Convex Discriminative Multitask Clustering

Xiao-Lei Zhang, *Member, IEEE*

Abstract—Multitask clustering tries to improve the clustering performance of multiple tasks simultaneously by taking their relationship into account. Most existing multitask clustering algorithms fall into the type of generative clustering, and none are formulated as convex optimization problems. In this paper, we propose two convex Discriminative Multitask Clustering (DMTC) objectives to address the problems. The first one aims to learn a shared feature representation, which can be seen as a technical combination of the convex multitask feature learning and the convex Multiclass Maximum Margin Clustering (M3C). The second one aims to learn the task relationship, which can be seen as a combination of the convex multitask relationship learning and M3C. The objectives of the two algorithms are solved in a uniform procedure by the efficient cutting-plane algorithm and further unified in the Bayesian framework. Experimental results on a toy problem and two benchmark data sets demonstrate the effectiveness of the proposed algorithms.

Index Terms—Convex optimization, cutting-plane algorithm, discriminative clustering, unsupervised multitask learning

1 Introduction

With the rapid development of information technology, massive amounts of unlabeled task-specific data are generated every day. Many tasks can be seen as self-contained, yet somewhat similar. Because labeling the data manually is time-consuming and expensive, we often resort to *clustering* algorithms for mining the undiscovered knowledge in the data.

In traditional data mining studies, we do clustering to each task independently. However, some tasks have so few data that the data distributions cannot be covered well. Hence, it is natural to think about clustering several unlabeled tasks together for improving the performance on each individual task. However, although some tasks are similar, there are still many tasks mutually unrelated, dissimilar, and even reverse. Simply merging all tasks together for clustering might be harmful. Therefore, it is urgent to develop a multitask clustering (MTC) algorithm that 1) not only is powerful in clustering each individual task 2) but also can mine the task relationships automatically from the data so as to further improve the clustering performance. For achieving our goal on MTC, we need to resort to two research areas—Multitask Learning (MTL) and clustering.

Multitask learning. MTL [1], also known as *learning to learn* [2], learns multiple (probably) related tasks simultaneously for improving the generalization performance on each task. It can be reviewed in three respects. They are 1) "what to learn", 2) "when to learn", and 3)"how to learn" [3].

"What to learn" asks what knowledge is shared across tasks [3]. In this respect, the MTL techniques can be categorized to two classes. The first class shares common feature or kernel representations, such as the hidden units of neural networks [1], [4], [5] and a common representation within the regularization framework [6], [7], [8], [9], [10], [11], [12]. The second class shares common model parameters, such as placing a common prior across tasks within the hierarchical Bayesian framework [13], [14], [15], [16], learning the differences of the task-specific models in Frobenius norms under the regularization framework [17], [18], [19], etc. Some of the methods are identical but appear in different mathematical forms, such as [8] and [19].

"When to learn" asks in which situation the tasks can share. Specifically, many MTL algorithms assume that the tasks are mutually related which is an ideal situation. In practice, there might be some outlier tasks or tasks with negative correlation. Learning with these tasks results in *negative transfer* or worsened performance. Hence, how to discover the task relationship is another key issue that is becoming more and more attractive [4], [19], [20], [21], [22]. One method is to group tasks into several clusters where the tasks in different groups are regarded as unrelated [4], [20], [21], [22]. Another method is to learn the inter-task covariance matrix of the multivariate Gaussian prior [19].

"How to learn" asks how the optimization problem can reach a good solution (i.e. performance) in a reasonable time when the first two respects are specified. In respect of effectiveness, among the aforementioned MTL methods, how to construct convex optimization objectives is a common thought in MTL since the global optimum solutions can be achieved and the optimization can be simplified. Until present, several convex MTL algorithms have been developed, and better performance was reported [8], [11], [12], [19], [20]. In respect of efficiency, the alternating optimization method that optimizes in turn one parameter with others fixed is a common efficient method.

Summarizing the aforementioned, in the new MTC design, we take the convexity and the task relationship mining as two important considerations.

Clustering. Clustering is the process of partitioning a set of data observations into multiple clusters so that the observations within a cluster are similar, and the observations in different clusters are very dissimilar [23]. Since the early works on k-means, many clustering algorithms have

• *The author is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. E-mail: huoshan6@126.com.. E-mail: huoshan6@126.com.*

Manuscript received 8 Mar. 2013; revised 22 Mar. 2014; accepted 3 July 2014. Date of publication 25 July 2014; date of current version 5 Dec. 2014.
Recommended for acceptance by G. Lanckriet.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2014.2343221

been developed, such as kernel k-means, spectral clustering [24], [25], hierarchical clustering, probabilistic-based clustering, metric clustering, clustering nonnumerical data, clustering high dimensional data, clustering graph data, etc.

Like supervised classification, clustering algorithms can be classified to two classes—*generative clustering* and *discriminative clustering*. The generative clustering algorithms model $p(\mathbf{x}, y; \theta)$ where $\mathbf{x}$ and y denotes the input and output of the learning system respectively and θ is the parameter. The discriminative clustering algorithms only focus on modeling $p(y \mid \mathbf{x}; \theta)$. Many traditional clustering algorithms fall into the class of the generative clustering, such as k-means, Gaussian mixture model, restricted Boltzman machine, etc. However, when we only care about the predicted labels but not the distribution of the observations, the generative clustering methods seem solving a more general problem than what we want. Moreover, if we make a wrong model assumption on the underlying data distribution, we may get a rather weak clustering result. This phenomenon has been observed in both the supervised classification [26] and the clustering [27]. Due to the above problems, many discriminative clustering methods have been developed [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], such as spectral clustering [25], Maximum Margin Clustering (MMC) [29], [30], [31], [32], [33], [34], regularized information maximization [35], etc.

Summarizing the aforementioned, in the new MTC design, we should try to construct a discriminative MTC (DMTC) clustering algorithm but not a generative one.

Multitask clustering. Although the supervised MTL has been studied extensively in the aforementioned respects, the unsupervised MTL, i.e. MTC [37], seems far from explored yet. Only very recently, it received more and more attention [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47]. 1) In respect of "what to learn", in [37], Teh et al. proposed to discover the clusters that can be shared via the hierarchical Dirichlet process. In [48], Kulis and Jordan first revisited a regularized k-means algorithm in the view of the Dirichlet process and then extended it to MTC by sharing the clusters of the observations across the tasks. In [38], Dai et al. extended the information theoretic co-clustering algorithm to MTC by making the tasks share the same feature attribute cluster, where they studied MTC in the *transfer learning* scenario, a special case of MTL that focuses on the performance of one target task. In [39], [40], [41], [42], [43], [44], [45], the authors tried to learn a shared feature or kernel representations in different distance metrics, such as Bregman distance. 2) In respect of "when to learn", in [40], [41], Zhang and Zhang proposed the pairwise task regularization and centralized task regularization methods for discovering the task relationship. 3) However, in respect of "how to learn", none of the MTC algorithms can hold the convexity.

Moreover, most of the MTC algorithms belong to the class of the generative clustering. To our best knowledge, the discriminative MTC seems lack of full study. Only in [42], [46], the authors proposed the spectral clustering based MTCs.

Contributions. In this paper, we propose two objectives of discriminative MTC, which are formulated as difficult mixed integer programming (MIP) problems. We relaxed the MIP problems to two convex optimization problems. The first one, named convex discriminative multitask feature clustering (DMTFC), can be seen as a technical combination of the convex supervised multitask feature learning (MTFL) [8] and the support vector regression based Multiclass MMC (SVR-M3C) [34]. The second one, named convex discriminative multitask relationship clustering (DMTRC), can be seen as a technical combination of the convex multitask relationship learning (MTRL) [19] and SVR-M3C. These combinations are quite natural and yield the following advantages:

1) In respect of "what to learn", DMTFC learns a shared feature representation between tasks. DMTRC minimizes the model differences of the related tasks. Both algorithms, as discriminative clustering algorithms, try to find the optimal label pattern directly. Both of them work in Frobenius norms under the regularization framework.
2) In respect of "when to learn", DMTRC can learn the task relationship automatically from the data by learning the inter-task covariance matrix.
3) In respect of "how to learn", both algorithms are formulated as convex optimization problems, and are solved in a uniform optimization procedure. A number of efficient SVM techniques are available for the problems. In this paper, we employ the cutting-plane algorithm (CPA) [49], [50], [51] that has achieved a great success in SVM to solve the DMTCs efficiently.

Besides, we further unify the two objectives together in the Bayesian framework. Experimental comparison with seven clustering algorithms and three state-of-the-art MTCs on the pendigits toy data set, the multi-domain newsgroups data set, and the multi-domain sentiment data set demonstrates the effectiveness of DMTC.

The remainder of the paper is organized as follows. In Section 2, we briefly review related techniques. In Sections 3, 4, and 5, we present the covex DMTFC and DMTRC objectives respectively. In Section 6, we solve DMTFC and DMTRC in a uniform optimization procedure. In Section 7, we extend DMTC to nonlinear kernels. In Section 8, we analyze the complexity theoretically. In Section 9, we view DMTC in the Baysian framework. In Section 10, we show the effectiveness of DMTC empirically. Finally, in Section 11, we conclude this paper and present some future work. The detailed derivation and experimental results are in the supplementary material, available at http://sites.google.com/site/zhangxiaolei321/

We first introduce some notations here. Bold small letters, e.g., $\mathbf{w}$ and α, indicate column vectors. Bold capital letters, e.g., $\mathbf{W}$, $\mathbf{K}$, indicate matrices. Letters in calligraphic bold fonts, e.g., $\mathcal{A}$, $\mathcal{B}$, and $\mathbb{R}$, indicate sets, where $\mathbb{R}^d$ denotes a d-dimensional real space. $\mathbf{0}_m$ ($\mathbf{1}_m$) is a vector with all m entries being 1 (0). $\mathbf{I}_d$ is a $d \times d$ identity matrix. The operator ^{T} denotes the transpose. The $\langle \mathbf{x}, \mathbf{y} \rangle$ defines the inner product of $\mathbf{x}$ and $\mathbf{y}$. The operator $\| \cdot \|^m$ denotes the m-norm, where m is a constant. The operator "$\mathrm{tr}(\cdot)$" denotes the trace of matrix. The abbreviation "s.t." is short for "subject to". $h(\boldsymbol{\alpha}; \boldsymbol{\beta})$ denotes a function h with parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The symbol $\{\mathbf{W}_c\}_{c=1}^C$ is short for the set $\{\mathbf{W}_1, \dots, \mathbf{W}_C\}$. Without confusion, we may further write $\{\mathbf{W}_c\}_{c=1}^C$ as $\{\mathbf{W}_c\}_c$ in equations for simplicity.

2 Related Work

Convex multitask learning. We introduce some related convex MTL [8], [11], [12], [19], [20], [21] as follows.

In [20], Jacob et al. proposed to learn the task relationship by clustering the similar tasks into the same group. Because the embedded clustering problem is non-convex, they relaxed the problem to a convex one. In [21], Zhou et al. proved that the alternating structure optimization (ASO) [6] and the clustered MTL (CMTL) [20] are equivalent except that ASO operates on the feature dimension of the multitask model but CMTL operates on the task dimension of the model. Observing the equivalence, in [11], [12], Chen et al. proposed a convex ASO that learns a shared feature subspace.

In [8], Argyriou et al. proposed to minimize the empirical risk of all tasks with a Frobenius norm penalty on the differences of the task-specific models, which is a non-convex optimization problem. Then, they proved that the problem is equivalent to a convex optimization problem—multitask feature learning. In [19], Zhang and Yeung first tried to learn the task covariance matrix of the multivariate Gaussian prior in the regularization framework. Because the concave function with respect to the covariance matrix variable makes the objective non-convex, they further replaced the concave function by two convex constraints, which results in a convex optimization problem, named MTRL. Although MTFL and MTRL are derived in different ways, they are identical. Moreover, they can be explained together in the Bayesian framework.

To prevent misleading, here, we have to emphasize that convex formulations do not mean absolutely better performance over non-convex ones. How to find good local minima in the non-convex formulations seems not a well explored field in MTL, but is emerging in the study of the regularization frameworks, such as [52] and the references therein.

Convex maximum margin clustering. Among the numbers of discriminative clustering algorithms, MMC [29], [30], [31], [32], [33], [34], which is an unsupervised extension of support vector machine (SVM), has received much attention since year 2005. The key idea of MMC is to find not only the maximum margin hyperplane in the feature space but also the optimal label pattern, such that if an SVM trained on the optimal label pattern, the optimal label pattern will yield the largest margin among all possible label patterns $\{\mathbf{y} \,|\, \mathbf{y} = \{y_j\}_{j=1}^n, \forall y_j\}$, where n is the number of observations and y_j denotes the possible class of the jth observation. The main difficulty of MMC lies in that it is originally formulated as a difficult mixed-integer programming problem [29] due to the integer vector variable $\mathbf{y}$ in the objective of MMC.

To overcome MIP, researchers either relaxed the objective as convex optimization problems [29], [30], [33], [34] or reformulated it to non-convex ones [31], [32]. Because the convex relaxation methods achieve better clustering results than non-convex ones in general, we pay particular attention to this kind.

Originally, in [29], Xu et al. proposed to reformulate MMC as a convex semi-definite programming problem by relaxing $\mathbf{M} = \mathbf{y}\mathbf{y}^T$ to a continuous matrix. In [30], they further extended the binary-class MMC to the multiclass scenario which has a time complexity as high as $\mathcal{O}(n^{6.5})$. Recently, in [34], Zhang and Wu proposed to construct a convex hull [53] on $\{\mathbf{y}\}$, and further extended the binary-class algorithm to the multiclass problem, i.e. SVR-M3C, which can be solved in an alternating method in time $\mathcal{O}(n \log n)$.

We found that SVR-M3C and MTFL/MTRL can be combined quite naturally within DMTC, and a number of popular SVM techniques are available for solving the problem efficiently. MMC contributes to the implementation of the proposed DMTC.

Cluster ensemble. The most similar work with MTC in machine learning and data mining is *cluster ensemble* [54], [55], [56], [57], [58], [59], [60], [61], [62]. The cluster ensemble aims to combine multiple clusterings with a so-called *consensus function* for enhancing the stability and accuracy of the base clusterings. The scenario that each base clustering processes only a part of the observations is called the *observation-distributed scenario* [54], [57] or *crowdclustering* [59], [61]. The main difference between MTC and the crowdclustering is that the crowdclustering assumes that all parts of observations are sampled from the same underlying distribution while MTC does not assume so. But, we have to note that several cluster ensemble techniques can be adapted to MTC, such as [57], [59], [61], [62]. Still, to our knowledge, none of the cluster ensembles can both hold convexity and be constructed on discriminative clusterings.

3 Problem Formulation

Suppose there are m clustering tasks. The ith task consists of n_i unlabeled observations $\{\mathbf{x}_j^i\}_{j=1}^{n_i}$, $\mathbf{x}_j^i \in \mathbb{R}^d$. We cluster each task to the same number of classes, denoted as C with $C \geq 2$. The prediction function of the cth class for the ith task is defined as $f_c^i(\mathbf{x}^i) = \mathbf{w}_{i,c}^T \mathbf{x}^i$, where $\mathbf{w}_{i,c}$ is the parameter of f_c^i, and where we have omitted the bias term $b_{i,c}$ in f_c^i for simplicity. The observation $\mathbf{x}^i$ is assigned to the $c^\star$th class, if $c^\star = \arg\max_c f^i_{\mathbf{w}_{i,c}}$ holds. Note that the reason why we assume all tasks have the same number of classes is clarified as follows. 1) In practice, the related tasks tend to share a similar structure. 2) We can easily extend this assumption to the scenario that the tasks have different number of classes by extending the Frobenius norm in Eq. (1) (or in Eq. (13)) from the one-class-versus-one-class correlation to one-class-versus-all-classes correlation. For clarity, we use a more strict assumption.

For a C class clustering problem with the label $y \in \{1, 2, \ldots, C\}$, we extend y to a C dimensional *row* indicator vector $\bar{\mathbf{y}}$, i.e. $\bar{\mathbf{y}} = [\bar{y}_1, \ldots, \bar{y}_C]$, where the label vector $\bar{\mathbf{y}}$ takes 1 for the kth element and $-\frac{1}{C-1}$ for the others when $y = k$. For instance, if $\mathbf{x}$ falls into the first class, then $\bar{\mathbf{y}} = [1, -\frac{1}{C-1}, \ldots, -\frac{1}{C-1}]$. This coding method is a common strategy in the multiclass problems, such as k-means. Here, a set $\mathcal{B}_{\bar{\mathbf{y}}}$ is defined for all possible $\bar{\mathbf{y}}$, i.e. $\mathcal{B}_{\bar{\mathbf{y}}} = \{[1, -\frac{1}{C-1}, \ldots, -\frac{1}{C-1}], [-\frac{1}{C-1}, 1, \ldots, -\frac{1}{C-1}], \ldots, [-\frac{1}{C-1}, -\frac{1}{C-1},$

$\ldots,\bigstar 1]\}$. For a m-task MTC problem, we denote $\mathbf{W}_c=[\mathbf{w}_{1,c},\ldots,\mathbf{w}_{m,c}]$, $\mathbf{X}^i=[\mathbf{x}^i_1,\ldots,\mathbf{x}^i_{n_i}]$, and $\mathbf{Y}^i=[(\overline{\mathbf{y}}^i_1)^T,\ldots,(\overline{\mathbf{y}}^i_{n_i})^T]^T$.

4 Convex Discriminative Multitask Feature Clustering

In this section, we will introduce the convex objective of the proposed DMTFC.

We extend the MTFL algorithm [8] to its multiclass unsupervised counterpart, the objective of DMTFC, which is formulated as the following MIP problem:

$$\begin{aligned}\min_{\{\mathbf{Y}^i\in\mathcal{B}^i\}_{i=1}^m}\min_{\{\mathbf{W}_c\}_{c=1}^C}\min_{\mathbf{D}\in\mathcal{D}}\sum_{c=1}^C&\left(\frac{\lambda_2}{2}\mathrm{tr}\big(\mathbf{W}_c^T\mathbf{D}^{-1}\mathbf{W}_c\big)\right.\\ &\left.+\frac{\lambda_1}{2}\mathrm{tr}\big(\mathbf{W}_c^T\mathbf{W}_c\big)+\sum_{i=1}^m\frac{1}{n_i}\sum_{j=1}^{n_i}\big(\overline{y}^i_{j,c}-\mathbf{w}_{i,c}^T\mathbf{x}^i_j\big)^2\right),\end{aligned}\tag{1}$$

where λ_1 and λ_2 are two tunable hyper parameters, the matrix variable $\mathbf{D}$ represents a covariance matrix that models the relationships between the features, the convex constraint set $\mathcal{D}$ constrains $\mathbf{D}$ to be a valid covariance matrix which is defined as

$$\mathcal{D}=\{\mathbf{D}\,|\,\mathbf{D}\in\mathbb{R}^{d\times d},\mathbf{D}\succeq 0,\mathrm{tr}(\mathbf{D})=1\},\tag{2}$$

and $\mathcal{B}^i$ is defined as:

$$\mathcal{B}^i\triangleq\left\{\mathbf{Y}^i\left|\left\{\begin{array}{l}-\frac{l_{i,c}}{C-1}\le\frac{\mathbf{1}_{n_i}^T\overline{\overline{\mathbf{y}}}^i_c}{n_i}\le l_{i,c},\forall c=1,\ldots,C,\\ \overline{\mathbf{y}}^i_j\in\mathcal{B}_{\overline{\mathbf{y}}},\forall j=1,\ldots,n_i.\end{array}\right.\right.\right\},\tag{3}$$

where $\overline{\overline{\mathbf{y}}}^i_c=[\overline{y}^i_{1,c},\ldots,\overline{y}^i_{n_i,c}]^T$ represents the cth column of $\mathbf{Y}^i$ and $\{\{l_{i,c}\}_{c=1}^C\}_{i=1}^m$ are user defined parameters that control the class balance. The constraint $-\frac{l_{i,c}}{C-1}\le\frac{\mathbf{1}_{n_i}^T\overline{\overline{\mathbf{y}}}^i_c}{n_i}\le l_{i,c}$ specifies the class evenness of the cth class, while the constraint $\overline{\mathbf{y}}^i_j\in\mathcal{B}_{\overline{\mathbf{y}}}$ commands that $\mathbf{Y}^i$ must be a legal indicator matrix. As will be shown in the experimental section, a correct class balance assumption is very important to the success of DMTC. It not only can help DMTC detect a reasonable label pattern but also can prevent the interference of outliers. If we know the class distribution, we can set $l_{l,c}$ to a value around $\mathbf{1}_{n_i}^T{\mathbf{y}^\star}^i_c/n_i$ where ${\mathbf{y}^\star}^i_c$ is the cth column of the ground truth label matrix of the ith task, otherwise, we just set all $l_{l,c}$ to the same empirical value.

Problem (1) is quite similar with [8, Theorem 1] except that Problem (1) is a regularized multiclass problem and label $\mathbf{Y}^i$ is an integer matrix variable.

To void MIP, we construct a convex hull [53] on $\mathcal{B}^i$ as in [33], [34]. Specifically, fixing $\{\mathbf{Y}^i\}_{i=1}^m$ and $\mathbf{D}$, Problem (1) is formulated as:

$$\begin{aligned}\sum_{c=1}^C&\left(\min_{\mathbf{W}_c}\frac{\lambda_1}{2}\mathrm{tr}\big(\mathbf{W}_c^T\mathbf{W}_c\big)+\frac{\lambda_2}{2}\mathrm{tr}\big(\mathbf{W}_c^T\mathbf{D}^{-1}\mathbf{W}_c\big)\right.\\ &\left.+\sum_{i=1}^m\frac{1}{n_i}\sum_{j=1}^{n_i}\big(\overline{y}^i_{j,c}-\mathbf{w}_{i,c}^T\mathbf{x}^i_j\big)^2\right),\end{aligned}\tag{4}$$

where the problems in the big brackets are mutually independent. We rewrite the problem in the big brackets in the constrained form as follows:

$$\begin{aligned}\min_{\mathbf{W}_i}\ &\frac{\lambda_1}{2}\mathrm{tr}\big(\mathbf{W}_c^T\mathbf{W}_c\big)+\frac{\lambda_2}{2}\mathrm{tr}\big(\mathbf{W}_c^T\mathbf{D}^{-1}\mathbf{W}_c\big)\\ &+\sum_{i=1}^m\frac{1}{n_i}\sum_{j=1}^{n_i}\big(\xi^i_{j,c}\big)^2\\ \text{s.t.}\ \ &\overline{y}^i_{j,c}-\mathbf{w}_{i,c}^T\mathbf{x}^i_j=\xi^i_{j,c},\forall i=1,\ldots,m,\forall j=1,\ldots,n_i,\end{aligned}\tag{5}$$

which is a regularized minimization of the squared-loss. According to the Karush-Kuhn-Tucker conditions, the dual form of Problem (5) is written as:

$$\max_{\boldsymbol{\alpha}_c}\sum_{j=1}^{n_i}\alpha^i_{j,c}\overline{y}^i_{j,c}-\frac{1}{2}{\boldsymbol{\alpha}_c}^T\widetilde{\mathbf{K}}_{\mathrm{F}}\boldsymbol{\alpha}_c,\tag{6}$$

where $\alpha_c=[\alpha^1_{1,c},\ldots,\alpha^m_{n_m,c}]^T$ are the dual variables, $\widetilde{\mathbf{K}}_{\mathrm{F}}=\mathbf{K}_{\mathrm{F}}+\frac{1}{2}\boldsymbol{\Lambda}$ with $\boldsymbol{\Lambda}$ as the diagonal matrix whose diagonal element equals to n_i if the corresponding observation belongs to the ith task, and $\mathbf{K}_{\mathrm{F}}$ denoted as the multitask-kernel matrix for feature learning which is defined as:

$$K_{\mathrm{F}}\big(\mathbf{x}^{i_1}_{j_1},\mathbf{x}^{i_2}_{j_2}\big)={\mathbf{x}^{i_1}_{j_1}}^T\mathbf{D}(\lambda_1\mathbf{D}+\lambda_2\mathbf{I}_d)^{-1}\mathbf{x}^{i_2}_{j_2}\langle\mathbf{e}_{i_1},\mathbf{e}_{i_2}\rangle.\tag{7}$$

$\mathbf{W}_c$ is obtained as:

$$\mathbf{W}_c=\sum_{i=1}^m\sum_{j=1}^{n_i}\alpha^i_{j,c}\mathbf{D}(\lambda_1\mathbf{D}+\lambda_2\mathbf{I}_d)^{-1}\mathbf{x}^i_j\mathbf{e}_i^T,\tag{8}$$

where $\mathbf{e}_i$ represents the ith column of the identity matrix. The detailed derivation of Eq. (6) is in the supplementary material, available online.

Substituting (6) back to Problem (4) and then substituting (4) back to problem (1) can get an equivalent optimization Problem of (4) as follows:

$$\min_{\{\mathbf{Y}^i\in\mathcal{B}^i\}_{i=1}^m}\min_{\mathbf{D}\in\mathcal{D}}\max_{\{\boldsymbol{\alpha}_c\}_{c=1}^C}\sum_{i,c,j}\alpha^i_{j,c}\overline{y}^i_{j,c}-\frac{1}{2}\sum_{c=1}^C\boldsymbol{\alpha}_c^T\widetilde{\mathbf{K}}_{\mathrm{F}}\boldsymbol{\alpha}_c.\tag{9}$$

Because the second term of Problem (9) is irrelevant to the integer matrix variable $\mathbf{Y}^i$, it is easy to see that the following problem learns a lower bound of Problem (9):

$$\begin{aligned}\min_{\mathbf{D}\in\mathcal{D}}\max_{\{\boldsymbol{\alpha}_c\}_{c=1}^C}&\left\{\max_{\{\theta_i\}_{i=1}^m}\sum_{i=1}^m\theta_i-\frac{1}{2}\sum_{c=1}^C\boldsymbol{\alpha}_c^T\widetilde{\mathbf{K}}_{\mathrm{F}}\boldsymbol{\alpha}_c\right.\\ \text{s.t.}\quad&\left.\theta_i\le\sum_{c=1}^C\sum_{j=1}^{n_i}\alpha^i_{j,c}\overline{y}^i_{j,c},\forall i=1,\ldots,m,\forall k:\mathbf{Y}^i_k\in\mathcal{B}^i\right\}.\end{aligned}\tag{10}$$

Reformulating the problem in the braces of (10) to its dual can get the following equivalent problem:

$$\begin{aligned}\min_{\mathbf{D}\in\mathcal{D}}\max_{\{\boldsymbol{\alpha}_c\}_{c=1}^C}\min_{\{\boldsymbol{\mu}^i\in\mathcal{M}^i\}_{i=1}^m}&-\frac{1}{2}\sum_{c=1}^C\boldsymbol{\alpha}_c^T\widetilde{\mathbf{K}}_{\mathrm{F}}\boldsymbol{\alpha}_c\\ &+\sum_{i=1}^m\sum_{c=1}^C\sum_{j=1}^{n_i}\alpha^i_{j,c}\sum_{k:\mathbf{Y}^i_k\in\mathcal{B}^i}\mu^i_k\overline{y}^i_{k,j,c},\end{aligned}\tag{11}$$

where $\overline{y}^i_{k,j,c}$ is the element of $\mathbf{Y}^i_k$ at the jth row and cth column, and $\mathcal{M}^i$ is defined as $\mathcal{M}^i=\{\boldsymbol{\mu}^i\,|\,0\le\mu^i_k\le 1,\ \sum_{k:\mathbf{Y}^i_k\in\mathcal{B}^i}$

$\mu_k^i = 1\}$. If we denote $\widetilde{\mathcal{B}}^i = \{\widetilde{\mathbf{Y}}^i \mid \widetilde{\mathbf{Y}}^i = \sum_{k:\mathbf{Y}_k^i \in \mathcal{B}^i} \mu_k^i \mathbf{Y}_k^i,$ $\boldsymbol{\mu}^i \in \mathcal{M}^i\}$, according to [53, p. 24], $\widetilde{\mathcal{B}}^i$ is the convex hull of $\mathcal{B}^i$ which is the tightest convex relaxation of $\mathcal{B}^i$. Note that the optimization order of $\{\boldsymbol{\mu}, \mathbf{D}, \boldsymbol{\alpha}\}$ is exchangeable. The detailed derivation of Eq. (11) is in the supplementary material, available online.

Writing the objective function in (11) back to its primal form derives the following equivalent convex optimization problem:

$$\min_{\{\boldsymbol{\mu}^i \in \mathcal{M}^i\}_{i=1}^m} \min_{\{\mathbf{W}_c\}_{c=1}^C} \min_{\mathbf{D}\in\mathcal{D}} \sum_{c=1}^C \Bigg(\frac{\lambda_1}{2}\operatorname{tr}\big(\mathbf{W}_c^T\mathbf{W}_c\big) + \frac{\lambda_2}{2}\operatorname{tr}\big(\mathbf{W}_c^T\mathbf{D}^{-1}\mathbf{W}_c\big) + \sum_{i=1}^m \frac{1}{n_i}\sum_{j=1}^{n_i}\Bigg(\sum_{k:\mathbf{Y}_k^i\in\mathcal{B}^i} \mu_k^i \overline{y}_{k,j,c}^i - \mathbf{w}_{i,c}^T\mathbf{x}_j^i\Bigg)^2\Bigg). \tag{12}$$

Theorem 1. *Problem (12) is convex with respect to* $\{\boldsymbol{\mu}^i\}_{i=1}^m$, $\{\mathbf{W}_c\}_{c=1}^C$, *and* $\mathbf{D}$.

Proof. Because $\{\mathcal{M}^i\}_{i=1}^m$, $\{\mathbb{R}^{d\times m}\}_{c=1}^C$ and $\mathcal{D}$ are all convex sets, their Cartesian product $\mathcal{M}^1 \times \cdots \times \mathcal{M}^m \times \mathbb{R}^{d\times m}, \ldots, \mathbb{R}^{d\times m} \times \mathcal{D}$, i.e. the constraint, is also convex [53, p. 38], where $n = \sum_i n_i$. It is easy to see that the first and third terms of the objective function are convex by verifying that their *Hessian* matrices are positive semidefinite [53, p. 71]. The second term has been proved to be convex in [8]. Because the summation operation can preserve convexity, the objective function is convex. Therefore, Problem (12) is jointly convex with respect to all variables. □

Summarizing the aforementioned, Problem (12) is a convex relaxation of the original Problem (1). It has two equivalent forms (10) and (11). Problem (10) is the objective function of DMTFC.

5 Convex Discriminative Multitask Relationship Clustering

In this section, we will introduce the convex objective of the proposed DMTRC.

We extend the MTRL algorithm [19, Eq. (5)] to its multiclass unsupervised counterpart, the objective of DMTRC, which is formulated as the following MIP problem:

$$\min_{\{\mathbf{Y}^i \in \mathcal{B}^i\}_{i=1}^m} \min_{\{\mathbf{W}_c\}_{c=1}^C} \min_{\Omega\in\mathcal{A}} \sum_{c=1}^C \Bigg(\frac{\lambda_2}{2}\operatorname{tr}\big(\mathbf{W}_c\Omega^{-1}\mathbf{W}_c^T\big) + \frac{\lambda_1}{2}\operatorname{tr}\big(\mathbf{W}_c\mathbf{W}_c^T\big) + \sum_{i=1}^m \frac{1}{n_i}\sum_{j=1}^{n_i}\big(\overline{y}_{j,c}^i - \mathbf{w}_{i,c}^T\mathbf{x}_j^i\big)^2\Bigg), \tag{13}$$

where the matrix variable Ω is the covariance matrix that models the relationships between the task-specific models $\mathbf{w}_{i,c}$, and $\mathcal{A}$ is a convex constraint set defined as:

$$\mathcal{A} = \{\Omega \mid \Omega \in \mathbb{R}^{m\times m}, \Omega \succeq 0, \mathbf{tr}(\Omega) = 1\}. \tag{14}$$

Observing the factors that cause Problems (1) and (13) the MIP problems are the same, we use a similar convex relaxation procedure with (1)'s for (13). Due to the length limitation of the paper, we only report the main results.

The relaxed convex optimization problem of Problem (13) is formulated formally as follows:

$$\min_{\{\boldsymbol{\mu}^i \in \mathcal{M}^i\}_{i=1}^m} \min_{\{\mathbf{W}_i\}_{c=1}^C} \min_{\Omega\in\mathcal{A}} \sum_{c=1}^C \Bigg(\frac{\lambda_1}{2}\operatorname{tr}\big(\mathbf{W}_c\mathbf{W}_c^T\big) + \frac{\lambda_2}{2}\operatorname{tr}\big(\mathbf{W}_c\Omega^{-1}\mathbf{W}_c^T\big) + \sum_{i=1}^m \frac{1}{n_i}\sum_{j=1}^{n_i}\Bigg(\sum_{k:\mathbf{Y}_k^i\in\mathcal{B}^i} \mu_k^i \overline{y}_{k,j,c}^i - \mathbf{w}_{i,c}^T\mathbf{x}_j^i\Bigg)^2\Bigg). \tag{15}$$

The proof of the convexity of Problem (15) is similar with the proof of Theorem 1. Problem (15) has two equivalent forms. The first one is written as:

$$\min_{\boldsymbol{\Omega}\in\mathcal{A}} \max_{\{\boldsymbol{\alpha}_c\}_{c=1}^C} \Bigg\{ \max_{\{\theta_i\}_{i=1}^m} \sum_{i=1}^m \theta_i - \frac{1}{2}\sum_{c=1}^C \alpha_c^T \widetilde{\mathbf{K}}_{\mathrm{R}} \alpha_c \quad \text{s.t. } \theta_i \le \sum_{c=1}^C\sum_{j=1}^{n_i} \alpha_{j,c}^i \overline{y}_{j,c}^i, \forall i = 1,\ldots,m, \forall k : \mathbf{Y}_k^i \in \mathcal{B}^i \Bigg\}, \tag{16}$$

where $\widetilde{\mathbf{K}}_{\mathrm{R}} = \mathbf{K}_{\mathrm{R}} + \frac{1}{2}\boldsymbol{\Lambda}$ with $\mathbf{K}_{\mathrm{R}}$ denoted as the multitask-kernel matrix for relationship learning which is defined as:

$$K_{\mathrm{R}}\big(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2}\big) = \mathbf{e}_{i_1}^T \Omega(\lambda_1\Omega + \lambda_2\mathbf{I}_m)^{-1}\mathbf{e}_{i_2}\big\langle \mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2}\big\rangle. \tag{17}$$

We also obtain $\mathbf{W}_c$ as:

$$\mathbf{W}_c = \sum_{i=1}^m\sum_{j=1}^{n_i} \alpha_{j,c}^i \mathbf{x}_j^i \mathbf{e}_i^T \Omega(\lambda_1\Omega + \lambda_2\mathbf{I}_m)^{-1}. \tag{18}$$

The second equivalent form is written as:

$$\min_{\boldsymbol{\Omega}\in\mathcal{A}} \max_{\{\boldsymbol{\alpha}_c\}_{c=1}^C} \min_{\{\boldsymbol{\mu}^i\in\mathcal{M}^i\}_{i=1}^m} -\frac{1}{2}\sum_{c=1}^C \boldsymbol{\alpha}_c^T \widetilde{\mathbf{K}}_{\mathrm{R}} \boldsymbol{\alpha}_c + \sum_{i=1}^m\sum_{c=1}^C\sum_{j=1}^{n_i} \alpha_{j,c}^i \sum_{k:\mathbf{Y}_k^i\in\mathcal{B}^i} \mu_k^i \overline{y}_{k,j,c}^i. \tag{19}$$

Summarizing the aforementioned, Problem (15) is a convex relaxation of the original Problem (13). It has two equivalent forms (15) and (19). Problem (16) is the objective function of DMTRC.

6 Optimization Algorithm

In this section, we are to solve DMTFC (10) and DMTRC (16) in a uniform framework. This framework utilizes the fact that there are only two different points between them: 1) the multitask kernel functions are different, see Eqs. (7) and (17); 2) the convex sets $\mathcal{D}$ and $\mathcal{A}$ are different, see Eqs. (2) and (14). To facilitate the mathematical representation, we write (10) and (16) as the following uniform objective:

$$\max_{\{\boldsymbol{\alpha}_c\}_{c=1}^C} \min_{\mathbf{Z}\in\mathcal{Z}} \max_{\{\boldsymbol{\theta}_i\}_{i=1}^m} \sum_{i=1}^m \theta_i - \frac{1}{2}\sum_{c=1}^C \boldsymbol{\alpha}_c^T \widetilde{\mathbf{K}} \boldsymbol{\alpha}_c \quad \text{s.t. } \theta_i \le \sum_{c=1}^C\sum_{j=1}^{n_i} \alpha_{j,c}^i \overline{y}_{j,c}^i, \forall i = 1,\ldots,m, \forall k : \mathbf{Y}_k^i \in \mathcal{B}^i, \tag{20}$$

where $\mathbf{Z}$ stands for $\mathbf{D}$ in (10) or Ω in (16), $\mathcal{Z}$ stands for $\mathcal{D}$ in (10) or $\mathcal{A}$ in (16), and $\widetilde{\mathbf{K}}$ stands for $\widetilde{\mathbf{K}}_{\mathrm{F}}$ in (7) or $\widetilde{\mathbf{K}}_{\mathrm{R}}$ in (17).

Due to the length limitation of the paper, we present the optimization algorithm briefly as follows, leaving the detailed derivation in the supplementary material, available online.

The solution framework is an alternating method. First, it decomposes the unsupervised problem (20) to a serial supervised multiclass MTL problem by the cutting-plane algorithm [49] and the extended level method (ELM) [50], [51], where the decomposition algorithm can be seen as a multitask extension of the SVR-M3C algorithm [34]. Then, it solves each supervised multiclass MTL problem in an alternating way, which decomposes the multiclass MTL to a serial supervised single-task regression problems eventually. Note that the difference of the optimization procedure between DMTFC and DMTRC only appears in the supervised learning in Section 6.3.

6.1 Optimizing (20) via Cutting-Plane Algorithm

Because the number of the constraints in Problem (20) is exponential large with respect to n, directly optimizing (20) is impossible when the data set contains over dozens of examples. Hence, we adopt CPA [49] to solve it approximately. CPA iterates the following two steps. The first step is to solve the following reduced cutting plane subproblem:

$$\begin{aligned}&\max_{\{\boldsymbol{\alpha}_c\}_{c=1}^C}\min_{\mathbf{Z}\in\mathcal{Z}}\max_{\{\theta_i\}_{i=1}^m}\sum_{i=1}^m\theta_i-\frac{1}{2}\sum_{c=1}^C\boldsymbol{\alpha}_c^T\widetilde{\mathbf{K}}\boldsymbol{\alpha}_c\\&\left.\text{s.t. }\ \theta_i\le\sum_{c=1}^C\sum_{j=1}^{n_i}\alpha_{j,c}^i\overline{y}_{j,c}^i,\forall i=1,\ldots,m,\forall k:\mathbf{Y}_k^i\in\mathcal{Y}^i\right\},\end{aligned}\tag{21}$$

where $\mathcal{Y}^i\subset\mathcal{B}^i$ represents the pool of the most violated constraints, The second step is to calculate the most violated constraint, denoted as $\{\mathbf{Y}^i_{|\mathcal{Y}^i|+1}\}_{i=1}^m$, by solving the following integer matrix optimization problem

$$\min_{\mathbf{Y}^i_{|\mathcal{Y}^i|+1}}\sum_{c=1}^C\sum_{j=1}^{n_i}\alpha_{j,c}^i\overline{y}^i_{|\mathcal{Y}^i|+1,j,c},\forall i=1,\ldots,m,\tag{22}$$

and then add $\mathbf{Y}^i_{|\mathcal{Y}^i|+1}$ to $\mathcal{Y}^i,\forall i=1,\ldots,m$, respectively. Thanks to the constraints on $\mathbf{Y}^i$ (defined in $\mathcal{B}^i$, i.e. Eq. (3)), the problem can be solved in time $\mathcal{O}(\sum_{i=1}^m Cn_i\log(Cn_i))$, see [34, Algorithm 6] for the algorithm.

6.2 Optimizing (21) via Extended Level Method

Like the full Problem (20), the cutting-plane subproblem (21) also has an equivalent form:

$$\begin{aligned}&\max_{\{\boldsymbol{\alpha}_c\}_{c=1}^C}\min_{\mathbf{Z}\in\mathcal{Z}}\min_{\{\boldsymbol{\mu}^i\in\mathcal{M}_{\mathcal{Y}}^i\}_{i=1}^m}-\frac{1}{2}\sum_{c=1}^C\boldsymbol{\alpha}_c^T\widetilde{\mathbf{K}}\boldsymbol{\alpha}_c\\&\qquad+\sum_{i=1}^m\sum_{c=1}^C\sum_{j=1}^{n_i}\alpha_{j,c}^i\sum_{k:\mathbf{Y}_k^i\in\mathcal{Y}^i}\mu_k^i\overline{y}_{k,j,c}^i,\end{aligned}\tag{23}$$

where $\mathcal{M}_{\mathcal{Y}}^i=\{\boldsymbol{\mu}^i\,|\,0\le\mu_k^i\le 1,\sum_{k=1}^{|\mathcal{Y}^i|}\mu_k^i=1\}$.

Problem (23) is a concave-convex optimization problem that is convex on $\boldsymbol{\mu}$ and $\mathbf{Z}$ and concave on $\boldsymbol{\alpha}$. We will optimize it via ELM [50] which is an efficient alternating method that aims to find the saddle point of the problem. ELM iterates the following two steps until convergence. The first step is to optimize $\{\boldsymbol{\mu}^i\}_{i=1}^m$ given fixed $\{\boldsymbol{\alpha}\}_{c=1}^C$ and $\mathbf{Z}$ by constructing a cutting-plane model on the problem. See the supplement for this complicated cutting-plane model. The second step is to optimize $\{\boldsymbol{\alpha}\}_{c=1}^C$ and $\mathbf{Z}$ together given fixed $\{\boldsymbol{\mu}^i\}_{i=1}^m$, which is formulated as follows:

$$\min_{\mathbf{Z}\in\mathcal{Z}}\max_{\{\alpha_c\}_{c=1}^C}-\frac{1}{2}\sum_{c=1}^C\boldsymbol{\alpha}_c^T\widetilde{\mathbf{K}}\boldsymbol{\alpha}_c+\sum_{i=1}^m\sum_{c=1}^C\sum_{j=1}^{n_i}\alpha_{j,c}^i\sum_{k:\mathbf{Y}_k^i\in\mathcal{Y}^i}\mu_k^i\overline{y}_{k,j,c}^i.\tag{24}$$

Note that Problem (24) is the dual form of a supervised MTL problem. The reason why we solve DMTC in the dual form but not primal form is because that we need the Lagrange parameter α to solve Problem (22) but not only for introducing the nonlinear kernels.

6.3 Optimizing (24) via Alternating Method

We adopt an alternating method that is similar with [19] for Problem (24), which iterates the following two steps until convergence.

The first step is to optimize $\{\boldsymbol{\alpha}\}_{c=1}^C$ given fixed $\mathbf{Z}$, which is equivalent to the following problem:

$$\sum_{c=1}^C\left(\max_{\boldsymbol{\alpha}_c}\sum_{i=1}^m\sum_{j=1}^{n_i}\alpha_{j,c}^i\sum_{k:\mathbf{Y}_k^i\in\mathcal{Y}^i}\mu_k^i\overline{y}_{k,j,c}^i-\frac{1}{2}\boldsymbol{\alpha}_c^T\widetilde{\mathbf{K}}\boldsymbol{\alpha}_c\right).\tag{25}$$

When $\mathbf{Z}$ is fixed, the terms in the brackets are mutually independently. Hence, we solve each term independently, which is a supervised single-task regression problem, where the data from all tasks are considered as the data from a single task.

The second step is to optimize $\mathbf{Z}$ given fixed $\{\alpha_c\}_{c=1}^C$, which is formulated as

$$\min_{\mathbf{Z}\in\mathcal{Z}}-\frac{1}{2}\sum_{c=1}^C\boldsymbol{\alpha}_c^T\widetilde{\mathbf{K}}\boldsymbol{\alpha}_c+\sum_{i=1}^m\sum_{c=1}^C\sum_{j=1}^{n_i}\alpha_{j,c}^i\sum_{k:\mathbf{Y}_k^i\in\mathcal{Y}^i}\mu_k^i\overline{y}_{k,j,c}^i.\tag{26}$$

Note that $\widetilde{\mathbf{K}}$ is a function of $\mathbf{Z}$.

Specifying (25) and (26) as a part of DMTFC. We replace $\mathbf{Z}$ and $\mathcal{Z}$ in the equations by $\mathbf{D}$ and $\mathcal{D}$ respectively. For (25), the multitask kernel $\widetilde{\mathbf{K}}$ should be specified by Eq. (7). The calculation of $\widetilde{\mathbf{K}}$ will be expensive when the dimension of the observation d is large, since the time complexity of the matrix inversion in (7) is $\mathcal{O}(d^3)$ in the worst cases. For (26), we can get the closed solution of $\mathbf{D}$ as

$$\mathbf{D}=\frac{\left(\sum_{c=1}^C\mathbf{W}_c\mathbf{W}_c^T\right)^{\frac{1}{2}}}{\mathrm{tr}\left(\left(\sum_{c=1}^C\mathbf{W}_c\mathbf{W}_c^T\right)^{\frac{1}{2}}\right)}$$

where $\mathbf{W}_c$ is defined in (8). The derivation is analogous to [8, Appendix 1].

Specifying (25) and (26) as a part of DMTRC. We replace $\mathbf{Z}$ and $\mathcal{Z}$ by Ω and $\mathcal{A}$ respectively in the equations. For (25), $\widetilde{\mathbf{K}}$ should be specified by Eq. (17). The calculation of $\widetilde{\mathbf{K}}$ will be expensive when the task number m is large, since the time complexity of the matrix inversion in (17) is $\mathcal{O}(m^3)$ in the worst cases. For (26), we can get the closed solution of Ω as

$$\Omega = \frac{\left(\sum_{c=1}^{C} \mathbf{W}_c^T \mathbf{W}_c\right)^{\frac{1}{2}}}{\operatorname{tr}\left(\left(\sum_{c=1}^{C} \mathbf{W}_c^T \mathbf{W}_c\right)^{\frac{1}{2}}\right)}$$

where $\mathbf{W}_c$ is defined in (18). The derivation is analogous to [19, Eq. (13)].

7 Learning with Nonlinear Kernels

To incorporate the nonlinear feature mapping to DMTFC and DMTRC, we only need to modify the multitask kernel. Specifically, for DMTFC, we only need to modify Eq. (7) to $K_{\mathrm{F}}(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2}) = \mathbf{e}_{i_1}^T \phi(\mathbf{x}_{j_1}^{i_1})^T \mathbf{D}\,(\lambda_1 \mathbf{D} + \lambda_2 \mathbf{I}_d)^{-1} \phi(\mathbf{x}_{j_2}^{i_2}) \mathbf{e}_{i_2}$ and modify Eq. (8) to $\mathbf{W}_c = \sum_i \sum_j \alpha_j^i \mathbf{D}(\,\lambda_1 \mathbf{D} + \lambda_2 \mathbf{I}_d)^{-1} \phi(\mathbf{x}_j^i) \mathbf{e}_i^T$, where $\phi(\cdot)$ is the kernel-induced feature mapping. We may use the kernel decomposition techniques, such as kernel principle component analysis or Cholesky decomposition, to get $\phi(\cdot)$ approximately and explicitly, or the methods described in [8], [63] to incorporate the kernels. In this paper, we used the kernel principle component analysis to get $\phi(\cdot)$ explicitly. Similarly, for DMTRC, we only need to modify Eq. (17) to $K_{\mathrm{R}}(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2}) = \mathbf{e}_{i_1}^T \Omega(\lambda_1 \Omega + \lambda_2 \mathbf{I}_m)^{-1} \mathbf{e}_{i_2} K(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2})$ and modify Eq. (18) to $\mathbf{W}_c = \sum_i \sum_j \alpha_j^i \phi(\mathbf{x}_j^i) \mathbf{e}_i \Omega(\lambda_1 \Omega + \lambda_2 \mathbf{I}_m)^{-1}$, where $K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$.

8 Complexity Analysis

Because the optimization algorithm can be seen as a technical combination of SVR-M3C [34], MTFL [8], and MTRL [19], where the outer two loops (i.e. Sections 6.1 and 6.2) is a multitask extension of SVR-M3C and the inner loop (i.e. Section 6.3) can be seen as a special case of the multiclass extensions of MTFL/MTRL, the overall time and storage complexities of the optimization algorithm are dominated by the most expensive algorithm between SVR-M3C and MTFL/MTRL. SVR-M3C has a time complexity of $\mathcal{O}(n \log n)$ and a storage complexity of $\mathcal{O}(n)$ [34]. It is also easy to observe that the worst case of MTFL has a time complexity of $\mathcal{O}(n^2 + d^3)$ and a storage complexity of $\mathcal{O}(n^2)$, and that the worst case of MTRL has a time complexity of $\mathcal{O}(n^2 + m^3)$ and a storage complexity of $\mathcal{O}(n^2)$. Hence, DMTFC is suitable to middle scale and low dimensional problems, while DMTRC is suitable to middle scale problems with small task numbers. The main obstacle that hinders DMTFC and DMTRC from large scale problems is the time-demanding kernel calculation and matrix inversion in (7) and (17). To overcome it, dimension reduction techniques, sparse MTL techniques, distributed cluster ensembles and sparse kernel estimations might be helpful. But as will be shown in the experimental section, when the data size is large scale, the benefit of multitask clustering over single-task clustering (STC) will vanish. Finally, we think the complexity will not hinder them from practical use.

9 Discussion: Bayesian Framework of Discriminative Multitask Clustering

In this section, we will uniform the two DMTC objectives in the Bayesian framework.

For a m-task DMTC problem, we try to estimate the *maximum a posteriori* of $\{\mathbf{W}_c\}_{c=1}^{C}$ as

$$\begin{aligned} &\max_{\{\mathbf{W}_c\}_c, \{\mathbf{Y}^i\}_i} p(\{\mathbf{W}_c\}_c, \{\mathbf{Y}^i\}_i \,|\, \{\mathbf{X}^i\}_i) \\ &= \max_{\{\mathbf{W}_c\}_c, \{\mathbf{Y}^i\}_i} p(\{\mathbf{W}_c\}_c) p(\{\mathbf{Y}^i\}_i \,|\, \{\mathbf{X}^i\}_i, \{\mathbf{W}_c\}_c). \end{aligned} \tag{27}$$

Equation (27) contains two parts. The first part $p(\{\mathbf{W}_c\}_c)$ is a prior defining the task relationship. The second part defines a serial discriminative clusterings on all tasks. How to specify the prior and the discriminative model is the central problem.

Here, we make four probabilistic assumptions on Problem (27) for balancing the difficulty of solving DMTC and the effectiveness of DMTC.

a) *Class evenness assumption.* We assume that the empirical marginal distribution of label $p(y)$ in each task is known, which results in the class balance constraint in (3).

b) *Multivariate Gaussian prior assumption.* The prior defines what to share in MTC. In this paper, we follow Zhang and Yeung's formulation [19, Equation (2)] for the multivariate Gaussian prior

$$p(\{\mathbf{W}_c\}_c) \propto \prod_{c=1}^{C} \left(q(\mathbf{W}_c) \prod_{i=1}^{m} \mathcal{N}\left(\mathbf{w}_{i,c} \,|\, \mathbf{0}_d, \sigma_1^2 \mathbf{I}_d\right) \right), \tag{28}$$

where $\mathcal{N}(\mathbf{A}, \mathbf{B})$ is a multivariate normal distribution with $\mathbf{A}$ and $\mathbf{B}$ as the mean and covariance matrix respectively, and $q(\mathbf{W}_c)$ is a distribution that the rows or columns of $\mathbf{W}_c$ are independent Gaussians. See (29) and (30) below for the definition of $\mathbf{W}_c$. Note that restricting all tasks to have the same covariance $\sigma_1^2 \mathbf{I}_d$ might be too restrictive. In practice, we can use different covariances for different tasks.

In this paper, we consider two kinds of $q(\mathbf{W}_c)$. The first kind defines a shared feature representation:

$$q_f(\mathbf{W}_c) = \frac{\exp\left(-\frac{1}{2}\operatorname{tr}\left(\mathbf{W}_c^T \mathbf{D}^{-1} \mathbf{W}_c\right)\right)}{(2\pi)^{md/2} |\mathbf{D}|^{d/2}}. \tag{29}$$

The second kind follows Zhang and Yeung's formulation [19, Eq. (2)], which defines the relationship between the tasks:

$$q_r(\mathbf{W}_c) = \frac{\exp\left(-\frac{1}{2}\operatorname{tr}\left(\mathbf{W}_c \Omega^{-1} \mathbf{W}_c^T\right)\right)}{(2\pi)^{md/2} |\Omega|^{m/2}}. \tag{30}$$

c) *Task independence assumption.* We assume that when $\{\mathbf{W}_c\}_c$ is sampled from the prior distribution, the tasks are mutually independent:

$$\begin{aligned} &p\left(\{\mathbf{Y}^i\}_i \middle| \{\mathbf{X}^i\}_i, \{\mathbf{W}_c\}_c\right) = \prod_{i=1}^{m} p(\mathbf{Y}^i | \mathbf{X}^i, \{\mathbf{w}_c^i\}_c) \\ &= \prod_{i=1}^{m} \prod_{c=1}^{C} p(\overline{\overline{\mathbf{y}}}_c^i \,|\, \mathbf{X}^i, \mathbf{w}_c^i) = \prod_{i=1}^{m} \prod_{j=1}^{n_i} \prod_{c=1}^{C} p(\overline{y}_{j,c}^i \,|\, \mathbf{x}_j^i, \mathbf{w}_c^i). \end{aligned} \tag{31}$$

With this assumption, we can incorporate any advanced binary-class discriminative clustering into $p(\overline{\mathbf{y}}_c^i \,|\, \mathbf{X}^i, \mathbf{w}_c^i)$ without modifying the clustering algorithm significantly.

d) *Gaussian assumption on the discriminative clustering model*. We assume $p(\overline{y}_{j,c}^i \,|\, \mathbf{x}_j^i, \mathbf{w}_c^i)$ in (31) is Gaussian:

$$p(\overline{y}_{j,c}^i \,|\, \mathbf{x}_j^i, \mathbf{w}_{i,c}) = \mathcal{N}(\overline{y}_{j,c}^i \,|\, \mathbf{w}_{i,c}^T \mathbf{x}_j^i, \sigma_2^2). \tag{32}$$

This assumption makes the discriminative clustering a regression problem but not a classification problem, which might not be the real case since $\overline{y}_{j,c}^i \in \{-\frac{1}{C-1}, 1\}$ is a discrete variable. However, it is known that even in the supervised classification problem, if we set Problem (31) with a non-Gaussian likelihood, the computations of predictions are analytically intractable [64, p. 39].

Substituting Eqs. (3), (28), (29), (31) and (32) into Problem (27) and taking the negative logarithm of (27) can derive the following objective function:

$$\begin{aligned}\min_{\{\mathbf{Y}^i \in \mathcal{B}^i\}_{i=1}^m} \min_{\{\mathbf{W}_c\}_{c=1}^C} \min_{\mathbf{D}} \sum_{c=1}^{C} \Bigg(& \frac{\lambda_1}{2} \mathrm{tr}(\mathbf{W}_c^T \mathbf{W}_c) \\ & + \frac{\lambda_2}{2} \mathrm{tr}(\mathbf{W}_c^T \mathbf{D}^{-1} \mathbf{W}_c) + \frac{d\lambda_2}{2} \ln|\mathbf{D}| \\ & + \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} (\overline{y}_{j,c}^i - \mathbf{w}_{i,c}^T \mathbf{x}_j^i)^2 \Bigg),\end{aligned} \tag{33}$$

where λ_1 and λ_2 are two tunable hyper parameters that are related to σ_1 and σ_2. Replacing the concave function $\ln|\mathbf{D}|$ to the convex constraint set $\mathcal{D}$ derives the DMTFC objective (1). The DMTRC objective can be derived in the same way as the above, except using (30) instead of (29).

10 Experiments

In this section, we will compare the proposed DMTFC and DMTRC algorithms with 10 clustering algorithms on the UCI *pendigits* toy data set and two benchmark data sets—*multi-domain newsgroups* data set and *multi-domain sentiment* data set. When we evaluate the running time, each algorithm is run with only one CPU.

The competitive algorithms can be categorized to two classes. The first class are the single task clustering algorithms. They are 1) K-Means (KM), 2) Kernel K-Means (KKM) with the RBF kernel, 3) Normalized Cut (NC) [24] with the RBF kernel, 4) SVR-M3C with linear kernel [34], 5) the Discriminative STC (DSTC) algorithm, 6) KM that groups all tasks into a single task (ALL KM), 7) ALL KKM, and 8) ALL NC, where DSTC is the single task version of our DMTRC. The DSTCs with the linear kernel and the RBF kernel are denoted as DSTC_l and DSTC_r respectively. The second class are the state-of-the-art MTC algorithms. They are 1) Learning the Shared Subspace for MTC (LSSMTC) [39], 2) Learning a Spectral Kernel for MTC (LSKMTC) [42], and 3) Multitask Bregman Clustering with Pairwise task regularization (MBC-P) [41]. The experiments of the competitive algorithms are run exactly with the authors' experimental settings.

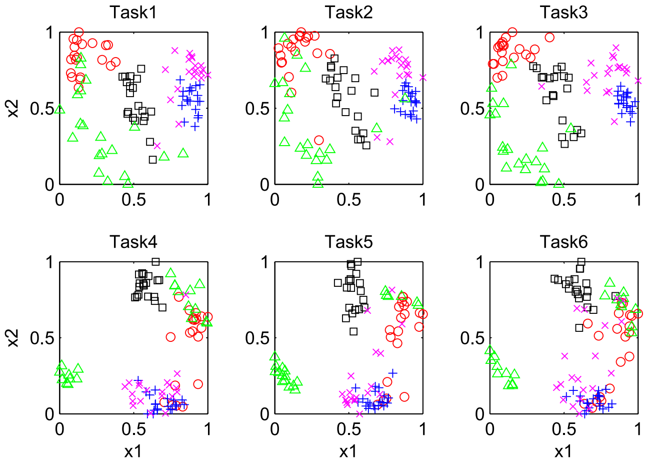

Fig. 1. Visualization of the tasks on the pendigits data. The true labels are indicated by different colors and different symbols. PCA is used to generate the figure.

For our DMTFC and DMTRC, λ_1 and λ_2 are both searched from $\{0, 2^{-10}, 2^{-8}, \ldots, 2^{-2}\}$, we make a strong assumption that we know the class distribution beforehand, so that $l_{i,c}$ in Eq. (3) is set to $l_{i,c} = \mathbf{1}_{n_i}^T \mathbf{y}^{\star i}_c / n_i$ where $\mathbf{y}^{\star i}_c$ is the cth column of the ground truth label matrix $\mathbf{Y}^i$ of the ith task. The DMTFC and DMTRC with the linear kernel are denoted as DMTFC_l and DMTRC_l respectively, and those with the RBF kernel are denoted as DMTFC_r and DMTRC_r respectively.

The kernel width of all algorithms that work with the RBF kernel is searched from $\{2^{-2}, 2^{-1}, 2^0, 2^1, 2^2\} \cdot A$, where A is the average euclidean distance of the data. The data are normalized into the range of [0, 1] in dimension. All computation time is recorded except that consumed on normalizing the data set. The data sets used in experiments are provided with labels. Therefore, the performance is evaluated as comparing the predicted labels with the ground truth labels using normalized mutual information (NMI) [54].

10.1 Results on Pendigits Data Set

In this subsection, the *pendigits* data set in the UCI machine learning repository is used as a toy data set for capturing the main characteristics of the proposed DMTC algorithms. The pendigits data set contains 10 hand written integer digits ranging from 0 to 9. It consists of 11,256 observations and 16 attributes. Each digit consists of about 1,100 observations. Although the pendigits data set is a single task clustering problem, we generate a multitask clustering problem from it: First, we take $0, 3, 6, 8, 9$ as one group, and $1, 2, 4, 5, 7$ as the other group. Then, we repeatedly sample 20 observations from each digit in the first group for three times. Again, we do the same thing to the second group. Because each repeat forms a five-class clustering task that contains 100 observations, we obtain six tasks in total, where Tasks 1, 2 and 3 are examples from the first group and Tasks 4, 5, and 6 are examples from the second group. Because the data are too small to cover the distributions of the digits, we can regard Tasks 1, 2 and 3 are relevant but not the same, so as to Tasks 4, 5, and 6. We also regard that Tasks 1, 2 and 3 are irrelevant to Tasks 4, 5, and 6. A visualized example of

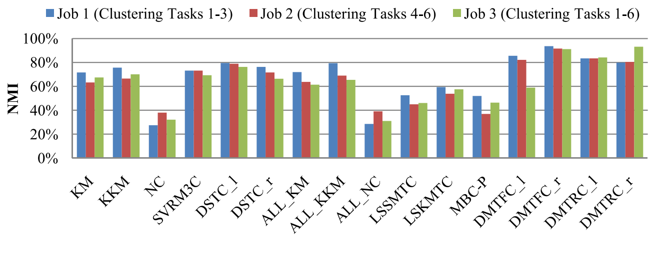

Fig. 2. NMI comparison on the pendigits data set.

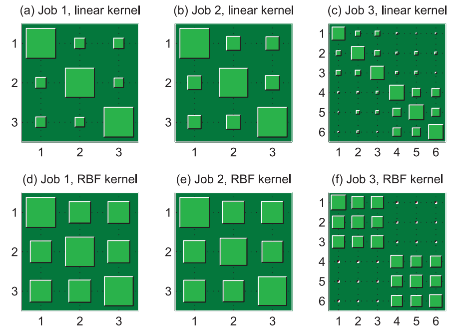

Fig. 4. Hinton diagram of the task relationship learned by DMTRC on the pendigits data set (i.e. the learned covariance between the task-specific models, i.e. Ω). The grid in green means the tasks are related. The grid in red means the tasks are reverse. The bigger the grid is, the more positive/negative the relationship is.

the data distributions associated with the six tasks are shown in Fig. 1. We run three jobs on the six tasks. Job 1 is to cluster Tasks 1, 2, and 3. Job 2 is to cluster Tasks 4, 5, and 6. Job 3 is to cluster Tasks 1, 2, 3, 4, 5 and 6 together. For each MTC job, we repeat the experiment 30 times. For each single repeat, we also repeat the referenced algorithms 50 times and report the average results. For DMTFC$_r$, KPCA is used for getting $\phi(\mathbf{x})$ explicitly. It retains the top 100 largest eigenvalues and their eigenvectors.

Fig. 2 shows the NMI comparison over the three jobs. From the figure, we can get the following interesting phenomena. First, except for DMTFC$_l$, the proposed DMTC algorithms achieve higher NMIs than the referenced methods. This phenomenon demonstrates the effectiveness of the proposed MTC algorithms. Second, except for DMTRC$_r$, the NMIs of all algorithms in Job 3 are lower than those in Jobs 1 and 2. This phenomenon is particularly apparent in DMTFC$_l$. It shows that the unrelated tasks or the reverse distributions worsen the clustering performance significantly. This phenomenon also shows that when the tasks are really related, learning a powerful feature representation is better than minimizing the distances between the task-specific models, but when the tasks are irrelevant, learning a feature representation forcibly is very harmful while learning the task relationship can avoid the negative transfer amazingly. To better explain this, we visualize $\mathbf{D}$ and $\mathbf{\Omega}$ in Figs. 3 and 4 respectively. For DMTFC, in Figs. 3a, 3b, 3d, 3e, and 3f, the relationships of the features have been learned successfully by DMTFC. But in Fig. 3c, DMTFC$_l$ fails in learning a common feature representation, i.e., most features are recognized as mutually independent. For DMTRC, in Fig. 4, we can observe that DMTRC can capture the relationships of the tasks successfully no matter in Jobs 1 and 2 or in Job 3, which accounts for the immunity of DMTRC to the negative transfer. Note that this study has been conducted in many supervised MTL works, but to our knowledge, this is the first work that captures the task relationship successfully in the unsupervised learning scenario. Third, the referenced MTCs do not achieve better NMIs than the STCs. One possible explanation for this is that the referenced MTCs suffer from local minima more seriously than the STCs.

The above experiment assumes that the class distributions are known with all parameters $l_{i,c}$ setting to the ideal situation $\mathbf{1}_{n_i}^T \mathbf{y}{\star}_c^i / n_i = 0$. In this paragraph, we will investigate how the class evenness assumption affects the performance by setting all $\{\{l_{i,c}\}_{c=1}^C\}_{i=1}^m$ to the same value that is selected from $\{0, 0.03, 0.1, 0.2, 0.3\}$. The results are shown in Fig. 5. From the figure, we can observe the following phenomena: 1) In all settings, DMTC can benefit from joint training of all tasks except DMTFC$_l$. 2) Setting the class balance parameters to a value 0.03 that is slightly biased from the ideal situation can achieve even better performance, which means that if we select l properly around the ideal value, the performance is guaranteed. 3) DMTC is sensitive to l, if parameter l is set improperly, the performance will degrade dramatically. Hence, for DMTC's practical use, we should select l carefully.

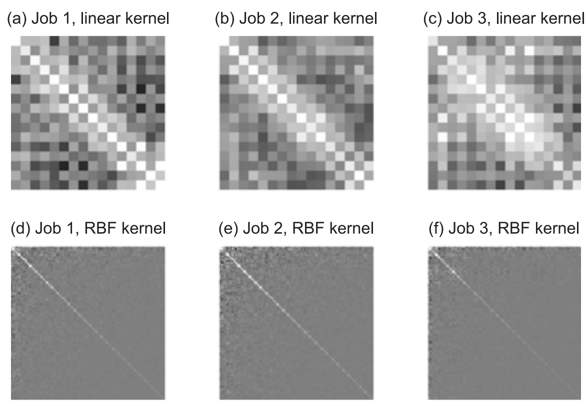

Fig. 3. Visualization of the shared feature filter learned by DMTFC on the pendigits dataset (i.e. the learned covariance between the features, i.e. $\mathbf{D}$). The more grey the grid is, the weaker the filter contributes to the new feature representation.

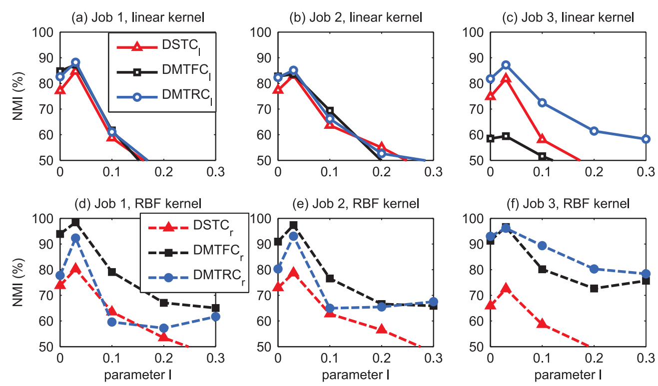

Fig. 5. Clustering performance with respect to the class balance parameter l on the pendigits data set.

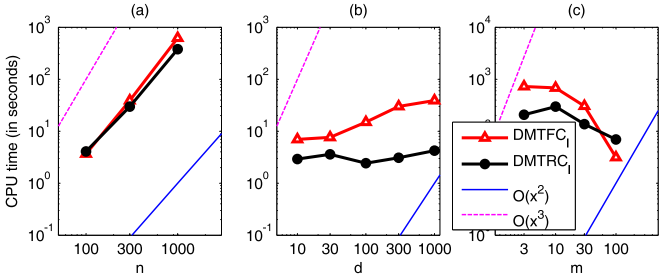

Fig. 6. Time complexities with respect to the data set size of each task (n), feature dimension (d), and number of tasks (m). The symbol x in the legends $\mathcal{O}(x^2)$ and $\mathcal{O}(x^2)$ stands for n, d or m in (a), (b) or (c) respectively.

10.2 Complexity Analysis on Synthetic Data Set

In this subsection, we will study the time complexities of DMTFC and DMTRC with respect to the number of examples of each task (i.e. n), feature dimension (i.e. d), and number of tasks (i.e. m) respectively. We generate each dimension of each class of each binary-class synthetic task from a Gaussian distribution, whose mean is sampled uniformly from $[0, 1]$ and variance varies uniformly in $[0.5, 5]$. The parameters of the proposed methods are as follows. Only linear kernel is considered. $\lambda_1 = \lambda_2 = 2^{-10}$, $l = 0$.

The time complexities with respect to n are shown in Fig. 6a, where $d = 3$ and $m = 3$. The time complexities with respect to d are shown in Fig. 6b, where $n = 100$ and $m = 3$. The time complexities with respect to m are shown in Fig. 6c where $n = 3,000/m$ and $d = 10$. From the figures, we can conclude that the time complexities with respect to n are $\mathcal{O}(n^2)$, but the time complexities with respect to d and m are generally not in the worst cases, i.e. $\mathcal{O}(d^3)$ and $\mathcal{O}(m^3)$. The reasons are analyzed as follows. Compared to the CPU time consumed on constructing the kernel, which scales with $\mathcal{O}((nm)^2)$, the time consumed on the matrix inverse is quite small. Moreover, when nm is given, more task number only means the multitask-kernel matrix is more sparse, so that the methods need even less time to calculate the kernel matrix. This accounts for the interesting phenomenon of Fig. 6c.

10.3 Results on Multi-Domain Newsgroups Data Set

The 20-newsgroups data set is a widely used benchmark data set that is a collection of about 20,000 messages collected from 20 different *usenet* newsgroups, 1,000 messages from each. After postprocessing, each message is a vector with 26,214 dimensions. We define a three class MTC job on the 20-newsgroups in Table 1. From the table, we can see that Tasks 1 and 2 are highly related, Tasks 1 to 5 are somewhat related, while Task 6 seems an outlier task. Based on the above task definition, we generate four MTC problems by randomly selecting 5, 10, 20, and 40 percent of the data from each class, so as to observe how the data number influences the effectiveness of DMTC. Because most algorithms are quite inefficient in high dimensional data sets, we use PCA to project the data set to a 100-dimensional subspace. DMTC and DSTC only use the linear kernel. The DMTRC$_l$ and DSTC$_l$ without the PCA projection, which are denoted as *DMTRC$_l$ and *DSTC$_l$ respectively, will also be investigated.

TABLE 1
Task Definition on the 20-Newsgroups Data Set

ID	Names of the classes
Task 1	comp.sys.mac.hardware *versus* rec.sport.hockey *versus* sci.electronics
Task 2	comp.sys.ibm.pc.hardware *versus* rec.sport.baseball *versus* sci.crypt
Task 3	comp.windows.x *versus* rec.autos *versus* talk.politics.guns
Task 4	comp.os.ms-windows.misc *versus* sci.med *versus* talk.politics.mideast
Task 5	rec.motorcycles *versus* sci.space *versus* talk.politics.misc
Task 6	misc.forsale *versus* alt.atheism *versus* soc.religion.christian

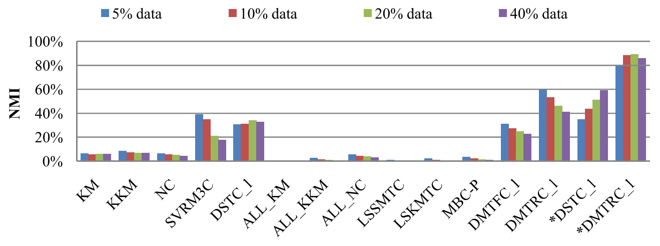

Fig. 7. NMI comparison on the 20-newsgroups dataset. a percent is short for "experiments running with a percent data of the data set."

Fig. 7 shows the NMI comparison. From the figure, we can observe the following experimental phenomena. First, the proposed convex discriminative clustering algorithms are apparently better than the referenced methods in the same experimental environment. Second, DMTRC$_l$ is much better than DSTC$_l$ which shows that the task relationship is learned successfully. Third, DMTFC$_l$ is slightly worse than DSTC$_l$ which means that we cannot learn a strong shared feature representation across the tasks. This phenomenon might be caused by the PCA projection where much useful information for constructing the feature representation is lost, however, we cannot get its performance in the original data set due to its inefficiency in high dimensional data. Fourth, when the PCA projection is used to form the experimental environment, the performances of the clustering algorithms are getting worse when more data is used. On the contrary, when PCA is not used, the performances of both *DSTC$_l$ and *DMTRC$_l$ are getting better. This phenomenon tells us that when more data is available, the features should provide more abundant information so as to make the models available to be more complicated for describing the more variant distributions. It also shows the power of DSTC and DMTRC on high dimensional data sets. Moreover, it demonstrates that the power of the proposed discriminative clusterings do not rely on the predefined models for

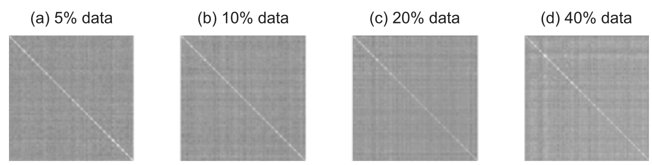

Fig. 8. Visualizations of $\mathbf{D}$ of DMTFC$_l$ on the 20-newsgroups data set.

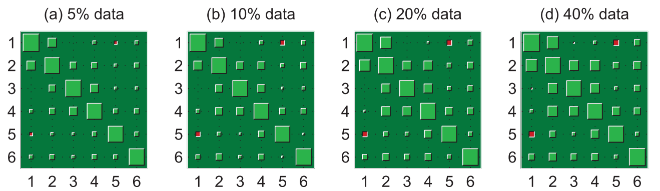

Fig. 9. Hinton diagrams of $\mathbf{\Omega}$ of DMTRC$_l$ on the 20-newsgroups dataset.

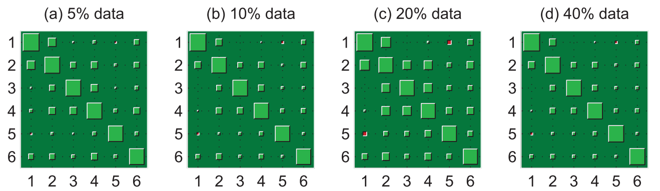

Fig. 10. Hinton diagrams of $\mathbf{\Omega}$ of *DMTRC$_l$ on the 20-newsgroups data set.

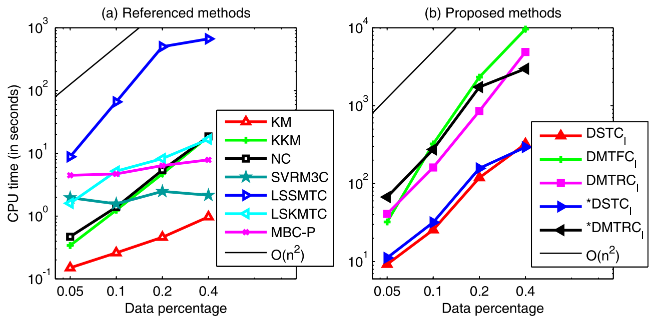

Fig. 11. CPU time comparison on the 20-newsgroups.

describing the data distribution which is an apparent superiority to the generative clusterings.

To show how well the feature representation is learned, we visualize $\mathbf{D}$ of DMTFC$_l$ in Fig. 8. The figure shows that most features are considered as mutually independent, which might account for the ineffectiveness of DMTFC$_l$.

To demonstrate how well the task relationship is learned, we list the hinton diagrams of $\mathbf{\Omega}$ of DMTRC$_l$ and *DMTRC$_l$ in Figs. 9 and 10 respectively. The figures show that both methods can learn the task relationships in different percentages of data equivalently well. They also show that the task relationship is different from what we have defined in Table 1. As an example, Task 6 is originally designed as an outlier task, but it contributes to the performance positively. This phenomenon is worth of further study.

Fig. 11 gives the CPU time comparison. The figure shows that although the proposed methods have higher absolute time, both the proposed algorithms and the referenced methods have a time complexity of $\mathcal{O}(n^2)$ except KM, SVR-M3C, LSKMTC and MBC-P, which means that they are all unavailable for large-scale problems.

The results on each individual task and the stability analysis are described in the supplementary materials, available online.

10.4 Results on Multi-Domain Sentiment Data Set

The multi-domain sentiment data set is a widely used benchmark data set that was originally designed for the MTL research propose. It contains product reviews taken from Amazon.com from many product types (domains or

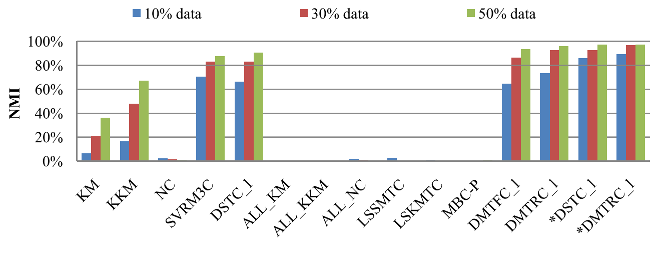

Fig. 12. NMI comparison on the sentiment data set. a percent is short for "experiments running with a percent data."

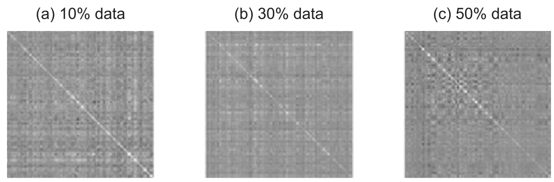

Fig. 13. Visualizations of $\mathbf{D}$ of DMTFC$_l$ on the sentiment data set.

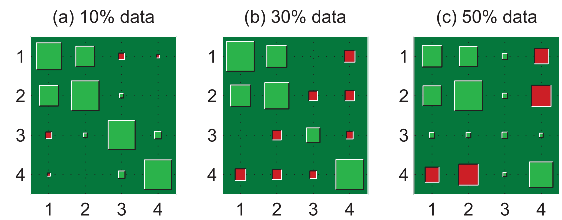

Fig. 14. Hinton diagrams of $\mathbf{\Omega}$ of DMTRC$_l$ on the sentiment data set.

tasks). For a convenient comparison with the supervised MTFL and MTRL, we adopt the same experimental setting as [19]. Specifically, the data set in use is a postprocessed version that aims to classify the reviews of some products to two classes: positive or negative reviews. It contains four binary-class tasks: books, DVDs, electronics, and kitchen appliances. Each task contains 2,000 observations, in which 1,000 reviews are labeled as positive and the other 1,000 as negative. Each observation is a vector with 4,73,853 dimensions.Note that we discarded three features that contain unrecognized characters. We generate three MTC problems by randomly selecting 10, 30, and 50 percent of the data from each task. Other experimental settings are the same as those on the 20-newsgroups data set.

Fig. 12 gives the NMI comparison. The experimental phenomena are quite similar with those on the 20-newsgroups data set. The only difference is that when more data is available and when PCA is used to project the high dimensional data set to a low dimensional space, the clustering algorithms are generally getting better on the sentiment data set while the algorithms are getting worse on the 20-newsgroups data set. This might be caused by the difficulties of the data sets. That is to say, projecting the data to 100 dimensional subspace is enough to catch the useful information on the sentiment data set while doing so is not enough on the 20-newsgroups data set. To support this explanation, we visualize $\mathbf{D}$ of DMTFC$_l$ in Fig. 13 and compare it with the visualizations of $\mathbf{D}$ in Fig. 8. We can see that the filters $\mathbf{D}$ on the sentiment set are more effective than those on the 20-newsgroups set.

We provide the hinton diagrams of $\mathbf{\Omega}$ of DMTRC$_l$ and *DMTRC$_l$ in Figs. 14 and 15. We further provide the

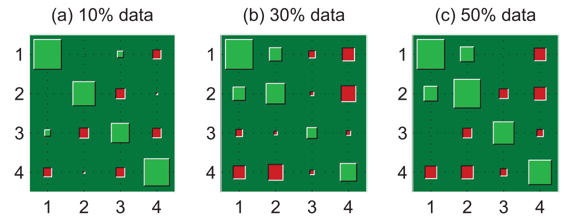

Fig. 15. Hinton diagrams of Ω of *DMTRC$_l$ on the sentiment data set.

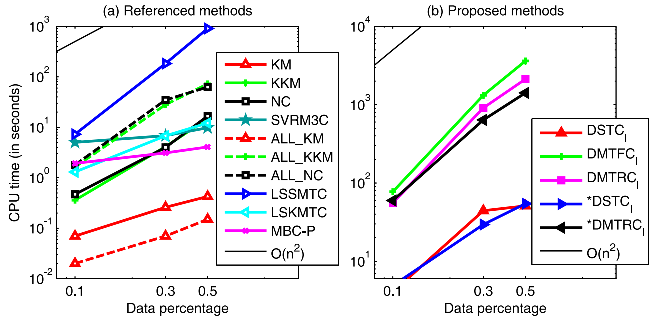

Fig. 17. CPU time comparison on the sentiment data set.

performance of the proposed algorithms on the individual tasks in Fig. 16. The experimental phenomena in Fig. 16 are consistent with those in Fig. 12 and are comparable with those yielded by the supervised counterparts of the proposed clusterings, i.e. MTFL and MTRL (see [19, Section 4.3]). Finally, we list the running time of the methods in Fig. 17. The results are consistent with the results in Fig. 11.

11 Conclusions and Future Work

In this paper, we have proposed a novel Bayesian DMTC framework. Within the framework, we have implemented two multiclass DMTC objectives by specifying the framework with four assumptions. The first one, named DMTFC, works under the multivariate Gaussian prior that models a shared feature representation across tasks, while the second one, named DMTRC, models the task relationship. Both objectives are formulated as difficult MIP problems. We have further relaxed the MIP problems to convex optimization problems and solve the relaxed problems efficiently in a uniform alternating optimization procedure. Technically, the two convex DMTC algorithms can be seen as the objective combination of the supervised MTFL/MTRL and the unsupervised SVRM3C. Experimental comparison with seven STC algorithms as well as three state-of-the-art MTC algorithms on the pendigits, multi-domain newsgroups and multi-domain sentiment data sets demonstrated the effectiveness of the proposed algorithms.

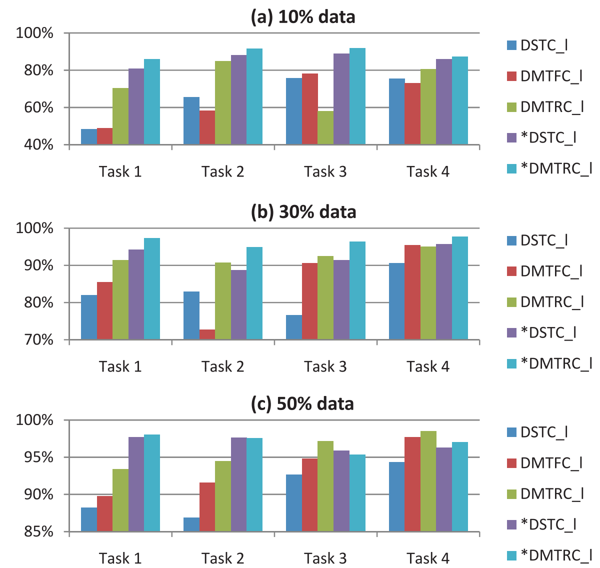

Fig. 16. NMI comparison between the proposed methods on the individual tasks with respect to different percentages of data on the sentiment data set.

Acknowledgments

The author would like to thank Prof. DeLiang Wang for providing the Ohio Supercomputing Center, Columbus, OH, for the experimental running. This work was supported by the China Postdoctoral Science Foundation funded project under Grant 2012M520278.

References

[1] R. Caruana, “Multitask learning,” *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.
[2] S. Thrun and L. Pratt, “Learning to learn: Introduction and overview,” in *Learning to Learn*. Norwell, MA, USA: Kluwer, 1998, pp. 3–17.
[3] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[4] B. Bakker and T. Heskes, “Task clustering and gating for Bayesian multitask learning,” *J. Mach. Learn. Res.*, vol. 4, pp. 83–99, 2003.
[5] Y. Bengio, “Deep learning of representations for unsupervised and transfer learning,” in *Proc. ICML Workshop Unsupervised Transfer Learn.*, 2011, vol. 7, pp. 1–20.
[6] R. K. Ando and T. Zhang, “A framework for learning predictive structures from multiple tasks and unlabeled data,” *J. Mach. Learn. Res.*, vol. 6, pp. 1817–1853, 2005.
[7] A. Argyriou, T. Evgeniou, and M. Pontil, “Multi-task feature learning,” in *Proc. Adv. Neural Inform. Process. Syst.*, 2007, vol. 19, pp. 41–49.
[8] A. Argyriou, T. Evgeniou, and M. Pontil, “Convex multi-task feature learning,” *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, 2008.
[9] J. Liu, S. Ji, and J. Ye, “Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization,” in *Proc. 25th Conf. Uncertain. Artif. Intell.*, 2009, pp. 339–348.
[10] Y. Zhang, D. Y. Yeung, and Q. Xu, “Probabilistic multi-task feature selection,” in *Proc. Adv. Neural Inform. Process. Syst.*, 2010, vol. 23, pp. 2559–2567.
[11] J. Chen, L. Tang, J. Liu, and J. Ye, “A convex formulation for learning shared structures from multiple tasks,” in *Proc. 26th Int. Conf. Mach. Learn.*, 2009, pp. 137–144.
[12] J. Chen, L. Tang, J. Liu, and J. Ye, “A convex formulation for learning a shared predictive structure from multiple tasks,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1025–1038, May 2013.
[13] N. D. Lawrence and J. C. Platt, “Learning to learn with the informative vector machine,” in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, pp. 65–73.
[14] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, “Multi-task learning for classification with dirichlet process priors,” *J. Mach. Learn. Res.*, vol. 8, pp. 35–63, 2007.
[15] T. Evgeniou, M. Pontil, and O. Toubia, “A convex optimization approach to modeling consumer heterogeneity in conjoint estimation,” *Mark. Sci.*, vol. 26, no. 6, pp. 805–818, 2007.
[16] Q. Liu, X. Liao, H. Li, J. Stack, and L. Carin, “Semisupervised multitask learning,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1074–1086, Jun. 2009.

[17] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2004, pp. 109–117.

[18] T. Evgeniou, C. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *J. Mach. Learn. Res.*, vol. 6, no. 1, pp. 615–637, 2006.

[19] Y. Zhang and D. Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," in *Proc. 26th Conf. Uncertain. Artif. Intell.*, 2010, pp. 733–742.

[20] L. Jacob, F. Bach, and J. P. Vert, "Clustered multi-task learning: A convex formulation," in *Proc. Adv. Neural Inform. Process. Syst.*, vol. 25, pp. 1–8, 2009.

[21] J. Zhou, J. Chen, and J. Ye, "Clustered multi-task learning via alternating structure optimization," in *Proc. Adv. Neural Inform. Process. Syst.*, 2011, vol. 25, pp. 702–710.

[22] B. Romera Paredes, A. Argyriou, N. Bianchi-Berthouze, and M. Pontil, "Exploiting unrelated tasks in multi-task learning," in *Proc. 15th Int. Conf. Aritif. Intell. Stat.*, 2012, pp. 951–959.

[23] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. 3rd ed. San Mateo, CA, USA: Morgan Kaufmann, 2011.

[24] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[25] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inform. Process. Syst.*, 2001, pp. 849–856.

[26] A. Y. Ng and A. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Proc. Adv. Neural Inform. Process. Syst.*, 2002, vol. 14, pp. 841–849.

[27] F. Bach and Z. Harchaoui, "Diffrac: A discriminative and flexible framework for clustering," in *Proc. Adv. Neural Inform. Process. Syst.*, 2007, vol. 20, pp. 1–8.

[28] J. Ye, Z. Zhao, and M. Wu, "Discriminative k-means for clustering," in *Proc. Adv. Neural Inform. Process. Syst.*, 2007, vol. 20, pp. 1649–1656.

[29] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *Proc. Adv. Neural Inform. Process. Syst.*, 2005, vol. 17, pp. 1537–1544.

[30] L. Xu and D. Schuurmans, "Unsupervised and semi-supervised multi-class support vector machines," in *Proc. 20th AAAI Conf. Artif. Intell.*, 2005, vol. 20, no. 2, pp. 904–910.

[31] K. Zhang, I. W. Tsang, and J. T. Kwok, "Maximum margin clustering made practical," *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 583–596, Apr. 2009.

[32] F. Wang, B. Zhao, and C. S. Zhang, "Linear time maximum margin clustering," *IEEE Trans. Neural Netw.*, vol. 21, no. 2, pp. 319–332, Feb. 2010.

[33] Y. F. Li, I. W. Tsang, J. T. Kwok, and Z. H. Zhou, "Tighter and convex maximum margin clustering," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, Clearwater Beach, FL, USA, 2009, pp. 344–351.

[34] X. L. Zhang and J. Wu, "Linearithmic time sparse and convex maximum margin clustering," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 6, pp. 1669–1692, Dec. 2012.

[35] R. Gomes, A. Krause, and P. Perona, "Discriminative clustering by regularized information maximization," in *Proc. Adv. Neural Inform. Process. Syst.*, 2010, vol. 23, pp. 775–783.

[36] L. Wang, X. Li, Z. Tu, and J. Jia, "Discriminative clustering via generative feature mapping," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 1–7.

[37] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Sharing clusters among related groups: Hierarchical dirichlet processes," in *Proc. Adv. Neural Inform. Process. Syst.*, 2005, pp. 1385–1392.

[38] W. Dai, Q. Yang, G. R. Xue, and Y. Yu, "Self-taught clustering," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 200–207.

[39] Q. Gu and J. Zhou, "Learning the shared subspace for multi-task clustering and transductive transfer classification," in *Proc. 9th IEEE Int. Conf. Data Min.*, 2009, pp. 159–168.

[40] J. Zhang and C. Zhang, "Multitask bregman clustering," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 655–660.

[41] J. Zhang and C. Zhang, "Multitask bregman clustering," *Neurocomputing*, vol. 74, no. 10, pp. 1720–1734, 2011.

[42] Q. Gu, Z. Li, and J. Han, "Learning a kernel for multi-task clustering," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 1–6.

[43] N. Thach, H. Shao, B. Tong, and E. Suzuki, "A compression-based dissimilarity measure for multi-task clustering," in *Proc. 19th Int. Conf. Found. Intell. Syst.*, 2011, pp. 123–132.

[44] S. Xie, H. Lu, and Y. He, "Multi-task co-clustering via nonnegative matrix factorization," in *Proc. 21st Int. Conf. Patt. Recogn.*, 2012, pp. 1–6.

[45] T. Huy, H. Shao, B. Tong, and E. Suzuki, "A feature-free and parameter-light multi-task clustering framework," *Knowl. Inform. Syst.*, vol. 36, no. 1, pp. 1–26, 2012.

[46] W. Jiang and F. Chung, "Transfer spectral clustering," in *Proc. Eur. Conf. Mach. Learn. Knowl. Disc. Databases*, pp. 789–803, 2012.

[47] Z. Zhang and J. Zhou, "Multi-task clustering via domain adaptation," *Patt. Recogn.*, vol. 45, no. 1, pp. 465–473, 2012.

[48] B. Kulis and M. I. Jordan, "Revisiting k-means: New algorithms via bayesian nonparametrics," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1–8.

[49] J. E. Kelley, "The cutting-plane method for solving convex programs," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 4, pp. 703–712, 1960.

[50] Z. Xu, R. Jin, I. King, and M. R. Lyu, "An extended level method for efficient multiple kernel learning," in *Proc. Adv. Neural Inform. Process. Syst.*, 2009, vol. 21, pp. 1825–1832.

[51] H. Yang, Z. Xu, J. Ye, I. King, and M. R. Lyu, "Efficient sparse generalized multiple kernel learning," *IEEE Trans. Neural Netw.*, vol. 2, no. 3, pp. 433–446, Mar. 2011.

[52] P. Gong, C. Zhang, Z. Lu, J. Huang, and J. Ye, "A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1–9.

[53] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[54] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, 2003.

[55] A. Topchy A. K. Jain, and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1866–1881, Dec. 2005.

[56] A. L. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 835–850, Jun. 2005.

[57] H. Wang, H. Shan, and A. Banerjee, "Bayesian cluster ensembles," in *Proc. 9th SIAM Int. Conf. Data Min.*, 2009, pp. 1–12.

[58] S. Vega-Pons and J. Ruiz-Shulcloper, "A survey of clustering ensemble algorithms," *Int. J. Pattern Recog. Artif. Intell.*, vol. 25, no. 03, pp. 337–372, 2011.

[59] R. Gomes, P. Welinder, A. Krause, and P. Perona, "Crowdclustering," *Adv. Neural Inform. Process. Syst.*, pp. 558–566, 2011.

[60] J. Yi, T. Yang, R. Jin, A. K. Jain, and M. Mahdavi, "Robust ensemble clustering by matrix completion," in *Proc. 12th IEEE Int. Conf. Data Min.*, 2012, pp. 1176–1181.

[61] J. Yi, R. Jin, A. K. Jain, and S. Jain, "Crowdclustering with sparse pairwise labels: A matrix completion approach," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 1–7.

[62] P. Wang, K. B. Laskey, C. Domeniconi, and M. I. Jordan, "Nonparametric bayesian co-clustering ensembles," in *Proc. SIAM Int. Conf. Data Min.*, 2011, pp. 1–10.

[63] A. Argyriou, C. A. Micchelli, and M. Pontil, "When is there a representer theorem? vector versus matrix regularizers," *J. Mach. Learn. Res.*, vol. 10, pp. 2507–2529, 2009.

[64] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

Xiao-Lei Zhang (S'08-M'12) received the PhD degree in Information and Communication Engineering from Tsinghua University, Beijing, China, in 2012. From 2013 to 2014, he was a visiting scholar with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH. His current research interests are the topics on machine learning, computational audition, statistical signal processing, and bioinformatics. He also had some research interests on information theory and computer networks.