the proposed clustering methods can be adopted to semi-supervised classification settings where there is limited amount of labeled data and many unlabeled data. Also, an interesting line of research can be application of the proposed clustering methods to multi-view data learning problem in which data samples are represented more than one feature set.

Author contributions HC helped in conceptualization, methodology, writing—review & editing, software. EC was involved in conducting experiments, review & editing, software.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). This study has not been funded by any research grant.

Data availability The datasets analyzed during the current study are available in the UCI repository, https://archive.ics.uci.edu/ml/datasets.php.

Code availability We will share the codes of the proposed methods in our website, https://web.ogu.edu.tr/mlcv/Sayfa/Index/12/software, after publication of the paper.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Cevikalp H, Franc V (2017) Large-scale robust transductive support vector machines. Neurocomputing 235:199–209
2. Hofmeyr DP (2017) Clustering by minimum cut hyperplanes. IEEE Trans Pattern Anal Mach Intell 39(8):1547–1560. https://doi.org/10.1109/TPAMI.2016.2609929
3. Bilenko M, Basu S, Mooney RJ (2004) Integrating constraints and metric learning in semi supervised clustering. In: Proceedings of the twenty-first international conference on machine learning (ICML)
4. Xing EP, Ng AY, Jordan MI, Russell S (2003) Distance metric learning with application to clustering with side-information. In: Advances in neural information processing systems (NIPS)
5. Cevikalp H (2010) New clustering algorithms for the support vector machine based hierarchical classification. Pattern Recogn Lett 31:1285–1291
6. Cevikalp H (2019) High-dimensional data clustering by using local affine/convex hulls. Pattern Recogn Lett 128:427–432
7. Hinneburg A, Aggarwal CC, Keim DA (2000) What is the nearest neighbor in high dimensional spaces?. In: Proceedings of the 26th international conference on very large databases
8. Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) When is nearest neighbor meaningful. Lect Notes Comput Sci 1540:217–235
9. Cortes C, Vapnik V (1995) Support vector networks. Mach Learn 20:273–297
10. Zhu Z, Wang Y, Robinson DP, Naiman DQ, Vidal R, Tsakiris MC (2018) Dual principal component pursuit: probability analysis and efficient algorithms. arXiv:1812.09924
11. Elhamifar E, Vidal R (2013) Sparse subspace clustering: algorithm, theory, and applications. CoRR. arXiv:1203.1005
12. Ho J, Yang MH, Lim J, Lee KC, Kriegman D (2003) Clustering appearances of objects under varying illumination conditions. In: 2003 IEEE Computer Society conference on computer vision and pattern recognition (CVPR)
13. Agarwal PK, Mustafa NH (2004) K-means projective clustering. In: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems
14. Zhang T, Szlam A, Lerman G (2009) Median k-flats for hybrid linear modeling with many outliers. In: ICCV workshops
15. Fischler M, Bolles R (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395
16. Tipping M, Bishop C (1999) Mixtures of probabilistic principal component analyzers. Neural Comput 11:443–482
17. Gruber A, Weiss Y (2004) Multibody factorization with uncertainty and missing data using the EM algorithm. In: Proceedings of the 2004 IEEE Computer Society conference on computer vision and pattern recognition (CVPR)

18. Vidal R, Ma Y, Sastry S (2005) Generalized principal component analysis (GPCA). IEEE Trans Pattern Anal Mach Intell 27:1–15
19. Yang AY, Rao SR, Ma Y (2006) Robust statistical estimation and segmentation of multiple subspaces. In: CVPR workshops
20. Elhamifar E, Vidal R (2009) Sparse subspace clustering. In: 2009 IEEE conference on computer vision and pattern recognition (CVPR)
21. Elhamifar E, Vidal R (2011) Sparse manifold clustering and embedding. In: Advances in neural information processing systems (NIPS)
22. Yan J, Pollefeys M (2006) A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: European conference on computer vision
23. Liu G, Lin Z, Yu Y (2010) Robust subspace segmentation by low-rank representation. In: Proceedings of the 27th international conference on machine learning (ICML)
24. Zhang T, Szlam A, Wang Y, Lerman G (2012) Hybrid linear modeling via local best-fit flats. Int J Comput Vis 1000:217–240
25. Wang B, Hu Y, Gao J, Sun Y, Ju F, Yin B (2021) Adaptive fusion of heterogeneous manifolds for subspace clustering. IEEE Trans Neural Netw Learn Syst 32:3484–3497
26. Passalis N, Tefas A (2019) Discriminative clustering using regularized subspace learning. Pattern Recogn 96:106982
27. Vidal R (2011) Subspace clustering. IEEE Signal Process Mag 28(2):52–68. https://doi.org/10.1109/MSP.2010.939739
28. Chapelle O, Zien A (2005) Semi-supervised classification by low density separation. In: Neural information processing systems (NIPS)
29. Xu L, Neufeld J, Larson B, Schuurmans D (2005) Maximum margin clustering. In: Saul L, Weiss Y, Bottou L (eds) Advances in neural information processing systems, vol 17. MIT Press, Cambridge
30. Valizadegan H, Jin R (2007) Generalized maximum margin clustering and unsupervised kernel learning. In: Schölkopf B, Platt J, Hoffman T (eds) Advances in neural information processing systems, vol 19. MIT Press, Cambridge
31. Zhang K, Tsang IW, Kwok JT (2009) Maximum margin clustering made practical. IEEE Trans Neural Netw 20(4):583–596. https://doi.org/10.1109/TNN.2008.2010620
32. Zhang X-L, Wu J (2012) Linearithmic time sparse and convex maximum margin clustering. IEEE Trans Syst Man Cybern Part B (Cybernetics) 42(6):1669–1692. https://doi.org/10.1109/TSMCB.2012.2197824
33. Zhao B, Wang F, Zhang C (2008) Efficient maximum margin clustering via cutting plane algorithm. In: 8th SIAM international conference on data mining 2008, Proceedings in applied mathematics, vol 130. Society for Industrial and Applied Mathematics, pp 751–762. https://doi.org/10.1137/1.9781611972788.68
34. Li YF, Tsang IW, Kwok JT, Zhou ZH (2009) Tighter and convex maximum margin clustering. J Mach Learn Res 5:344–351
35. Wang F, Wang X, Li T (2009) Maximum margin clustering on data manifolds. In: 2009 Ninth IEEE international conference on data mining (ICDM). IEEE Computer Society, pp 1028–1033
36. Hu Y, Wang J, Yu N, Hua XS (2008) Maximum margin clustering with pairwise constraints. In: IEEE international conference on data mining
37. Zeng H, Cheung Y-M (2012) Semi-supervised maximum margin clustering with pairwise constraints. IEEE Trans Knowl Data Eng 24(5):926–939. https://doi.org/10.1109/TKDE.2011.68
38. Hoai M, la Torre FD (2012) Maximum margin temporal clustering. In: Proceedings of 15th international conference on artificial intelligence and statistics (AISTATS '12), pp 520 – 528
39. Chen C, Zhu J, Zhang X (2014) Robust Bayesian max-margin clustering. In: Ghahramani Z, Welling M, Cortes C, Lawrence N, Weinberger KQ (eds) Advances in neural information processing systems, vol 27. Curran Associates Inc, New York
40. Zhou G-T, Lan T, Vahdat A, Mori G (2013) Latent maximum margin clustering. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ (eds) Advances in neural information processing systems, vol 26. Curran Associates Inc, New York
41. Li J, Sun J, Liu L, Liu B, Xiao C, Wang F (2019) Improved maximum margin clustering via the bundle method. IEEE Access 7:63709–63721. https://doi.org/10.1109/ACCESS.2019.2916724
42. Vijaya Saradhi V, Charly Abraham P (2016) Incremental maximum margin clustering. Pattern Anal Appl 19(4):1057–1067. https://doi.org/10.1007/s10044-015-0447-5
43. Xue H, Li S, Chen X, Wang Y (2019) A maximum margin clustering algorithm based on indefinite kernels. Front Comput Sci 13:813–827
44. Xiaoa Y, Zhanga J, Liub B, Zhaoa L, Konga X, Haoc Z (2023) Multi-view maximum margin clustering with privileged information learning. IEEE Trans Circuits Syst Video Technol. https://doi.org/10.1109/TCSVT.2023.3311174
45. Zhang T, Zhou ZH (2018) Optimal margin distribution clustering. In: AAAI conference on artificial intelligence
46. Zhou GT, Hwang SJ, Schmidt M, Sigal L, Mori G (2015) Hierarchical maximum-margin clustering. arXiv:1502.01827
47. Chen G (2015) Deep transductive semi-supervised maximum margin clustering. arXiv:1501.06237
48. Hofmeyr DP (2023) Incremental estimation of low-density separating hyperplanes for clustering large data sets. Pattern Recogn 139:109471
49. Farhadi A, Tabrizi MK (2008) Learning to recognize activities from the wrong view point. In: Forsyth D, Torr P, Zisserman A (eds) Computer vision—ECCV 2008. Springer, Berlin, pp 154–166
50. Hoai M, Zisserman A (2013) Discriminative sub-categorization. In: IEEE conference on computer vision and pattern recognition, pp 1666–1673. https://doi.org/10.1109/CVPR.2013.218
51. Wang Y, Cao L (2013) Discovering latent clusters from geotagged beach images. In: Li S, El Saddik A, Wang M, Mei T, Sebe N, Yan S, Hong R, Gurrin C (eds) Advances in multimedia modeling. Springer, Berlin, pp 133–142
52. Zhu B, Ding Y, Hao K (2014) Multiclass maximum margin clustering via immune evolutionary algorithm for automatic diagnosis of electrocardiogram arrhythmias. Appl Math Comput 227:428–436. https://doi.org/10.1016/j.amc.2013.11.028
53. Rahimi A, Recht B (2004) Clustering with normalized cuts is clustering with a hyperplane. Stat Learn Comput Vis 1–12
54. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905. https://doi.org/10.1109/34.868688
55. Shalev-Shwartz S, Singer Y, Srebro N (2007) Pegasos: Primal estimated sub-grdient solver for svm. In: International conference on machine learning
56. Yuille AL, Rangarajan A (2001) The concave-convex procedure (CCCP). Adv Neural Inf Process Syst
57. Collobert R, Sinz F, Weston J, Bottou L, Joachims T (2006) Large scale transductive SVMs. J Mach Learn Res 7:1687–1712
58. Cevikalp H, Triggs B, Yavuz HS, Kucuk Y, Kucuk M, Barkana A (2010) Large margin classifiers based on affine hulls. Neurocomputing 73:3160–3168
59. Tierney S, Gao J, Guo Y (2014) Subspace clustering for sequential data. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
60. Aeberhard S, Forina M (1991) Wine. UCI Mach Learn Repos 10:C5PC7J
61. Alpaydin E, Kaynak C (1998) Optical recognition of handwritten digits. UCI Mach Learn Repos. https://doi.org/10.24432/C50P49

62. Slate D (1991) Letter recognition. UCI Mach Learn Repos. https://doi.org/10.24432/C5ZP40
63. Srinivasan A (1993) Statlog (Landsat Satellite). UCI Mach Learn Repos. https://doi.org/10.24432/C55887
64. Hull JJ (1994) A database for handwritten text recognition research. IEEE Trans Pattern Anal Mach Intell 16(5):550–554. https://doi.org/10.1109/34.291440
65. Sigillito V, Wing S, Hutton L, Baker K (1989) Ionosphere. UCI Mach Learn Repos. https://doi.org/10.24432/C5W01B
66. Nayar S, Murase H (1996) Columbia object image library: coil-100. Tech. Rep. CUCS-006-96, Department of Computer Science, Columbia University
67. Ng H, Winkler S (2014) A data-driven approach to cleaning large face datasets. In: IEEE international conference on image processing (ICIP), pp 343–347
68. Lang K (1995) Newsweeder: learning to filter netnews. In: International conference on machine learning
69. Saha AH (2018) A machine learning approach to radiogenomics of breast cancer: a study of 922 subjects and 529 DCE-MRI features. Br J Cancer 119:508–516
70. Huang GB, Ramesh M, Berg T, Learned-Miller E (2007) Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst
71. Kumar N, Berg AC, Belhumeur PN, Nayar SK (2009) Attribute and simile classifiers for face verification. In: IEEE international conference on computer vision (ICCV)
72. Minear M, Park D (2004) A lifespan database of adult facial stimuli. Behav Res Methods Instrum Comput 36:630–633
73. Uřičář M, Franc V, Hlaváč V (2012) Detector of facial landmarks learned by the structured output SVM. In: Csurka G, Braz J (eds) VISAPP '12: Proceedings of the 7th international conference on computer vision theory and applications, vol 1. SciTePress—Science and Technology Publications, Porto, pp 547–556