Pattern Recognition 81 (2018) 575–584

Contents lists available at ScienceDirect

Pattern Recognition

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

Convex clustering with metric learning

Xiaopeng Lucia Sui[a,*], Li Xu[b], Xiaoning Qian[a], Tie Liu[a]

[a] Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA
[b] Institute of Computing Technology, Chinese Academy of Sciences, No. 6 Kexuyuan South Road, Haidian District, 100190, Beijing, China

ARTICLE INFO

Article history:
Received 20 September 2017
Revised 6 February 2018
Accepted 19 April 2018
Available online 21 April 2018

MSC:
62H30
90C25
68T10

Keywords:
Convex clustering
Alternating Direction Method of Multipliers
Metric learning
Unsupervised learning

ABSTRACT

The convex clustering formulation of Chi and Lange (2015) is revisited. While this formulation can be precisely and efficiently solved, it uses the standard Euclidean metric to measure the distance between the data points and their corresponding cluster centers and hence its performance deteriorates significantly in the presence of outlier features. To address this issue, this paper considers a formulation that combines convex clustering with metric learning. It is shown that: (1) for any given positive definite Mahalanobis distance metric, the problem of convex clustering can be precisely and efficiently solved using the Alternating Direction Method of Multipliers; (2) the problem of learning a positive definite Mahalanobis distance metric admits a closed-form solution; (3) an algorithm that alternates between convex clustering and metric learning can provide a significant performance boost over not only the original convex clustering formulation but also the recently proposed robust convex clustering formulation of Wang et al. (2017).

1. Introduction

Clustering refers to a procedure that groups similar objects together while separating dissimilar ones apart. This simple idea has a wide range of applications in different areas of scientific research. For example, in bioinformatics, clustering can identify co-expressed genes that work together for the same metabolic pathway [1]; in neuroscience,clustering can identify regions of neurons in the brain that are physically or functionally connected [2]; and in document and image analysis, clustering can identify handwritten words or characters in different languages [3,4]. Due to its fundamental importance, clustering has been an extensively studied topic in the literature, and many algorithms have been developed based on various problem formulations [5–10].

A common challenge for developing clustering algorithms is that many clustering formulations are inherently difficult to solve and in practice can only be *approximately* solved based on various heuristics. The famous *k*-means [6] and normalized-cut [5] algorithms are two prime examples. One interesting exception is the recently proposed *Convex Clustering (CC)* formulation by Chi and Lange [11].[1] In their formulation, each data point is associated with a cluster center, and the goal is to minimize the aggregate distance between the data points and their corresponding cluster centers. A regularization term is then added to the objective function to leverage *group sparsity* to the clustering solution. Varying the weight of the regularization term creates a clustering path that may contain multiple meaningful solutions. More importantly, as demonstrated in [11], this formulation leads to a convex optimization problem, which can be *precisely and efficiently* solved using the well-known Alternating Direction Method of Multipliers (ADMM) [13–15].

One potential drawback about the CC formulation of Chi and Lange [11] is that it uses the standard *Euclidean* metric to measure the distance between the data points and their corresponding cluster centers. As is well known, the Euclidean metric treats each feature of the data *equally*, and as a result, the performance of the CC algorithm of Chi and Lange [11] deteriorates significantly in the presence of *outlier* features.

To address this issue, Wang et al. [16] proposed the so-called *Robust Convex Clustering (RCC)* formulation, in which they introduced the so-called *robust component* to explicitly identify the outlier features of the data. By assuming that the outlier features are *sparse*, it was shown [16] that the robust component can be learned from the *unlabeled* data. However, even though RCC [16] can provide a performance boost over the CC algorithm of Chi and Lange [11], the underlying modeling assumption that the outlier features are sparse can be questionable. For example, for many

* Corresponding author.
E-mail address: xlsui@tamu.edu (X.L. Sui).

[1] Using convex optimization techniques to solve clustering problems has also been previously explored in [7,12].

https://doi.org/10.1016/j.patcog.2018.04.019
0031-3203/

real-world data sets, it is the *highly relevant* features, rather than the outlier features, that are sparse.

In this paper, we revisit the problem of CC by incorporating *Metric Learning (ML)* into the problem formulation. The fact that the accuracy of clustering can be significantly improved with a distance metric that is tailored to the specific data set is well documented [17], and the problem of learning an appropriate distance metric from a given labeled or partially labeled training data set has received a significant amount of attention in the literature lately [18–27]. The main challenge here is that for clustering there are usually no labeled or partially labeled training data available, so ML has to be done in an *unsupervised* fashion. Note that this situation is rather similar to that for learning the robust component in the RCC formulation [16]. Similar to [16], we shall consider an *alternating* procedure that alternates between CC and ML.

More specifically, in this paper we show that: (1) for any given positive definite *Mahalanobis* distance metric [19,28], the problem of CC can be precisely and efficiently solved within the ADMM framework [13–15]. In particular, at each iteration of the ADMM, the cluster centers can be efficiently updated via solving a *Sylvester* equation [29]; and (2) when considering the family of positive definite Mahalanobis distances for CC, the problem of ML admits a *closed-form* solution. This is in sharp contrast to many other ML problems considered in the literature [18–24]. Through simulated and real-world data sets, we show that the proposed algorithm that combines CC with ML can provide significant performance boosts over both CC and RCC.

The rest of the paper is organized as follows. Next in Section 2, we review the formulations of CC [11] and RCC [16] and the proposed algorithms for solving them. In Section 3, we discuss in detail the problem of CC under a positive definite Mahalanobis distance metric, and present our main results on an efficient clustering algorithm based on the ADMM framework and the closed-form solution for the corresponding ML problem. In Section 4, we use one set of synthetic data and three sets of real-world data collected from the UCI Machine Learning Repository [30] to benchmark the performance of CC, RCC, and the proposed *Convex Clustering with Metric Learning (CCML)*. Finally in Section 5, we conclude the paper with some remarks.

2. Convex clustering and robust convex clustering

2.1. Convex clustering

Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a collection of N data points to be clustered, and let $\mathbf{X}$ be the data matrix for which the jth column is given by $\mathbf{x}_j$ (so each row of $\mathbf{X}$ represents a feature of the data). In [11], the CC problem was formulated as the following optimization problem:

$$\underset{\mathbf{U}}{\text{Minimize}} \quad \frac{1}{2}\sum_{j=1}^{N}\|\mathbf{x}_j-\mathbf{u}_j\|_2^2+\gamma\sum_{1\le j_1<j_2\le N} w_{\{j_1,j_2\}}\|\mathbf{u}_{j_1}-\mathbf{u}_{j_2}\|_1 \quad (1)$$

where γ is a positive tuning constant, $w_{\{j_1,j_2\}}$ is a nonnegative weight, and the jth column $\mathbf{u}_j$ of the matrix $\mathbf{U}$ is the center of the cluster that the data point $\mathbf{x}_j$ belongs to. Multiple data points that belong to the same cluster will have the same cluster center vector, thus the columns of $\mathbf{U}$ are not unique: If there are k clusters, there will be k unique cluster centers, i.e. k unique columns of $\mathbf{U}$. Clearly, the goal of this convex optimization problem is to cluster the set of data points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ such that the aggregate distance between the data points and their corresponding cluster centers is minimized. The second term in the objective function is a regularizer that leverages group sparsity to control the complexity (the number of clusters) of the clustering solution.

To solve the optimization problem using the aforementioned ADMM framework, let $\mathcal{E}$ be the set of edges in a complete graph with nodes $1, 2, \ldots, N$, *i.e.*, $\mathcal{E}=\{\{j_1,j_2\}: 1\le j_1<j_2\le N\}$. We define a one-to-one edge-labeling mapping $\phi: \{1,2,\ldots,\varepsilon\}\longrightarrow\mathcal{E}$ with $\varepsilon=N(N-1)/2$, and let $\phi_1(\ell)=j_1$ and $\phi_2(\ell)=j_2$ if $\{j_1,j_2\}=\phi(\ell)$ and $j_1<j_2$. For notational simplicity, let $w_\ell:=w_{\phi(\ell)}$ for $1\le\ell\le\varepsilon$. For each $1\le\ell\le\varepsilon$, let $\mathbf{v}_\ell:=\mathbf{u}_{\phi_1(\ell)}-\mathbf{u}_{\phi_2(\ell)}$ be the difference between the centroids $\mathbf{u}_{\phi_1(\ell)}$ and $\mathbf{u}_{\phi_2(\ell)}$. The matrix $\mathbf{V}$ is given by the collection of $\mathbf{v}_\ell$, $1\le\ell\le\varepsilon$ as its columns. With this notion of the matrix $\mathbf{V}$, the convex clustering problem (1) can be recast as the following constrained optimization problem:

$$\begin{aligned}&\underset{\mathbf{U},\mathbf{V}}{\text{Minimize}} \quad \frac{1}{2}\sum_{j=1}^{N}\|\mathbf{x}_j-\mathbf{u}_j\|_2^2+\gamma\sum_{\ell=1}^{\varepsilon}w_\ell\|\mathbf{v}_\ell\|_1\\ &\text{Subject to} \quad \mathbf{u}_{\phi_1(\ell)}-\mathbf{u}_{\phi_2(\ell)}-\mathbf{v}_\ell=\mathbf{0}, \quad 1\le\ell\le\varepsilon. \end{aligned} \quad (2)$$

Considering a vectorization of $\mathbf{U}$ and $\mathbf{V}$, the optimization problem (2) is a special case of the following general optimization problem:

$$\begin{aligned}&\underset{\mathbf{u},\mathbf{v}}{\text{Minimize}} \quad f(\mathbf{u})+g(\mathbf{v})\\ &\text{Subject to} \quad \mathbf{A}_1\mathbf{u}+\mathbf{A}_2\mathbf{v}=\mathbf{c}.\end{aligned} \quad (3)$$

The *augmented* Lagrangian of this general optimization problem is given by:

$$\begin{aligned}\mathcal{L}_\nu(\mathbf{u},\mathbf{v},\boldsymbol{\lambda}) :&= f(\mathbf{u})+g(\mathbf{v})+\langle\boldsymbol{\lambda},\mathbf{c}-\mathbf{A}_1\mathbf{u}-\mathbf{A}_2\mathbf{v}\rangle\\ &\quad+\frac{\nu}{2}\|\mathbf{c}-\mathbf{A}_1\mathbf{u}-\mathbf{A}_2\mathbf{v}\|_2^2,\end{aligned} \quad (4)$$

where $\boldsymbol{\lambda}$ is a vector of Lagrangian multipliers, and ν is a nonnegative tuning parameter. The ADMM minimizes the augmented Lagrangian $\mathcal{L}_\nu(\mathbf{u},\mathbf{v},\boldsymbol{\lambda})$ over its variables **u, v** and $\boldsymbol{\lambda}$ separately and one block of variables at a time. This leads to the following sequential updates for **u, v**, and $\boldsymbol{\lambda}$:

$$\begin{aligned}\mathbf{u}^{m+1} &:= \arg\min_{\mathbf{u}} \mathcal{L}_\nu(\mathbf{u},\mathbf{v}^m,\boldsymbol{\lambda}^m);\\ \mathbf{v}^{m+1} &:= \arg\min_{\mathbf{v}} \mathcal{L}_\nu(\mathbf{u}^{m+1},\mathbf{v},\boldsymbol{\lambda}^m);\\ \boldsymbol{\lambda}^{m+1} &:= \boldsymbol{\lambda}^m+\nu(\mathbf{c}-\mathbf{A}_1\mathbf{u}^{m+1}-\mathbf{A}_2\mathbf{v}^{m+1}).\end{aligned} \quad (5)$$

The CC algorithm proposed in [11] is based on calculating the updates of $\mathbf{u}^{m+1}$ and $\mathbf{v}^{m+1}$ *efficiently* until convergence. We shall describe these updates as a special case of our more general CC algorithm under a positive definite Mahalanobis distance metric in the next section.

2.2. Robust convex clustering

To improve the performance of CC in the presence of the outlier features, Wang et al. [16] proposed the following RCC problem:

$$\begin{aligned}\underset{\mathbf{U},\mathbf{Q}}{\text{Minimize}} \quad &\frac{1}{2}\sum_{j=1}^{N}\|\mathbf{x}_j-(\mathbf{u}_j+\mathbf{q}_j)\|_2^2+\gamma\sum_{1\le j_1<j_2\le N}w_{\{j_1,j_2\}}\|\mathbf{u}_{j_1}\\ &-\mathbf{u}_{j_2}\|_1+\beta\|\mathbf{Q}\|_{2,1}\end{aligned} \quad (6)$$

where the matrix $\mathbf{Q}$ is the so-called *robust component* for which the jth column is given by $\mathbf{q}_j$, and β is a second tuning parameter in addition to γ. The penalization term $\beta\|\mathbf{Q}\|_{2,1}$ is introduced to achieve *row-wise* sparsity: If a feature is relevant, the corresponding row in $\mathbf{Q}$ will be zero for all elements; if a feature is an outlier, this row will be non-zero.

To solve the optimization problem (6), Wang et al. [16] proposed an alternating procedure that alternates between CC (minimizing over $\mathbf{U}$) and learning the robust component $\mathbf{Q}$. More specifically, for a fixed $\mathbf{Q}$, the optimization problem (6) reduces to the original CC problem (1) with the data set $\mathbf{X}$ replaced by $\mathbf{X}-\mathbf{Q}$. For a fixed $\mathbf{U}$, the optimization problem (6) admits a closed-form solution for $\mathbf{Q}$ whose ith row is given by [16]:

$$\max\left(0, 1-\frac{\beta}{\|(\mathbf{X}-\mathbf{U})_i\|_2}\right)(\mathbf{X}-\mathbf{U})_i, \tag{7}$$

where $(\mathbf{X}-\mathbf{U})_i$ denotes the ith row of the matrix $\mathbf{X}-\mathbf{U}$. Thus, to solve the optimization problem (6), we may begin by setting the robust component $\mathbf{Q}$ as zero and perform CC. For the next iterations, one may alternate between learning the robust component according to (7) and CC, where learning the robust component is based on the optimal $\mathbf{U}$ obtained from the previous iteration, and CC is then based on the just-updated robust component $\mathbf{Q}$. We may continue such iterations till the solutions converge.

3. Convex clustering with metric learning

To incorporate ML into the formulation of CC, let $\mathbf{B}$ be a positive definite matrix and consider the following optimization problem:

$$\begin{aligned}\underset{\mathbf{U},\mathbf{B}}{\text{Minimize}} \quad & \frac{1}{2}\sum_{j=1}^{N}\left(\mathbf{x}_j-\mathbf{u}_j\right)^T\mathbf{B}\left(\mathbf{x}_j-\mathbf{u}_j\right)\\ & +\gamma\sum_{1\le j_1<j_2\le N} w_{\{j_1,j_2\}}\|\mathbf{u}_{j_1}-\mathbf{u}_{j_2}\|_1\\ \text{Subject to} \quad & \log\det(\mathbf{B})\ge 0,\end{aligned} \tag{8}$$

where the choice of the constraint $\log\det(\mathbf{B})\ge 0$ was motivated by Hoi et al. [31] and ensures that the matrix $\mathbf{B}$ has a *full* rank. (As we shall see, maintaining the full rank of the matrix $\mathbf{B}$ is also crucial for developing the proper convex clustering algorithm.) The structure of the matrix $\mathbf{B}$ shows which features of the data are more congruent with the cluster assignment. In particular, when $\mathbf{B}$ is diagonal, the larger diagonal values of $\mathbf{B}$ correspond to the features that are of higher relevance or of lower noise corruptions. Note that for the original CC formulation [11] where $\mathbf{B}$ is an identity matrix, all features are uniformly weighted for clustering, which can be very suboptimal in the presence of outlier features. For a general positive definite $\mathbf{B}$, its operational meaning can be understood through the standard singular value decomposition.

To solve the optimization problem (8), we shall consider an alternating procedure that alternates between CC (minimizing over $\mathbf{U}$) and ML (minimizing over $\mathbf{B}$).

3.1. Solving $\mathbf{U}$ for a fixed $\mathbf{B}$

Fix $\mathbf{B}$ to be positive definite matrix and consider the artificial variables $\mathbf{v}_\ell := \mathbf{u}_{\phi_1(\ell)}-\mathbf{u}_{\phi_2(\ell)}$ for $1\le\ell\le\varepsilon$. The optimization problem (8) can be equivalently written as:

$$\begin{aligned}\underset{\mathbf{U},\mathbf{V}}{\text{Minimize}} \quad & \frac{1}{2}\sum_{j=1}^{N}\left(\mathbf{x}_j-\mathbf{u}_j\right)^T\mathbf{B}\left(\mathbf{x}_j-\mathbf{u}_j\right)+\gamma\sum_{\ell=1}^{\varepsilon} w_\ell\|\mathbf{v}_\ell\|_1\\ \text{Subject to} \quad & \mathbf{u}_{\phi_1(\ell)}-\mathbf{u}_{\phi_2(\ell)}-\mathbf{v}_\ell=\mathbf{0},\quad 1\le\ell\le\varepsilon.\end{aligned} \tag{9}$$

Note that when $\mathbf{B}$ is an identity matrix, the optimization problem (9) reduces to the original CC formulation (2), which can be solved efficiently and precisely using the ADMM framework.

To apply the ADMM framework to solve the optimization problem (9), note that its augmented Lagrangian is given by:

$$\begin{aligned}\mathcal{L}_\nu(\mathbf{U},\mathbf{V},\mathbf{\Lambda}) := {} & \frac{1}{2}\sum_{j=1}^{N}\left(\mathbf{x}_j-\mathbf{u}_j\right)^T\mathbf{B}\left(\mathbf{x}_j-\mathbf{u}_j\right)+\gamma\sum_{\ell=1}^{\varepsilon} w_\ell\|\mathbf{v}_\ell\|_1\\ & +\sum_{\ell=1}^{\varepsilon}\boldsymbol{\lambda}_\ell^T\left(\mathbf{v}_\ell-\mathbf{u}_{\phi_1(\ell)}+\mathbf{u}_{\phi_2(\ell)}\right)\\ & +\frac{\nu}{2}\sum_{\ell=1}^{\varepsilon}\|\mathbf{v}_\ell-\mathbf{u}_{\phi_1(\ell)}+\mathbf{u}_{\phi_2(\ell)}\|_2^2,\end{aligned} \tag{10}$$

where $\mathbf{\Lambda} := (\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_\varepsilon)$. We shall update $\mathbf{U}$ and $\mathbf{V}$ in each iteration of the ADMM according to the procedure described in (5).

Updating $\mathbf{U}$. To update $\mathbf{U}$, we need to minimize the function

$$\begin{aligned}f(\mathbf{U}) := {} & \frac{1}{2}\sum_{j=1}^{N}\left(\mathbf{x}_j-\mathbf{u}_j\right)^T\mathbf{B}\left(\mathbf{x}_j-\mathbf{u}_j\right)+\frac{\nu}{2}\sum_{\ell=1}^{\varepsilon}\|\tilde{\mathbf{v}}_\ell\\ & -\mathbf{u}_{\phi_1(\ell)}+\mathbf{u}_{\phi_2(\ell)}\|_2^2,\end{aligned} \tag{11}$$

where $\tilde{\mathbf{v}}_l := \mathbf{v}_\ell+\nu^{-1}\boldsymbol{\lambda}_\ell$. Let $\mathbf{u} := \vec{(}\mathbf{U})$ and $\mathbf{x} := \vec{(}\mathbf{X})$. Then, the function $f(\mathbf{U})$ can be equivalently written as:

$$f(\mathbf{u})=\frac{1}{2}(\mathbf{x}-\mathbf{u})^T\mathcal{B}(\mathbf{x}-\mathbf{u})+\frac{\nu}{2}\sum_{\ell=1}^{\varepsilon}\|\mathbf{E}_\ell\mathbf{u}-\tilde{\mathbf{v}}_\ell\|_2^2, \tag{12}$$

where $\mathcal{B} := \mathbf{I}\otimes\mathbf{B}$ and $\mathbf{E}_\ell := (\mathbf{e}_{\phi_1(\ell)}-\mathbf{e}_{\phi_2(\ell)})^T\otimes\mathbf{I}$. We can further simplify $f(\mathbf{u})$ as follows. Let

$$\mathbf{E} := \begin{pmatrix}\mathbf{E}_1\\ \vdots\\ \mathbf{E}_\varepsilon\end{pmatrix} \quad\text{and}\quad \tilde{\mathbf{v}} := \begin{pmatrix}\tilde{\mathbf{v}}_1\\ \vdots\\ \tilde{\mathbf{v}}_\varepsilon\end{pmatrix}. \tag{13}$$

Then

$$f(\mathbf{u})=\frac{1}{2}(\mathbf{x}-\mathbf{u})^T\mathcal{B}(\mathbf{x}-\mathbf{u})+\frac{\nu}{2}(\mathbf{E}\mathbf{u}-\tilde{\mathbf{v}})^T(\mathbf{E}\mathbf{u}-\tilde{\mathbf{v}}). \tag{14}$$

We calculate the optimality condition for minimizing the quadratic function (14) as:

$$\left(\mathcal{B}+\nu\mathbf{E}^T\mathbf{E}\right)\mathbf{u}=\mathcal{B}\mathbf{x}+\nu\mathbf{E}^T\tilde{\mathbf{v}}. \tag{15}$$

Note that

$$\mathbf{E}^T\mathbf{E}=\left[\sum_{\ell=1}^{\varepsilon}\left(\mathbf{e}_{\phi_1(\ell)}-\mathbf{e}_{\phi_2(\ell)}\right)\left(\mathbf{e}_{\phi_1(\ell)}-\mathbf{e}_{\phi_2(\ell)}\right)^T\right]\otimes\mathbf{I} \tag{16}$$

$$=\left(N\mathbf{I}-\mathbf{1}\mathbf{1}^T\right)\otimes\mathbf{I} \tag{17}$$

and

$$\mathbf{E}^T\tilde{\mathbf{v}}=\sum_{\ell=1}^{\varepsilon}\left[\left(\mathbf{e}_{\phi_1(\ell)}-\mathbf{e}_{\phi_2(\ell)}\right)\otimes\mathbf{I}\right]\tilde{\mathbf{v}}_\ell. \tag{18}$$

Then, the optimality condition (15) can be written as:

$$\left[\mathbf{I}\otimes\mathbf{B}+\nu\left(N\mathbf{I}-\mathbf{1}\mathbf{1}^T\right)\otimes\mathbf{I}\right]\mathbf{u}=\mathcal{B}\mathbf{x}+\nu\sum_{\ell=1}^{\varepsilon}\left[\left(\mathbf{e}_{\phi_1(\ell)}-\mathbf{e}_{\phi_2(\ell)}\right)\otimes\mathbf{I}\right]\tilde{\mathbf{v}}_\ell, \tag{19}$$

yielding the following equivalent linear system:

$$\mathbf{B}\mathbf{U}+\mathbf{U}\mathbf{D}=\mathbf{B}\mathbf{X}+\mathbf{R}, \tag{20}$$

where $\mathbf{D} := \nu\left(N\mathbf{I}-\mathbf{1}\mathbf{1}^T\right)$ and $\mathbf{R} := \nu\sum_{\ell=1}^{\varepsilon}[\tilde{\mathbf{v}}_l(\mathbf{e}_{\phi_1(\ell)}-\mathbf{e}_{\phi_2(\ell)})^T]$. Note that the system Eq. (20) is in fact a *Sylvester* Eq. [32].

By assumption $\mathbf{B}$ is positive definite so all eigenvalues of $\mathbf{B}$ are positive, while the eigenvalues of $-\mathbf{D}$ are $0,-N,\ldots,-N$. By the unique solution criterion [32], the Sylvester equation (20) must have a unique solution. To solve (20), note that when $\mathbf{B}=\mathbf{I}$, we simply have $\mathbf{U}=(\mathbf{X}+\mathbf{R})(\mathbf{I}+\mathbf{D})^{-1}$. This is the update procedure proposed in [11]. For a general positive definite $\mathbf{B}$, we can first transform $\mathbf{B}$ into a lower real Schur form [29] and $\mathbf{D}$ into an upper real Schur form as follows:

$$\tilde{\mathbf{B}} := \mathbf{P}^T\mathbf{B}\mathbf{P}=\begin{bmatrix}\tilde{\mathbf{B}}_{1,1} & & (0,0)\mathbf{0} & \\ \tilde{\mathbf{B}}_{2,1} & \tilde{\mathbf{B}}_{2,2} & & \\ \vdots & \vdots & \ddots & \\ \tilde{\mathbf{B}}_{d,1} & \tilde{\mathbf{B}}_{d,2} & \cdots & \tilde{\mathbf{B}}_{d,d}\end{bmatrix} \tag{21}$$

and

$$\tilde{\mathbf{D}} := \mathbf{Q}^T\mathbf{D}\mathbf{Q} = \begin{bmatrix} \tilde{\mathbf{D}}_{1,1} & \tilde{\mathbf{D}}_{2,1} & \cdots & \tilde{\mathbf{D}}_{r,1} \\ & \tilde{\mathbf{D}}_{2,2} & \cdots & \tilde{\mathbf{D}}_{r,2} \\ & (0,0)\mathbf{0} & \ddots & \vdots \\ & & & \tilde{\mathbf{D}}_{r,r} \end{bmatrix}, \tag{22}$$

where $\tilde{\mathbf{B}}$ is lower quasi-triangular, $\tilde{\mathbf{D}}$ is upper quasi-triangular, the diagonal blocks $\tilde{\mathbf{B}}_{i,i}$ and $\tilde{\mathbf{D}}_{i,i}$ are order of at most two, and $\mathbf{P}$ and $\mathbf{Q}$ are both orthogonal. Then, we can solve the transformed equation

$$\tilde{\mathbf{B}}\tilde{\mathbf{U}} + \tilde{\mathbf{U}}\tilde{\mathbf{D}} = \mathbf{P}^T(\mathbf{B}\mathbf{X} + \mathbf{R})\mathbf{Q} = \tilde{\mathbf{B}}\mathbf{P}^T\mathbf{X}\mathbf{Q} + \mathbf{P}^T\mathbf{R}\mathbf{Q} \tag{23}$$

by *backward substitutions* [33]. The solution of the original Eq. (20) is thus given by $\mathbf{U} = \mathbf{P}\tilde{\mathbf{U}}\mathbf{Q}^T$.

Updating $\mathbf{V}$. To update $\mathbf{V}$, observe that the augmented Lagrangian $\mathcal{L}_\nu(\mathbf{U}, \mathbf{V}, \mathbf{\Lambda})$ is *separable* in the vectors $\mathbf{v}_\ell$. Thus, for any $1 \le \ell \le \varepsilon$, $\mathbf{v}_\ell$ can be updated as [11]:

$$\begin{aligned} \mathbf{v}_\ell &= \arg\min_{\mathbf{v}} \left[\frac{1}{2}\|\mathbf{v} - \left(\mathbf{u}_{\phi_1(\ell)} - \mathbf{u}_{\phi_2(\ell)} - \nu^{-1}\boldsymbol{\lambda}_\ell\right)\|_2^2 + \frac{\gamma w_\ell}{\nu}\|\mathbf{v}\|_1 \right] \\ &= \mathcal{S}\left(\mathbf{u}_{\phi_1(\ell)} - \mathbf{u}_{\phi_2(\ell)} - \nu^{-1}\boldsymbol{\lambda}_\ell, \frac{\gamma w_\ell}{\nu}\mathbf{1}\right), \end{aligned} \tag{24}$$

where $\mathcal{S}$ is the element-wise soft-thresholding function given by $\mathcal{S}(\mathbf{x}, \mathbf{a}) := (\mathbf{x} - \mathbf{a})_+ - (-\mathbf{x} - \mathbf{a})_+$.

Algorithm, convergence, and complexity. Algorithm 1 summarizes

Algorithm 1 Solving $\mathbf{U}$ for a fixed $\mathbf{B}$ via the ADMM.

Input: $\mathbf{X}, \mathbf{B}, \gamma, \nu$ **and** $\{w_l\}_{\ell=1}^{\varepsilon}$.
Output: $\mathbf{U}, \mathbf{V}$, **and** $\mathbf{\Lambda}$.

1: Set the maximum number of iterations ω.
2: Initialize $\mathbf{\Lambda}^{(0)}$ and $\mathbf{V}^{(0)}$.
3: $\mathbf{D} := \nu\left(N\mathbf{I} - \mathbf{1}\mathbf{1}^T\right)$.
4: Find the Schur forms $\tilde{\mathbf{B}} = \mathbf{P}^T\mathbf{B}\mathbf{P}$ and $\tilde{\mathbf{D}} = \mathbf{Q}^T\mathbf{D}\mathbf{Q}$ of $\mathbf{B}$ and $\mathbf{D}$ by (22), respectively.
5: $\tilde{\mathbf{X}} := \tilde{\mathbf{B}}\mathbf{P}^T\mathbf{X}\mathbf{Q}$.
6: **for** $m = 1, 2, 3, \ldots, \omega$ **do**
7: $\quad \mathbf{R}^{(m)} := \nu\sum_{\ell=1}^{\varepsilon}\left[(\mathbf{v}_\ell^{(m-1)} + \nu^{-1}\boldsymbol{\lambda}_\ell^{(m-1)})(\mathbf{e}_{\phi_1(\ell)} - \mathbf{e}_{\phi_2(\ell)})^T\right]$.
8: $\quad$ Find the solution $\tilde{\mathbf{U}}^{(m)}$ of $\tilde{\mathbf{B}}\tilde{\mathbf{U}} + \tilde{\mathbf{U}}\tilde{\mathbf{D}} = \tilde{\mathbf{X}} + \mathbf{P}^T\mathbf{R}^{(m)}\mathbf{Q}$ by backward substitution.
9: $\quad \mathbf{U}^{(m)} := \mathbf{P}\tilde{\mathbf{U}}^{(m)}\mathbf{Q}^T$.
10: $\quad$ **for** $\ell = 1, 2, \ldots, \varepsilon$ **do**
11: $\quad\quad \mathbf{v}_\ell^{(m)} := \mathcal{S}\left(\mathbf{u}_{\phi_1(\ell)}^{(m)} - \mathbf{u}_{\phi_2(\ell)}^{(m)} - \nu^{-1}\boldsymbol{\lambda}_\ell^{(m-1)}, \frac{\gamma w_\ell}{\nu}\mathbf{1}\right)$
12: $\quad\quad \boldsymbol{\lambda}_\ell^{(m)} := \boldsymbol{\lambda}_\ell^{(m-1)} + \nu\left(\mathbf{v}_\ell^{(m)} - \mathbf{u}_{\phi_1(\ell)}^{(m)} + \mathbf{u}_{\phi_2(\ell)}^{(m)}\right)$.
13: $\quad$ **end for**
14: **end for**
15: **return** $\mathbf{U} := \mathbf{U}^{(\omega)}$, $\mathbf{V} := \mathbf{V}^{(\omega)}$, and $\mathbf{\Lambda} := \mathbf{\Lambda}^{(\omega)}$.

the updates of $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{\Lambda}$ in the ADMM. It is straightforward to verify that the optimization problem (9) satisfies Slater's condition [34] and hence that the strong duality holds. It then follows from the saddle-point property [35] that there exists a $(\mathbf{U}^*, \mathbf{V}^*, \mathbf{\Lambda}^*)$ such that the un-augmented Lagrangian $\mathcal{L}_0$ satisfies:

$$\mathcal{L}_0(\mathbf{U}^*, \mathbf{V}^*, \mathbf{\Lambda}) \le \mathcal{L}_0\left(\mathbf{U}^*, \mathbf{V}^*, \mathbf{\Lambda}^*\right) \le \mathcal{L}_0\left(\mathbf{U}, \mathbf{V}, \mathbf{\Lambda}^*\right) \tag{25}$$

for any $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{\Lambda}$. We may thus conclude by the convergence criterion of ADMM [13,15] that Algorithm 1 converges to the optimal value of the optimization problem (9). Finally, we note that the computational complexity for solving the Sylvester equation (20) is $O(d^3 + d^2N + dN^2 + N^3)$ [33], where d is the number of features of the data and N is the number of data points. Considering that N is usually much larger than d, this is rather comparable to the $O(N^3)$ complexity for inverting the matrix $\mathbf{I} + \mathbf{D}$ needed for solving the original CC formulation of Chi and Lange [11].

3.2. Solving $\mathbf{B}$ for a fixed $\mathbf{U}$

Fixing $\mathbf{U}$, the optimization problem (8) can be equivalently written as:

$$\begin{aligned} \underset{\mathbf{B}}{\text{Minimize}} \quad & \sum_{j=1}^{N}\left(\mathbf{x}_j - \mathbf{u}_j\right)^T\mathbf{B}\left(\mathbf{x}_j - \mathbf{u}_j\right) \\ \text{Subject to} \quad & \log\det(\mathbf{B}) \ge 0. \end{aligned} \tag{26}$$

To find an optimal solution for $\mathbf{B}$, let

$$\mathbf{A} := \sum_{j=1}^{N}(\mathbf{x}_j - \mathbf{u}_j)(\mathbf{x}_j - \mathbf{u}_j)^T = (\mathbf{X} - \mathbf{U})(\mathbf{X} - \mathbf{U})^T. \tag{27}$$

The Lagrangian of (26) is given by:

$$\mathcal{L}(\mathbf{B}, \mu) = \sum_{j=1}^{N}(\mathbf{x}_j - \mathbf{u}_j)^T\mathbf{B}(\mathbf{x}_j - \mathbf{u}_j) - \mu\log\det(\mathbf{B}) \tag{28}$$

$$= \mathrm{tr}(\mathbf{A}\mathbf{B}) - \mu\log\det(\mathbf{B}). \tag{29}$$

Its Karush–Kuhn–Tucker conditions yield:

$$\mathbf{0} = \frac{\partial\mathcal{L}}{\partial\mathbf{B}} = \mathbf{A}^T - \mu\left(\mathbf{B}^{-1}\right)^T \tag{30}$$

$$\log\det(\mathbf{B}) \ge 0 \tag{31}$$

$$\mu \ge 0 \tag{32}$$

$$\mu\log\det(\mathbf{B}) = 0. \tag{33}$$

Assuming that $\mathbf{A}$ has a full rank, i.e., no features are completely redundant, a closed-form solution of (26) is given by:

$$\mathbf{B} = \det(\mathbf{A})\mathbf{A}^{-1}. \tag{34}$$

3.3. Iteration between convex clustering and metric learning

To solve the optimization problem (8), we shall begin by setting the matrix $\mathbf{B}$ as an identity matrix and perform Algorithm 1. This is equivalent to the ADMM algorithm for CC proposed in [11]. For the next iterations, we alternate between ML according to (34) and CC according to Algorithm 1, where ML is based on the optimal $\mathbf{U}$ obtained from the previous iteration, and CC is then based on the just-updated matrix $\mathbf{B}$ from the ML. We may continue such iterations till the solutions converge to a local minimum.

4. Numerical experiments

In this section, we use one set of synthetic data and three sets of real-world data to benchmark the performance of the proposed CCML with those of CC [11] and RCC [16]. Next, we shall first discuss some implementation details and then present the numerical results.

4.1. Implementation details

(*1*) *Choosing the weighting coefficients* $w_{\{j_1,j_2\}}$. As demonstrated in [11], the results of CC depend critically on the choice of the weighting coefficients $w_{\{j_1,j_2\}}$, and empirically, the best choice was based on the k-nearest neighbor method. In our implementations, we also followed the k-nearest neighbor method to determine the weighting coefficients $w_{\{j_1,j_2\}}$. More specifically, we chose the

weighting coefficient $w_{\{j_1,j_2\}}$ between the data points $\mathbf{x}_{j_1}$ and $\mathbf{x}_{j_2}$ as:

$$w_{\{j_1,j_2\}} = \iota^k_{\{j_1,j_2\}} \exp\left[-\alpha \|\mathbf{x}_{j_1} - \mathbf{x}_{j_2}\|_2^2\right],$$

where $\iota^k_{\{j_1,j_2\}}$ is 1 if both $\mathbf{x}_{j_1}$ and $\mathbf{x}_{j_2}$ are among the kth nearest neighbors (under the Euclidean distance metric) of each other and 0 otherwise, α is a nonnegative real constant, and k is a natural number. Note that setting $\alpha = 0$ gives uniform weights between the data points among the k-nearest neighbors of each other. In our implementations, however, we tuned α as a small positive number to improve the clustering accuracy. Following [11], we chose the value of k in our numerical experiments as the expected average cluster size.

(*2*) *Choosing the tuning parameters* γ, β, *and* ν. Note from (8) that if we set the tuning parameter $\gamma = 0$, this will lead to the trivial solution $\mathbf{u}_j = \mathbf{x}_j$ for each $j = 1, 2, \ldots, N$, i.e., the partition of the data points into singletons. On the other hand, if we set γ to be sufficiently large, this will lump all the data points into a single cluster. Varying γ in between gives rise to the entire clustering path. In our numerical experiments, we chose γ (and the second tuning parameter β in case of RCC) so that the results match the expected number of clusters. Even though mathematically there seems to be no guarantee that this is always possible, we were able to achieve the exact matches in all of our numerical experiments. The convergence of the proposed ADMM algorithm does not appear to be sensitive to the choice of the tuning parameter ν.

(*3*) *Measuring the clustering accuracy.* We measure the accuracy of the clustering results based on the accuracy of the adjacency matrix. For each set of testing data, the ground truth is known and from that we can construct an N-by-N binary ground truth adjacency matrix A with entries $A_{i,j} = 1$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same cluster and 0 otherwise. For a given output of Algorithm 1, we looked at the columns of the matrix $\mathbf{V}$, i.e., the difference variables $\mathbf{v}_\ell$. If $\mathbf{v}_\ell = \mathbf{0}$ (or close to $\mathbf{0}$ within the numerical accuracy), we set $\tilde{A}_{\phi_1(\ell),\phi_2(\ell)} = \tilde{A}_{\phi_2(\ell),\phi_1(\ell)} = 1$; otherwise, we set $\tilde{A}_{\phi_1(\ell),\phi_2(\ell)} = \tilde{A}_{\phi_2(\ell),\phi_1(\ell)} = 0$. The clustering accuracy was then calculated by counting the number of matching values in the upper triangles (excluding the diagonal entries) of A and $\tilde{A}$, normalized by the total number of adjacency pairs $N(N-1)/2$. This is known as the *Rand* index in the literature [36].

4.2. Synthetic data

The synthetic data that we considered were generated based on the standard Gaussian mixture model (GMM). We first generated three classes of data in $\mathbb{R}^3$, with 100 data points in each class. All data points were generated using the same variance but different mean for each class. Fig. 1 illustrates an example of the simulated GMM data. Then, outlier feature values were added to each of the data points, making each data point $\mathbf{x}_i$ a high-dimensional vector. Each outlier feature value was generated independently using an *identical* distribution across all 300 data points. We used different distributions of high variance for different outlier features.

Fig. 2 compares the clustering accuracies of CC, RCC, and CCML under different numbers of outlier features (from 0 to 7). (The standard deviations between CC, RCC, and CCML appear to be rather comparable for each set of experiments.) Each data point is calculated based on the average of 50 experiments, and for each experiment the tuning parameters are tuned such that the number of clusters matches that of the ground truth and the achieved clustering accuracy is highest possible. As illustrated, for CC the clustering accuracy decreases *rapidly* with the number of outlier features. For RCC, the clustering accuracy also decreases with the number of outlier features but much less rapidly than CC, and the decrease stops when the number of outlier features reaches 5. By comparison, CCML appears to be very robust to outlier features and provides significantly higher clustering accuracies over CC and RCC under *all* configurations.

Table 1
The clustering accuracies of the k-means, normalized-cut, CC, RCC, and CCML algorithms for three real-world data sets "seed", "wine", and "image".

Clustering accuracy (%)					
	k-means	Normalized-cut	CC	RCC	CCML
"Seed"	72.5	73.3	72.0	75.4	75.6
"Wine"	70.0	70.1	65.4	67.2	71.5
"Image"	73.4	72.0	80.5	83.0	82.2

Fig. 3 illustrates the clustering accuracy and the minimum value of the optimization problem (again averaged over 50 experiments) as a function of the number of iterations for both CCML and RCC. For both plots, the number of outlier features is set as 7 (so the total dimension of the data is 10). As illustrated, both algorithms exhibit very similar convergence behaviors and both appear to converge within in a few iterations. Fig. 4 illustrates the intensity map and the singular values of the Mahalanobis distance metric $\mathbf{B}$ learned from the final iteration for a particular experiment. As illustrated, the Mahalanobis distance metric $\mathbf{B}$ learned from the final iteration can successfully identify the three highly relevant features (the first three features) of the data.

4.3. Real-world data

We also tested the performance of the proposed CCML algorithm against the CC [11], RCC [16], and more traditional k-means [6] and normalized-cut [5] algorithms using the real-world data sets "seeds", "wine", and "image" from the UCI machine learning repository [30]:

- The "seeds" data set contains the measurements of geometrical properties of seeds belonging to three different types of wheat. There are 70 samples for each of the three classes. The three classes of wheat are Kama, Rosa, and Canadian. A soft X-ray technique was used to image the seed samples, and seven real-valued features were extracted from the X-ray images.
- The "wine" data set contains the results of a chemical analysis of wines grown in the same region of Italy, but derived from three different cultivars. There are 59, 71, and 48 samples in each of the three cultivars, respectively. Wines grown in the same cultivar are considered to be similar to each other. There are 13 features in this data.
- The "image" data set contains images of seven different classes of images, each with a different subject. The subjects are brickface, sky, foliage, cement, window, path, and grass. There are 330 data points in each of the seven classes. Each image was hand-segmented into 3-by-3 regions, from which 19 features were extracted.

The exact choices of the features of the above data sets can be found in [30].

Table 1 lists the clustering accuracies of the k-means, normalized-cut, CC, RCC, and CCML algorithms for the three real-world data sets mentioned above. As illustrated, the proposed CCML performs consistently among the best, if not the best, among the algorithms considered. In particular, we notice that while CCML and RCC perform rather similarly for the "Seed" and "Image" data sets, CCML outperforms significantly over RCC for the "Wine" data set. We postulate that this is mainly due to the fact that the number of outlier features is particularly large relative to the total number of features for the "Wine" data set.

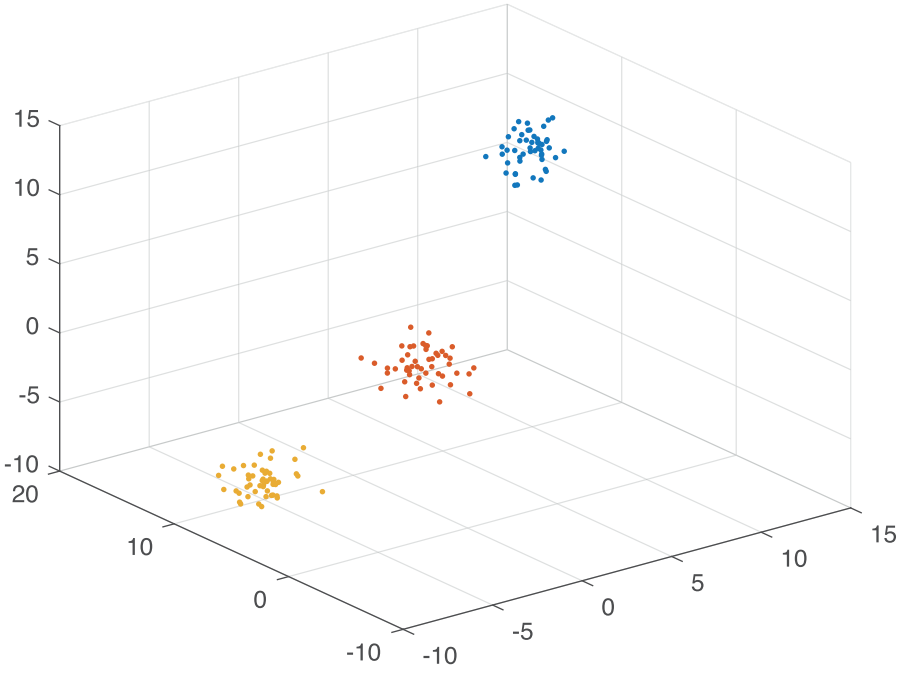

Fig. 1. An example of the simulated GMM data.

Fig. 5 illustrates the intensity map and the singular values of the Mahalanobis distance metric **B** learned from the final iteration for each of the data sets. From these plots, we can identify that: (1) or the "seeds" data, the third feature "Compactness" is clearly the most relevant one to this clustering; (2) for the "wine" data, the eighth feature "Nonflavnoids phenols" is the most relevant one to this clustering, and the third and the eleventh features "Ash" and "Hue" are also highly relevant to this clustering; (3) for the "image" data, the third, tenth, eleventh, twelfth, and thirteenth features "Region-pixel-count", "Intensity-mean", "Raw-red-mean", "Raw-blue-mean", and "Raw-green-mean" are the most relevant ones to this clustering.

5. Concluding remarks

CC [11] is a recently proposed clustering formulation that aims at minimizing the aggregate Euclidean distance between the data points and their corresponding cluster centers while leveraging group sparsity to the clustering solution through ℓ_1 penalizations. Compared with the traditional approach such as the k-means and the normalized-cut algorithms, the CC formulation can be efficiently and precisely solved using the ADMM [13–15]. However, the Euclidean metric treats each feature of the data equally. As a result, the performance of the CC algorithm deteriorates significantly in the presence of outlier features.

To address this issue, this paper considered a new formulation that combines CC and ML. It was shown that: (1) for any given positive definite Mahalanobis distance metric, the problem of CC

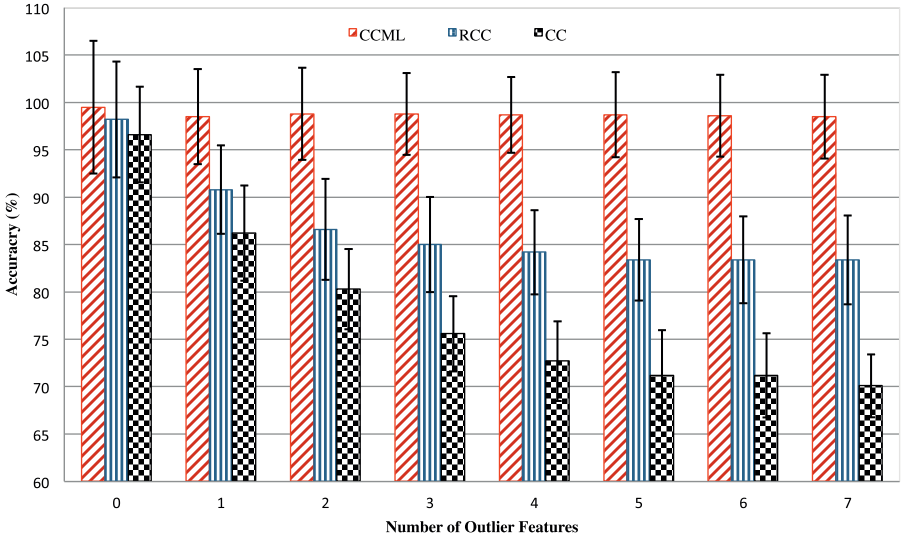

Fig. 2. Gaussian GMM data: clustering accuracy as a function of the number of outlier features. The narrow line on top of each bar indicates the standard deviation for each set of experiments.

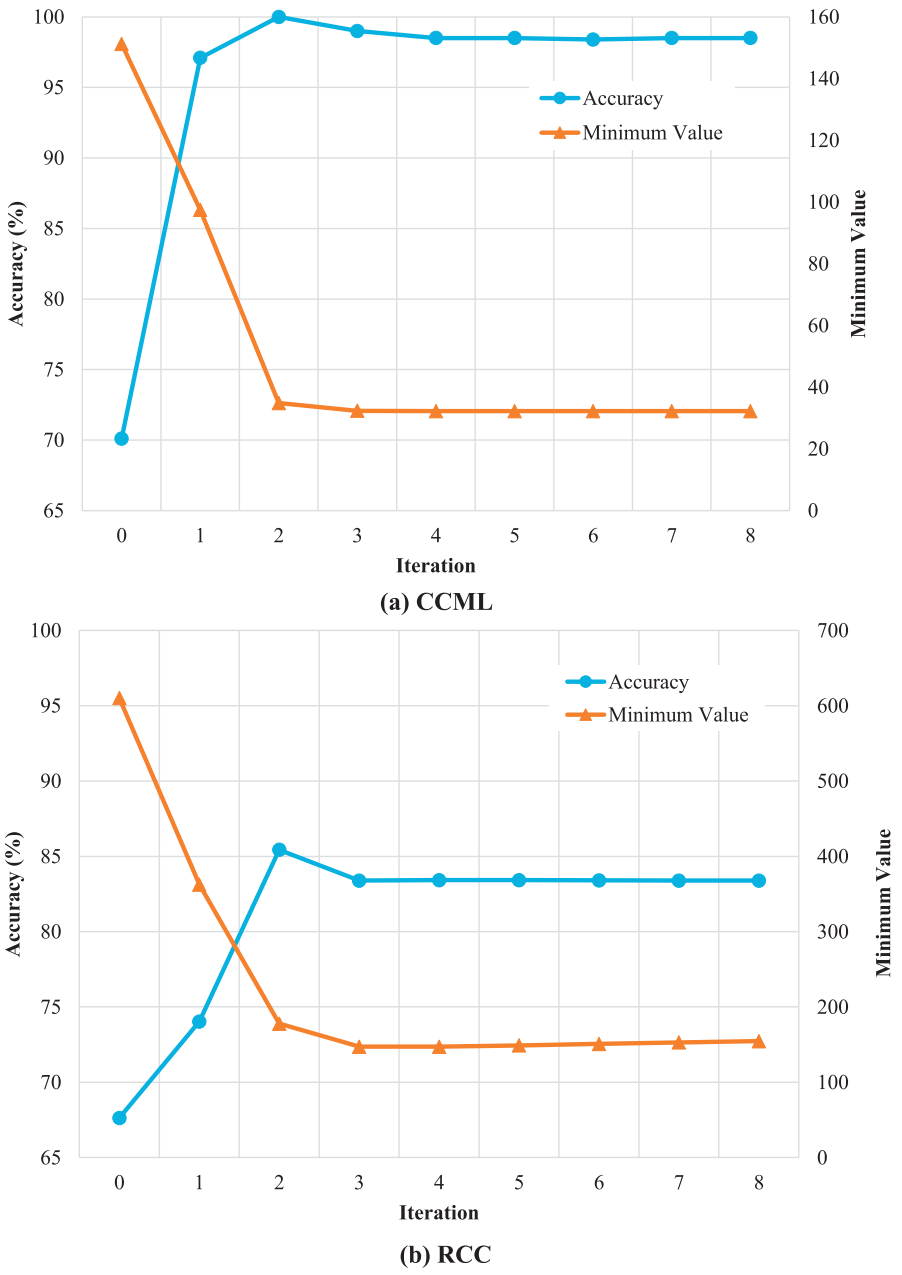

Fig. 3. Gaussian GMM data: convergence of the clustering accuracy and the minimum value of the optimization problem.

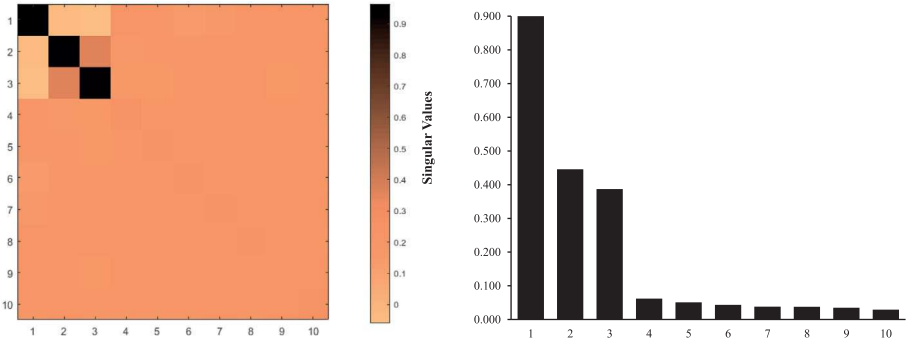

Fig. 4. Gaussian GMM data: intensity map and the singular values of the Mahalanobis distance metric **B** learned from the final iteration.

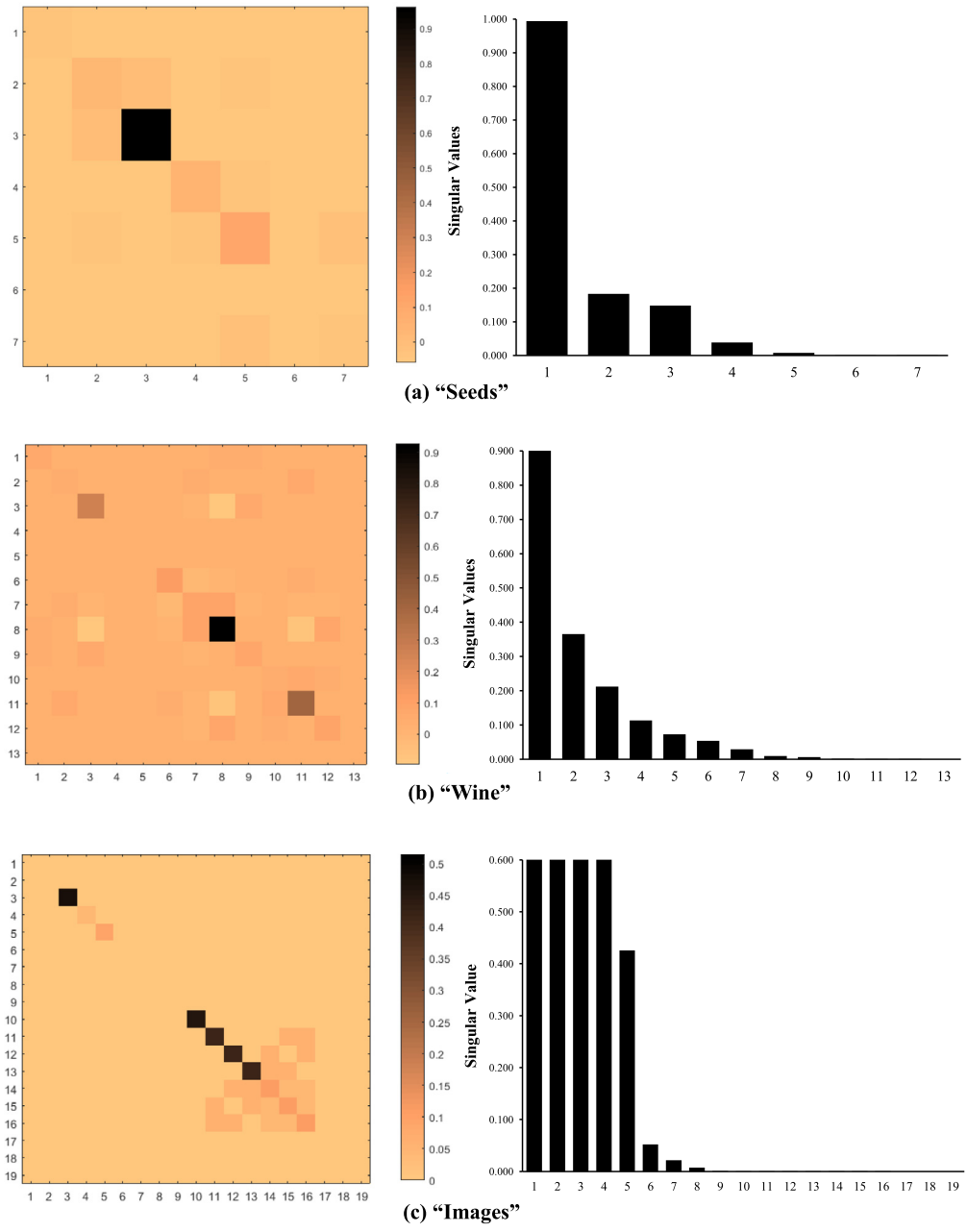

Fig. 5. Real-world data sets: Intensity map and the singular values of the Mahalanobis distance metric **B** learned from the final iteration.

can be precisely and efficiently solved within the ADMM framework; (2) when considering the family of positive definite Mahalanobis distances for convex clustering, the problem of ML admits a closed-form solution; (3) an algorithm that alternates between CC and ML can provide a significant performance boost over not only the original CC formulation of Chi and Lange [11] but also the recently proposed RCC formulation of Wang et al. [16].

Note that in our algorithm, ML is performed based on the *noisy* labels produced by the CC from the previous iteration. Our numerical experiments show that there is indeed a performance gap between learning from noisy labels and learning from the ground truth. We are currently working to bridge this gap by imposing additional structural constraints on the Mahalanobis distance metric as well as considering additional “cleanup” procedures for the intermediate clustering results.

Acknowledgments

This work was supported by the National Science Foundation (NSF) Grants 1719017, 1447235, 1547557 and 1553281.

References

[1] M. Dettling, P. Bühlmann, Supervised clustering of genes, Genome Biol. 3 (12) (2002) 0069.1–0069.15, doi:10.1186/gb-2002-3-12-research0069.
[2] X. Shen, F. Tokoglu, X. Papademetris, R. Constable, Groupwise whole-brain parcellation from resting-state fMRI data for network node identification, NeuroImage 82 (2013) 403–415, doi:10.1016/j.neuroimage.2013.05.081.
[3] F. Yin, C.-L. Liu, Handwritten chinese text line segmentation by clustering with distance metric learning, Pattern Recognit. 42 (12) (2009) 3146–3157, doi:10.1016/j.patcog.2008.12.013.
[4] J. Ye, Z. Zhao, H. Liu, Adaptive distance metric learning for clustering, in: Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–7. doi: 10.1109/CVPR.2007.383103.

[5] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.

[6] X. Jin, J. Han, K-Means Clustering, Springer, Boston, MA, US, pp. 563–564. doi: 10.1007/978-0-387-30164-8_425.

[7] T.D. Hocking, A. Joulin, F. Bach, J.-P. Vert, Clusterpath an algorithm for clustering using convex fusion penalties, in: Proceedings of the Twenty-Eighth International Conference on Machine Learning, United States, 2011, p. 1.

[8] R. Lajugie, F. Bach, S. Arlot, Large margin metric learning for constrained partitioning problems, in: Proceedings of the 2014 International Conference on Machine Learning, 2014.

[9] F.R. Bach, M.I. Jordan, Learning spectral clustering, in: Proceedings of the 2004 Advances in Neural Information Processing Systems, 2004, pp. 305–312.

[10] P. Zhu, W. Zhu, Q. Hu, C. Zhang, W. Zuo, Subspace clustering guided unsupervised feature selection, Pattern Recognit. 66 (Suppl C) (2017) 364–374, doi:10.1016/j.patcog.2017.01.016.

[11] E.C. Chi, K. Lange, Splitting methods for convex clustering, J. Comput. Graph. Stat. 24 (4) (2015) 994–1013.

[12] F. Lindsten, H. Ohlsson, L. Ljung, Just Relax and Come Clustering!: A Convexification of *k*-Means Clustering, Technical Report, Linkping University, The Institute of Technology, 2011.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2011) 1–122, doi:10.1561/2200000016.

[14] R. Glowinski, A. Marroco, Sur l'approximation, par lments finis d'ordre un, et la rsolution, par pnalisation-dualit d'une classe de problmes de dirichlet non linaires, ESAIM Math. Model. Numer. Anal. Modlisation Mathmatique Anal. Numrique 9 (R2) (1975) 41–76.

[15] D. Gabay, Chapter IX: applications of the method of multipliers to variational inequalities, Stud. Math. Appl. 15 (1983) 299–331.

[16] Q. Wang, P. Gong, S. Chang, T. Huang, J. Zhou, Robust convex clustering analysis, in: Proceedings of the Sixteenth IEEE International Conference on Data Mining, ICDM 2016, 2017, pp. 1263–1268. doi: 10.1109/ICDM.2016.123.

[17] F. Wang, J. Sun, Survey on distance metric learning and dimensionality reduction in data mining, Data Min Knowl. Discov. 29 (2) (2015) 534.

[18] K.Q. Weinberger, J. Blitzer, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, in: Proceedings of the Advances in Neural Information Processing Systems, 2006, pp. 1473–1480.

[19] S. Xiang, F. Nie, C. Zhang, Learning a Mahalanobis distance metric for data clustering and classification, Pattern Recognit. 41 (12) (2008) 3600–3612.

[20] X. Yin, S. Chen, E. Hu, D. Zhang, Semi-supervised clustering with metric learning: an adaptive Kernel method, Pattern Recognit. 43 (4) (2010) 1320–1333, doi:10.1016/j.patcog.2009.11.005.

[21] Y. Yu, J. Jiang, L. Zhang, Distance metric learning by minimal distance maximization, Pattern Recognit. 44 (3) (2011) 639–649, doi:10.1016/j.patcog.2010.09.019.

[22] C.-C. Chang, A boosting approach for supervised Mahalanobis distance metric learning, Pattern Recognit. 45 (2) (2012) 844–862, doi:10.1016/j.patcog.2011.07.026.

[23] B. Nguyen, C. Morell, B. De Baets, Supervised distance metric learning through maximization of the Jeffrey divergence, Pattern Recognit. 64 (2017) 215–225.

[24] Y. Ren, X. Li, X. Lu, Feedback mechanism based iterative metric learning for person re-identification, Pattern Recognit. (2017), doi:10.1016/j.patcog.2017.04.012.

[25] F. Wang, P. Li, A.C. König, M. Wan, Improving clustering by learning a bi-stochastic data similarity matrix, Knowl. Inf. Syst. 32 (2) (2012) 351–382, doi:10.1007/s10115-011-0433-1.

[26] X. Yang, L.J. Latecki, A. Gross, Distance learning based on convex clustering, in: G. Bebis, R. Boyle, B. Parvin, D. Koracin, Y. Kuno, J. Wang, R. Pajarola, P. Lindstrom, A. Hinkenjann, M.L. Encarnação, C.T. Silva, D. Coming (Eds.), Advances in Visual Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 747–756.

[27] H. Liu, M. Shao, S. Li, Y. Fu, Infinite ensemble for image clustering, in: Proceedings of the Twenty-Second ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 1745–1754.

[28] E.P. Xing, M.I. Jordan, S.J. Russell, A.Y. Ng, Distance metric learning with application to clustering with side-information, in: Proceedings of the Advances in Neural Information Processing Systems, 2003, pp. 521–528.

[29] Å. Björck, Numerical Methods in Matrix Computations, Springer, 2015.

[30] D. Dheeru, E. Karra Taniskidou, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences (2017) http://archive.ics.uci.edu/ml.

[31] S.C. Hoi, W. Liu, S.-F. Chang, Semi-supervised distance metric learning for collaborative image retrieval and clustering, ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) 6 (3) (2010) 18.

[32] A. Jameson, Solution of the equation $AX + XB = C$ by inversion of an M^*M or N^*N matrix, SIAM J. Appl. Math. 16 (5) (1968) 1020–1023.

[33] R.H. Bartels, G.W. Stewart, Solution of the matrix equation $AX + XB = C$, Commun. ACM 15 (9) (1972) 820–826.

[34] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[35] A.E. Bryson, Applied Optimal Control: Optimization, Estimation and Control, CRC Press, 1975.

[36] W. Rand, Objective criteria for the evaluation of clustering methods, J. Am. Stat. Assoc. 66 (336) (1971) 846–850, doi:10.1080/01621459.1971.10482356.

Xiaopeng Lucia Sui received her B.S. degree in Electrical Engineering from University of Alberta in 2012. She is currently a Ph.D. student in the Department of Electrical and Computer Engineering at Texas A&M University, College Station, TX, USA. Her research interests include machine learning and bioinformatics.

Li Xu received his Ph.D. degree from the Department of Mathematics, University of Hong Kong. Currently, he is an Associate Professor at the Institute of Computing Technology, Chinese Academy of Sciences. His research areas are Statistical Learning and Information Theory.

Xiaoning Qian received the Ph.D. degree in Electrical Engineering from Yale University in 2005. Currently, he is an assistant professor with the Department of Electrical & Computer Engineering, Texas A&M University. His research interests include computational network biology, genomic signal processing, and biomedical signal and image analysis.

Tie Liu is a Professor in the Department of Electrical and Computer Engineering at Texas A&M University. He received his Ph.D. degree in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 2006. His research interest is in the field of information theory and statistical information processing.