# Supplementary Material of the Paper: Convex Discriminative Multitask Clustering

Xiao-Lei Zhang, *Member, IEEE*

**Abstract**

In this supplementary material, we first give the detailed derivation of equations (6) and (11) in the main paper in Section 1 and Section 2 respectively. Then, we present the optimization procedure of DMTC in detail in Section 3. In the rest of the supplement, we will report the experimental results in detail in four evaluation metrics – Accuracy (ACC), Area Under the ROC Curves (AUC), Adjusted Rand Index (ARI), and Normalized Mutual Information (NMI).

## 1 Derivation of Equation (6) in the Main Paper

We can get the Lagrange of the equation (5) in the main paper as follows:

$$\begin{aligned}&\mathcal{L}(\mathbf{W}_c;\xi^i_{j,c};\alpha^i_{j,c})\\ =&\frac{\lambda_1}{2}\mathrm{tr}(\mathbf{W}_c^T\mathbf{W}_c)+\frac{\lambda_2}{2}\mathrm{tr}(\mathbf{W}_c^T\mathbf{D}^{-1}\mathbf{W}_c)+\sum_{i=1}^{m}\frac{1}{n_i}\sum_{j=1}^{n_i}(\xi^i_{j,c})^2+\sum_{i=1}^{m}\sum_{j=1}^{n_i}\alpha^i_{j,c}\left(\bar{y}^i_{j,c}-\mathbf{w}^T_{i,c}\mathbf{x}^i_j-\xi^i_{j,c}\right)\\ =&\frac{1}{2}\mathrm{tr}(\mathbf{W}_c^T\mathbf{D}^{-1}(\lambda_1\mathbf{D}+\lambda_2\mathbf{I}_d)\mathbf{W}_c)-\sum_{i=1}^{m}\sum_{j=1}^{n_i}\alpha^i_{j,c}\mathbf{w}^T_{i,c}\mathbf{x}^i_j+\sum_{i=1}^{m}\sum_{j=1}^{n_i}\alpha^i_{j,c}\bar{y}^i_{j,c}+\sum_{i=1}^{m}\sum_{j=1}^{n_i}\left(\frac{1}{n_i}(\xi^i_{j,c})^2-\alpha^i_{j,c}\xi^i_{j,c}\right)\end{aligned}\tag{1}$$

where $\alpha^i_{j,c}$ is the Lagrangian variable, and $\mathbf{I}_d$ is a $d$-dimensional identity matrix. Calculating the partial derivatives of problem (1) over $\mathbf{W}_c$ and $\xi^i_{j,c}$ and letting the derivatives equal to 0 can get the following two equations:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{W}_c}=\mathbf{D}^{-1}(\lambda_1\mathbf{D}+\lambda_2\mathbf{I}_d)\mathbf{W}_{i,c}-\sum_{i=1}^{m}\sum_{j=1}^{n_i}\alpha^i_{j,c}\mathbf{x}^i_j\mathbf{e}_i^T=0\tag{2}$$

$$\frac{\partial\mathcal{L}}{\partial\xi^i_{j,c}}=\frac{2}{n_i}\xi^i_{j,c}-\alpha^i_{j,c}=0\tag{3}$$

From the above equations, we can get:

$$\mathbf{W}_c=\sum_{i=1}^{m}\sum_{j=1}^{n_i}\alpha^i_{j,c}\mathbf{D}(\lambda_1\mathbf{D}+\lambda_2\mathbf{I}_d)^{-1}\mathbf{x}^i_j\mathbf{e}_i^T\tag{4}$$

$$\xi^i_{j,c}=\frac{n_i}{2}\alpha^i_{j,c}\tag{5}$$

Substituting (4) and (5) to (1) can derive the following maximization problem

$$\max_{\boldsymbol{\alpha}_c}\sum_{i=1}^{m}\sum_{j=1}^{n_i}\alpha^i_{j,c}\bar{y}^i_{j,c}-\frac{1}{2}\sum_{i_1=1}^{m}\sum_{i_2=1}^{m}\sum_{j_1=1}^{n_i}\sum_{j_2=1}^{n_i}\alpha^{i_1}_{j_1,c}{\mathbf{x}^{i_1}_{j_1}}^T\mathbf{D}\left(\lambda_1\mathbf{D}+\lambda_2\mathbf{I}_d\right)^{-1}\mathbf{x}^{i_2}_{j_2}\alpha^{i_2}_{j_2,c}\left\langle\mathbf{e}_{i_1},\mathbf{e}_{i_2}\right\rangle-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n_i}\frac{n_i}{2}(\alpha^i_{j,c})^2\tag{6}$$

After making the denotations that $\boldsymbol{\alpha}_c=[\alpha^1_{1,c},\ldots,\alpha^m_{n_m,c}]^T$, $K_{\mathrm{F}}\left(\mathbf{x}^{i_1}_{j_1},\mathbf{x}^{i_2}_{j_2}\right)={\mathbf{x}^{i_1}_{j_1}}^T\mathbf{D}(\lambda_1\mathbf{D}+\lambda_2\mathbf{I}_d)^{-1}\mathbf{x}^{i_2}_{j_2}\left\langle\mathbf{e}_{i_1},\mathbf{e}_{i_2}\right\rangle$, and $\boldsymbol{\Lambda}$ is the diagonal matrix whose diagonal element equals to $n_i$ if the corresponding observation belongs to the $i$-th task, we can write problem (6) briefly as follows:

$$\max_{\boldsymbol{\alpha}_c}\sum_{i=1}^{m}\sum_{j=1}^{n_i}\alpha^i_{j,c}\bar{y}^i_{j,c}-\frac{1}{2}{\boldsymbol{\alpha}_c}^T\left(\mathbf{K}_{\mathrm{F}}+\frac{1}{2}\boldsymbol{\Lambda}\right)\boldsymbol{\alpha}_c\tag{7}$$

- *Xiao-Lei Zhang was with Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing, China, 100084.*
  *E-mail: huoshan6@126.com*

*This work was finished when the author was a visiting scholar with the Department of Computer Science and Engineering, Ohio State University, OH, USA, 43210.*

## 2 Derivation of Equation (11) in the Main Paper

Again, we get the Lagrange of the problem in the braces of equation (10) in the main paper as follows:

$$\mathcal{L}(\{\theta_i\}_{i=1}^m)=\sum_{i=1}^m \theta_i - \frac{1}{2}\sum_{c=1}^C \boldsymbol{\alpha}_c^T \widetilde{\mathbf{K}}_{\mathrm{F}} \boldsymbol{\alpha}_c - \sum_{i=1}^m \sum_{k:\mathbf{Y}_k^i \in \mathcal{B}^i} \mu_k^i \left(\theta_i - \sum_{c=1}^C \sum_{j=1}^{n_i} \alpha_{j,c}^i y_{j,c}^i\right) \tag{8}$$

where $\mu_k^i \geq 0$ is the Lagrange variable. Calculating the partial derivative of problem (8) with respect to $\theta$ and letting the derivative equals to 0 can get the following equation:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = 1 - \sum_{k:\mathbf{Y}_k^i \in \mathcal{B}^i} \mu_k^i = 0 \tag{9}$$

We get

$$\sum_{k:\mathbf{Y}_k^i \in \mathcal{B}^i} \mu_k^i = 1 \tag{10}$$

Substituting (10) to (8) can derive equation (11) in the main paper.

## 3 Optimization Procedure

In this section, we will present the optimization procedure in detail. The readers can directly use the content of this section to replace Section 6 of the main paper.

We are to solve DMTFC (equation (10) in the main paper) and DMTRC (equation (16) in the main paper) in a uniform framework. To facilitate the mathematical representation, we write them as the following uniform objective:

$$\begin{aligned}\max_{\{\boldsymbol{\alpha}_c\}_{c=1}^C} \min_{\mathbf{Z}\in\mathcal{Z}} \Bigg\{ & \max_{\{\theta_i\}_{i=1}^m} \sum_{i=1}^m \theta_i - \frac{1}{2}\sum_{c=1}^C \boldsymbol{\alpha}_c^T \widetilde{\mathbf{K}} \boldsymbol{\alpha}_c \\ & \text{s.t.} \theta_i \leq \sum_{c=1}^C \sum_{j=1}^{n_i} \alpha_{j,c}^i y_{j,c}^i, \forall i = 1,\ldots,m, \forall k : \mathbf{Y}_k^i \in \mathcal{B}^i \Bigg\}.\end{aligned} \tag{11}$$

where $\mathbf{Z}$ stands for $\mathbf{D}$ or $\boldsymbol{\Omega}$, $\mathcal{Z}$ stands for $\mathcal{D}$ or $\mathcal{A}$, and $\widetilde{\mathbf{K}}$ stands for $\widetilde{\mathbf{K}}_{\mathrm{F}}$ or $\widetilde{\mathbf{K}}_{\mathrm{R}}$.

The solution framework is an alternating method. First, it decomposes the unsupervised problem (11) to a serial supervised multiclass MTL problem by the cutting-plane algorithm (CPA) [1] and the extended level method (ELM) [2], [3], where the decomposition algorithm can be seen as a multitask extension of the SVR-M3C algorithm [4]. 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 3.3.

### 3.1 Optimizing (11) Via Cutting Plane Algorithm

Because the number of the constraints in problem (11) is exponential large with respect to $n$, directly optimizing (11) is impossible when the data set contains over dozens of examples. Hence, we adopt the cutting-plane algorithm [1] to solve problem (11) approximately.

We present the key idea of the cutting-plane algorithm as follows. Generally, given a constrained optimization problem, the cutting plane algorithm alternates the following two steps until the objective value converges. The first step is to solve a reduced problem of the constrained problem, i.e. a problem that contains only a part of the constraints. The second step is to find the most violated constraint of the reduced problem, and add it to the constraint set so as to form a new reduced problem for the next iteration. It has been proved that the number of the cutting-plane iterations is upper bounded by $\mathcal{O}(1/\epsilon)$ [5], where $\epsilon$ is a user defined cutting-plane solution precision.

For problem (11), the cutting-plane algorithm iterates the following two steps:

a) Solving the following reduced problem of problem (11):

$$\begin{aligned}\max_{\{\boldsymbol{\alpha}_c\}_{c=1}^C} \min_{\mathbf{Z}\in\mathcal{Z}} \Bigg\{ & \max_{\{\theta_i\}_{i=1}^m} \sum_{i=1}^m \theta_i - \frac{1}{2}\sum_{c=1}^C \boldsymbol{\alpha}_c^T \widetilde{\mathbf{K}} \boldsymbol{\alpha}_c \\ & \text{s.t.} \theta_i \leq \sum_{c=1}^C \sum_{j=1}^{n_i} \alpha_{j,c}^i y_{j,c}^i, \forall i = 1,\ldots,m, \forall k : \mathbf{Y}_k^i \in \mathcal{Y}^i \Bigg\}.\end{aligned} \tag{12}$$

where $\mathcal{Y}^i$ is the reduced constraint subset of $\mathcal{B}^i$. Problem (12) is equivalent to

$$\max_{\{\boldsymbol{\alpha}_c\}_{c=1}^C} \min_{\mathbf{Z}\in\mathcal{Z}} \max_{\{\theta_i\}_{i=1}^m} \sum_{i=1}^m \theta_i - \frac{1}{2}\sum_{c=1}^C \boldsymbol{\alpha}_c^T \widetilde{\mathbf{K}} \boldsymbol{\alpha}_c \tag{13}$$
$$\text{s.t.} \theta_i \leq \sum_{c=1}^C \sum_{j=1}^{n_i} \alpha_{j,c}^i y_{j,c}^i, \forall i = 1,\ldots,m, \forall k : \mathbf{Y}_k^i \in \mathcal{Y}^i \Big\}.$$

where $\mathcal{M}_{\mathcal{Y}}^i = \Big\{\boldsymbol{\mu}^i | 0 \leq \mu_k^i \leq 1, \sum_{k=1}^{|\mathcal{Y}^i|} \mu_k^i = 1\Big\}$ with $|\mathcal{Y}^i|$ denoted as the size of $\mathcal{Y}^i$. Here, we leave this complicated problem to Section 3.2.

b) Calculating the most violated constraint $\left\{\mathbf{Y}_{|\mathcal{Y}^i|+1}^i\right\}_{i=1}^m$ by solving the following problem and adding $\mathbf{Y}_{|\mathcal{Y}^i|+1}^i$ to $\mathcal{Y}^i$.

$$\min_{\left\{\mathbf{Y}_{|\mathcal{Y}^i|+1}^i\right\}_{i=1}^m} \sum_{i=1}^m \sum_{c=1}^C \sum_{j=1}^{n_i} \alpha_{j,c}^i \bar{y}_{|\mathcal{Y}^i|+1,j,c}^i. \tag{14}$$

We can observe that the subitems $\sum_c \sum_j \alpha_{j,c}^i \bar{y}_{|\mathcal{Y}^i|+1,j,c}^i$ are mutually independent with respect to $i$. Therefore, optimizing problem (14) is equivalent to optimizing the summation of the following problems:

$$\max_{\mathbf{Y}_{|\mathcal{Y}^i|+1}^i} \sum_{c=1}^C \sum_{j=1}^{n_i} \alpha_{j,c}^i \bar{y}_{|\mathcal{Y}^i|+1,j,c}^i \ , \ \forall i = 1,\ldots,m. \tag{15}$$

Although the above problem is a binary integer matrix optimization problem, it can be solved in time $\mathcal{O}\left(\sum_{i=1}^m Cn_i \log(Cn_i)\right)$ thanks to the constraints of $\mathbf{Y}^i$ in equation (3) of the main paper. See [4, Algorithm 6] for the efficient algorithm.

### 3.2 Optimizing (13) Via Extended Level Method

Like the full problem (11), the cutting-plane subproblem (13) also has an equivalent form:

$$\max_{\{\boldsymbol{\alpha}_c\}_{c=1}^C} \min_{\mathbf{Z}\in\mathcal{Z}} \min_{\{\boldsymbol{\mu}^i\in\mathcal{M}_{\mathcal{Y}}^i\}_{i=1}^m} \sum_{i=1}^m \sum_{c=1}^C \sum_{j=1}^{n_i} \alpha_{j,c}^i \sum_{k:\mathbf{Y}_k^i\in\mathcal{Y}^i} \mu_k^i \bar{y}_{k,j,c}^i - \frac{1}{2}\sum_{c=1}^C \boldsymbol{\alpha}_c^T \widetilde{\mathbf{K}} \boldsymbol{\alpha}_c \tag{16}$$

Problem (13) 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 the efficient Extended Level Method (ELM) [2], [3].

We present the key idea of the ELM algorithm as follows. ELM tries to solve the concave-convex optimization problem $\min_a \max_b f(a,b)$ that is convex on $a$ and concave on $b$ by iteratively constructing tighter upper and lower bounds for the optimal objective value $f(a^\star;b^\star)$, where $(a^\star,b^\star)$ denotes the optimal solution. Specifically, it iterates the following two steps. The first step is to construct the lower bound $\underline{f}_s = \min_a \max_{1\leq r\leq s} f(a;b_r)$ and the upper bound $\overline{f}^s = \min_{1\leq r\leq s} f(a_r;b_r)$ of $f$, where $r$ and $s$ denotes the indices of the iterations (i.e. solutions) and $\max_{1\leq r\leq s} f(a;b_r)$ is also a cutting-plane model. The second step is to first get $a_{s+1}$ by solving the following optimization problem

$$\min_{a_{s+1}} \|a_{s+1} - a_s\|^2 \tag{17}$$
$$\text{s.t. } f(a_{s+1};b_r) \leq \tau\overline{f}_s + (1-\tau)\underline{f}_s, \forall r = 1,\ldots,s,$$

and then get $b_{s+1}$ by solving $\max_{b_{s+1}} f(a_{s+1},b_{s+1})$, where $\tau$ is a user defined constant. equation (17) performs like a regularizer that prevents $a_{s+1}$ far from $a_s$.

For problem (16), because optimizing $\boldsymbol{\mu}$ and $\mathbf{Z}$ jointly is difficult, setting $a = \{\boldsymbol{\mu},\mathbf{Z}\}$ is improper. We propose to set $a = \boldsymbol{\mu}$ and optimize $\mathbf{Z}$ and $\boldsymbol{\alpha}$ jointly. It is easy to prove the correctness of this new optimization strategy. The proof is similar with the proof of [2, Theorem 1]. Another very important issue is that to make the cutting-plane algorithm presented in Section 3.1 converges, for problem (16) at the $S$-th cutting-plane iteration, we should inherit all previous $S-1$ ELM models to initialize the upper and lower bounds of the problem, otherwise, the cutting-plane algorithm will fail. The proof is the same as [4, Theorem 3].

With the aforementioned two key points, the ELM algorithm for problem (13) is presented as follows. Suppose we are currently at the $S$-th cutting-plane subproblem. That is to say, we are to solve the $S$-th problem (13). Suppose solving the $R$-th cutting-plane subproblem yields $T_R$ ELM solutions, denoted as $\left\{\left\{\boldsymbol{\alpha}_{c,r}^R\right\}_{c=1}^C, \mathbf{Z}_r^R, \left\{\boldsymbol{\mu}_r^{i,R}\right\}_{i=1}^m\right\}_{r=1}^{T_R}$, where $R = 1,\ldots,S-1$. Suppose the constraint set of the $i$-th task at the $R$-th cutting plane iteration, denoted as $\mathcal{Y}^{i,R}$, contains $\left|\mathcal{Y}^{i,R}\right|$ constraints ($\left|\mathcal{Y}^{i,R}\right| \leq R$).

**Initialization of ELM.** All previous $S-1$ ELM models should be inherited by adding $\boldsymbol{\mu}_r^{i,R}$ with $\left|\mathcal{Y}^{i,S}\right| - \left|\mathcal{Y}^{i,R}\right|$ zeros: $\boldsymbol{\mu}_r^{i,R'} = \left[\left(\boldsymbol{\mu}_r^{i,R}\right)^T, \mathbf{0}^T_{\left|\mathcal{Y}^{i,S}\right|-\left|\mathcal{Y}^{i,R}\right|}\right]^T, \forall i = 1,\ldots,m, \forall r = 1,\ldots,T_R$. Without lose of generality, we further denote all inherited ELM solutions as $\left\{\{\boldsymbol{\alpha}_c^r\}_{c=1}^C, \mathbf{Z}_r, \{\boldsymbol{\mu}^{r,i}\}_{i=1}^m\right\}_{r=1}^s = \left\{\left\{\left\{\boldsymbol{\alpha}_{c,r}^R\right\}_{c=1}^C, \mathbf{Z}_r^R, \left\{\boldsymbol{\mu}_r^{i,R'}\right\}_{i=1}^m\right\}_{r=1}^{T_R}\right\}_{R=1}^{S-1}, \forall i = 1,\ldots,m,$ where $s = \sum_{R=1}^{S-1} T_R$.

**ELM.** The ELM algorithm for (13) iterates the following steps:

a) Constructing the lower bound $\underline{h}_s$ and upper bound $\overline{h}_s$ by

$$\underline{h}_s = \min_{\{\boldsymbol{\mu}^i \in \mathcal{M}_{\mathcal{Y}}^i\}_{i=1}^m} \max_{1\le r\le s} \sum_{i=1}^m \sum_{c=1}^C \sum_{j=1}^{n_i} \alpha_{r,j,c}^i \sum_{k:\mathbf{Y}_k^i \in \mathcal{Y}^i} \mu_k^i \bar{y}_{k,j,c}^i - \frac{1}{2}\sum_{c=1}^C \boldsymbol{\alpha}_{r,c}^T \widetilde{\mathbf{K}}_r \boldsymbol{\alpha}_{r,c}, \tag{18}$$

$$\overline{h}_s^i = \min_{1\le r\le s} \sum_{i=1}^m \sum_{c=1}^C \sum_{j=1}^{n_i} \alpha_{r,j,c}^i \sum_{k:\mathbf{Y}_k^i \in \mathcal{Y}^i} \mu_{r,k}^i \bar{y}_{k,j,c}^i - \frac{1}{2}\sum_{c=1}^C \boldsymbol{\alpha}_{r,c}^T \widetilde{\mathbf{K}}_r \boldsymbol{\alpha}_{r,c} \tag{19}$$

where $\widetilde{\mathbf{K}}_r$ is a function of $\mathbf{Z}_r$. With $\mathbf{Z}_r$ fixed, the tasks are mutually independent, hence we can get the bounds of each task separately:

$$\underline{h}_s^i = \min_{\boldsymbol{\mu}^i \in \mathcal{M}_{\mathcal{Y}}^i} \max_{1\le r\le s} \sum_{c=1}^C \sum_{j=1}^{n_i} \alpha_{r,j,c}^i \sum_{k:\mathbf{Y}_k^i \in \mathcal{Y}^i} \mu_k^i \bar{y}_{k,j,c}^i, \tag{20}$$

$$\overline{h}_s^i = \min_{1\le r\le s} \sum_{c=1}^C \sum_{j=1}^{n_i} \alpha_{r,j,c}^i \sum_{k:\mathbf{Y}_k^i \in \mathcal{Y}^i} \mu_{r,k}^i \bar{y}_{k,j,c}^i \tag{21}$$

b) Get $\left\{\boldsymbol{\mu}_{s+1}^i\right\}_{i=1}^m$ by

$$\begin{aligned} \min_{\{\boldsymbol{\mu}_{s+1}^i \in \mathcal{M}_{\mathcal{Y}}^i\}_{i=1}^m} & \sum_{i=1}^m \|\boldsymbol{\mu}_{s+1}^i - \boldsymbol{\mu}_s^i\|^2 \\ \text{s.t.} & \sum_{i=1}^m \sum_{c=1}^C \sum_{j=1}^{n_i} \alpha_{r,j,c}^i \sum_{k:\mathbf{Y}_k^i \in \mathcal{Y}^i} \mu_{s+1,k}^i \bar{y}_{k,j,c}^i - \frac{1}{2}\sum_{c=1}^C \boldsymbol{\alpha}_{r,c}^T \widetilde{\mathbf{K}}_r \boldsymbol{\alpha}_{r,c} \le \tau \overline{h}_s + (1-\tau)\underline{h}_s, \quad \forall r = 1,\ldots,s. \end{aligned} \tag{22}$$

Similar with Step a), we replace problem (22) with the summation of the following problems

$$\begin{aligned} \min_{\boldsymbol{\mu}_{s+1}^i \in \mathcal{M}_{\mathcal{Y}}^i} & \left\|\boldsymbol{\mu}_{s+1}^i - \boldsymbol{\mu}_s^i\right\|^2 \\ \text{s.t.} & \sum_{i=1}^m \sum_{c=1}^C \sum_{j=1}^{n_i} \alpha_{r,j,c}^i \sum_{k:\mathbf{Y}_k^i \in \mathcal{Y}^i} \mu_{s+1,k}^i \bar{y}_{k,j,c}^i \le \tau \overline{h}_s^i + (1-\tau)\underline{h}_s^i, \quad \forall r = 1,\ldots,s. \end{aligned} \tag{23}$$

c) Get $(\boldsymbol{\alpha}_{s+1}, \mathbf{Z}_{s+1})$ by

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

Here, we leave problem (24) to Section 3.3.

### 3.3 Optimizing (24) Via the Alternating Method

Problem (24) is a supervised multiclass MTL problem. We adopt an alternating method that is similar with [6] for it.

Specifically, for each problem (24), the method iterates the following two steps:

a) Given fixed $\mathbf{Z}$, we aim to solve

$$\begin{aligned} & \max_{\{\boldsymbol{\alpha}_c\}_{c=1}^C} \sum_{i=1}^m \sum_{c=1}^C \sum_{j=1}^{n_i} \alpha_{j,c}^i \sum_{k:\mathbf{Y}_k^i \in \mathcal{Y}^i} \mu_k^i \bar{y}_{k,j,c}^i - \frac{1}{2}\sum_{c=1}^C \boldsymbol{\alpha}_c^T \widetilde{\mathbf{K}} \boldsymbol{\alpha}_c \\ =& \sum_{c=1}^C \max_{\boldsymbol{\alpha}_c} \sum_{i=1}^m \sum_{j=1}^{n_i} \alpha_{j,c}^i \sum_{k:\mathbf{Y}_k^i \in \mathcal{Y}^i} \mu_k^i \bar{y}_{k,j,c}^i - \frac{1}{2}\boldsymbol{\alpha}_c^T \widetilde{\mathbf{K}} \boldsymbol{\alpha}_c \end{aligned} \tag{25}$$

When $\mathbf{Z}$ is fixed, the subitems in the right side of equation (25) are mutually independently. Hence, we solve each item

$$\max_{\boldsymbol{\alpha}_c} \sum_{i=1}^{m}\sum_{j=1}^{n_i} \alpha_{j,c}^{i} \sum_{k:\mathbf{Y}_k^i \in \mathcal{Y}^i} \mu_k^i \bar{y}_{k,j,c}^i - \frac{1}{2}\boldsymbol{\alpha}_c^T \widetilde{\mathbf{K}} \boldsymbol{\alpha}_c, \quad \forall c = 1, \ldots, C \tag{26}$$

independently, which is a supervised regression problem.

b) Given fixed $\{\boldsymbol{\alpha}_c\}_{c=1}^{C}$, we aim to optimize

$$\min_{\mathbf{Z}\in\mathcal{Z}} \sum_{i=1}^{m}\sum_{c=1}^{C}\sum_{j=1}^{n_i} \alpha_{j,c}^{i} \sum_{k:\mathbf{Y}_k^i \in \mathcal{Y}^i} \mu_k^i \bar{y}_{k,j,c}^i - \frac{1}{2}\sum_{c=1}^{C} \boldsymbol{\alpha}_c^T \widetilde{\mathbf{K}} \boldsymbol{\alpha}_c. \tag{27}$$

For solving problems (26) and (27), DMTFC and DMTRC should be considered separately as follows:

*Specifying (26) and (27) as a part of DMTFC:* We replace $\mathbf{Z}$ and $\mathcal{Z}$ by $\mathbf{D}$ and $\mathcal{D}$ respectively in the equations. For (26), the multitask kernel $\widetilde{\mathbf{K}}$ should be specified by equation (7) in the main paper. For (27), 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) in the main paper. The derivation is analogous to [7, Appendix 1].

*Specifying (26) and (27) as a part of DMTRC:* We replace $\mathbf{Z}$ and $\mathcal{Z}$ by $\boldsymbol{\Omega}$ and $\mathcal{A}$ respectively in the equations. For (26), $\widetilde{\mathbf{K}}$ should be specified by equation (17) in the main paper. For (27), we can get the closed solution of $\boldsymbol{\Omega}$ as $\boldsymbol{\Omega} = \frac{\left(\sum_{c=1}^{C} \mathbf{w}_c^T \mathbf{w}_c\right)^{\frac{1}{2}}}{\mathrm{tr}\left(\left(\sum_{c=1}^{C} \mathbf{w}_c^T \mathbf{w}_c\right)^{\frac{1}{2}}\right)}$ where $\mathbf{W}_c$ is defined in (18) in the main paper. The derivation is analogous to [6, equation 13].

### 3.4 Overview of the Algorithm

Observing that the relaxed DMTFC (equation (10) in the main paper) and DMTRC (equation (16) in the main paper) are quite similar, we propose a unified convex optimization objective (11), which can be solved alternatively by combining several existing efficient algorithms [1]–[4], [8], [9]. The optimization procedure is summarized as Algorithm 1.

## 4 Supplement to the Results of the Pendigits Dataset

In this section, we will first give the average performance of all tasks in the metrics of Accuracy (ACC), Adjusted Rand Index (ARI), and Normalized Mutual Information (NMI) respectively. Then, we will give the performance of each individual task in different jobs in the metric of NMI.

### 4.1 Average Performance of All Tasks in Different Metrics

Figs. 1, 2, and 3 show the average performance on the Pendigits dataset in the metrics of NMI, ARI, and ACC respectively.

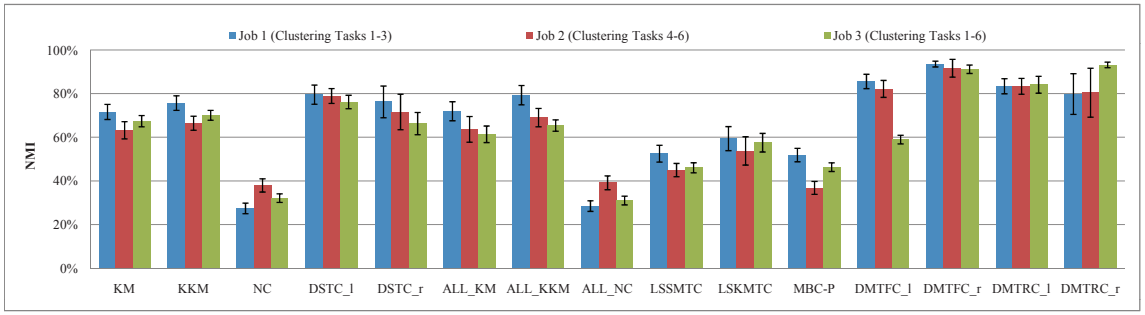


Fig. 1. Comparison of the average NMI of all tasks on the Pendigits dataset. The error bar denotes the standard deviation.

**Algorithm 1** Discriminative Multitask Clustering.

**Input:** The unlabeled observations from $m$ tasks $\left\{\{\mathbf{x}_j^i\}_{j=1}^{n_i}\right\}_{i=1}^m$, class number $P$, regularization parameters $\lambda_1$ and $\lambda_2$, and the parameters that control the class balance $\{l_i\}_{i=1}^m$

**Output:** $\hat{\mathbf{y}}$

**Initialization:** Arbitrary initial constraint label matrices $\left\{\mathbf{Y}_1^i\right\}_{i=1}^m$ that satisfy the constraints in equation (3) in the main paper, initial constraint sets $\mathcal{Y}^i \leftarrow \left\{\mathbf{Y}_1^i\right\}$ and $\mu_1^i \leftarrow 1$ with $i = 1, \ldots, m$

**repeat**
  Inherit the ELM solutions in the previous cutting-plane iterations
  **repeat**
    **for** task $i = 1, \ldots, m$ **do**
      Construct the lower bound via (20) and the upper bound via (21)
      Update $\boldsymbol{\mu}^i$ by solving (23)
    **end for**
    **repeat**
      **for** class $c = 1, \ldots, C$ **do**
        Update $\boldsymbol{\alpha}_c$ by solving (26)
      **end for**
      Update $\mathbf{Z}$ by solving (27)
    **until** the objective value converges
  **until** the objective value converges
  **for** task $i = 1, \ldots, m$ **do**
    Calculate the most violated constraint $\mathbf{Y}_{|\mathcal{C}^i|+1}^i$ of the cutting-plane algorithm by solving (15)
    $\mathcal{Y}^i \leftarrow \mathcal{Y}^i \cup \left\{\mathbf{Y}_{|\mathcal{Y}^i|+1}^i\right\}$
  **end for**
**until** the objective value converges or $\left\{\mathcal{Y}^i\right\}_{i=1}^m$ are unchanged
/*Prediction*/
**for** task $i = 1, \ldots, m$ **do**
  **for** $j = 1, \ldots, n_i$ **do**
    $\hat{y}_j^i \leftarrow \arg\max_p \sum_{u=1}^m \sum_{v=1}^{n_u} \alpha_{v,p}^u K_{MT}(\mathbf{x}_v^u, \mathbf{x}_j^i)$, where the kernel function $K_{MT}$ is defined in (7) in the main paper for DMTFC and in (17) in the main paper for DMTRC
  **end for**
**end for**

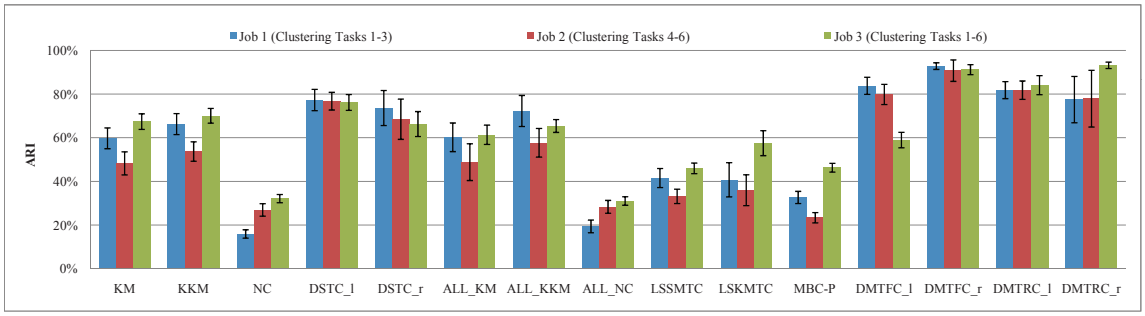


Fig. 2. Comparison of the average ARI of all tasks on the Pendigits dataset.

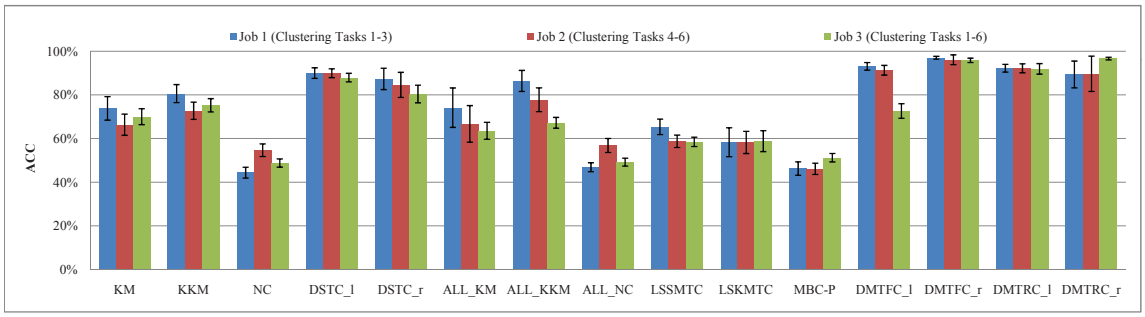


Fig. 3. Comparison of the average ACC of all tasks on the Pendigits dataset.

### 4.2 Performance of Each Individual Task in the Metric of NMI

Figs. 4, 5, and 6 show the NMIs of the clustering algorithms with respect to each individual task in Jobs 1, 2, and 3 respectively.

In this subsection, we should pay particular attention to the standard deviations of the NMIs on each individual task, since the standard deviations represent the stabilities of the clustering algorithms partially.

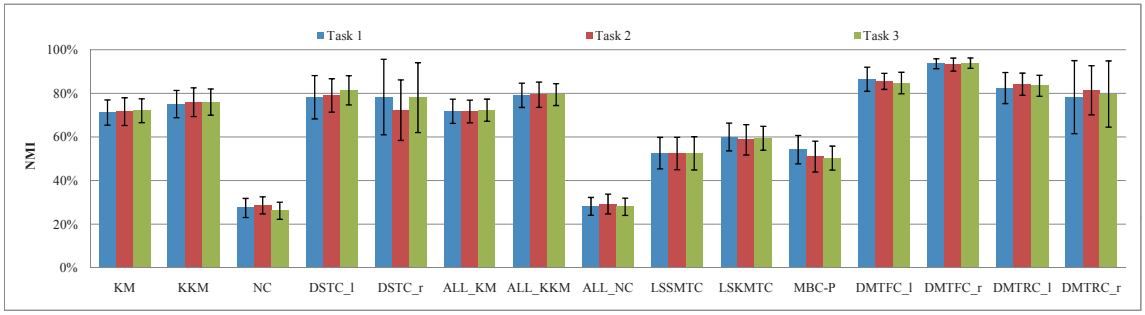


Fig. 4. NMI comparison with respect to each individual task in Job 1 on the Pendigits dataset.

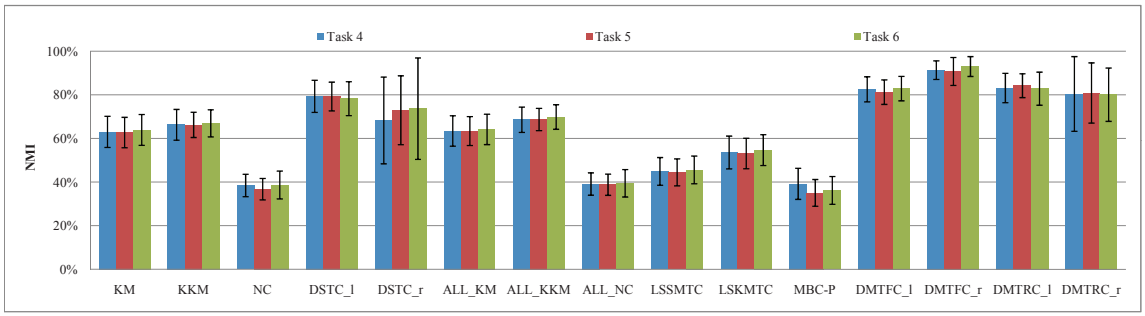


Fig. 5. NMI comparison with respect to each individual task in Job 2 on the Pendigits dataset.

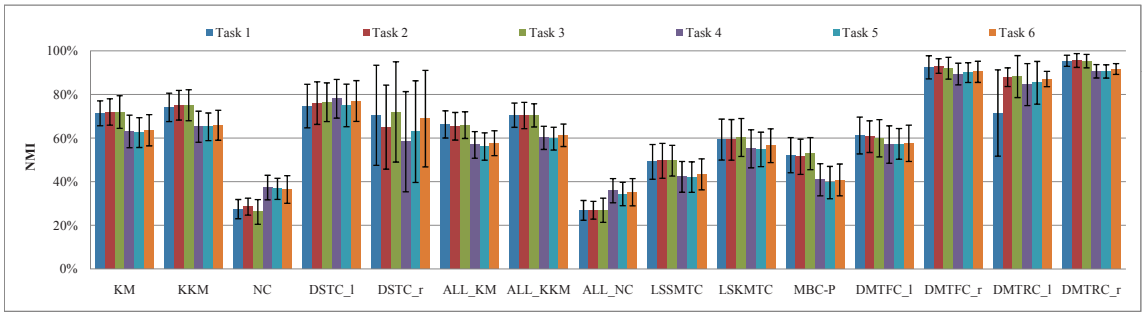


Fig. 6. NMI comparison with respect to each individual task in Job 3 on the Pendigits dataset.

## 5 Supplement to the Results of the Multi-Domain Newsgroups Dataset

In this section, we will first give the average performance of all tasks in the metrics of NMI, ARI, and ACC respectively. Then, we will give the performance of each individual task with 5%, 10%, 20% and 40% data of the 20-newsgroups dataset in the metric of NMI.

### 5.1 Average Performance of All Tasks in Different Metrics

Figs. 7, 8, and 9 show the average performance on the 20-newsgroups dataset in the metrics of NMI, ARI, and ACC respectively.

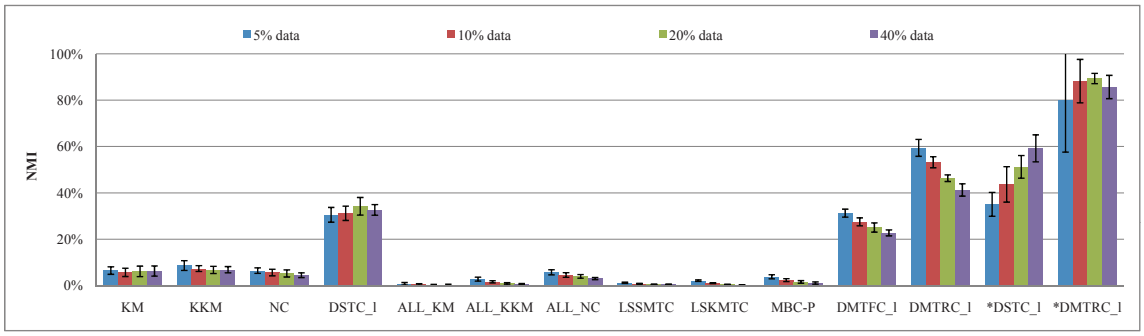


Fig. 7. Comparison of the average NMI of all tasks on the 20-Newgroups dataset.

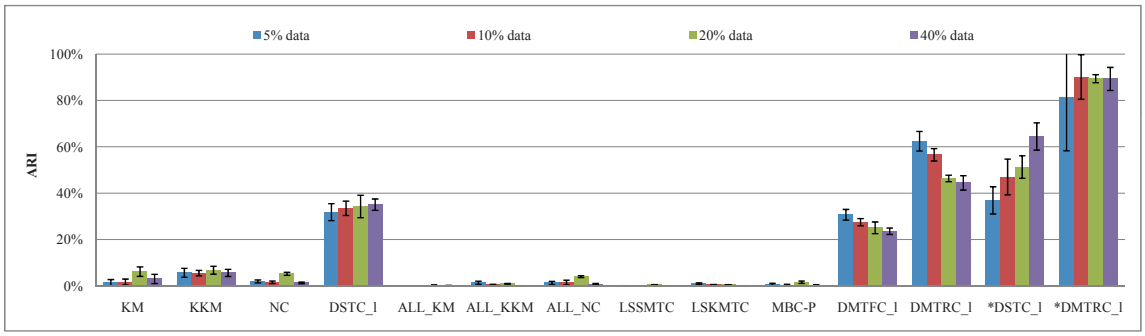


Fig. 8. Comparison of the average ARI of all tasks on the 20-Newgroups dataset.

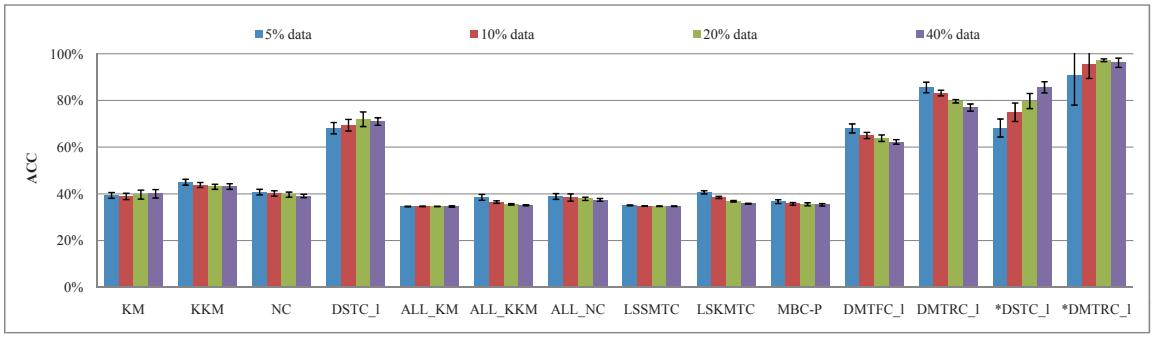


Fig. 9. Comparison of the average ACC of all tasks on the 20-Newgroups dataset.

### 5.2 Performance of Each Individual Task in the Metric of NMI

Figs. 10, 11, 12, and 13 show the NMIs of the clustering algorithms with 5%, 10%, 20% and 40% data of the 20-newsgroups dataset in the metric of NMI.

In this subsection, we should pay particular attention to the standard deviations of the NMIs on each individual task, since the standard deviations represent the stabilities of the clustering algorithms partially. From the figures, we can see that when more data is available, the stability of DMTRC is improved greatly.

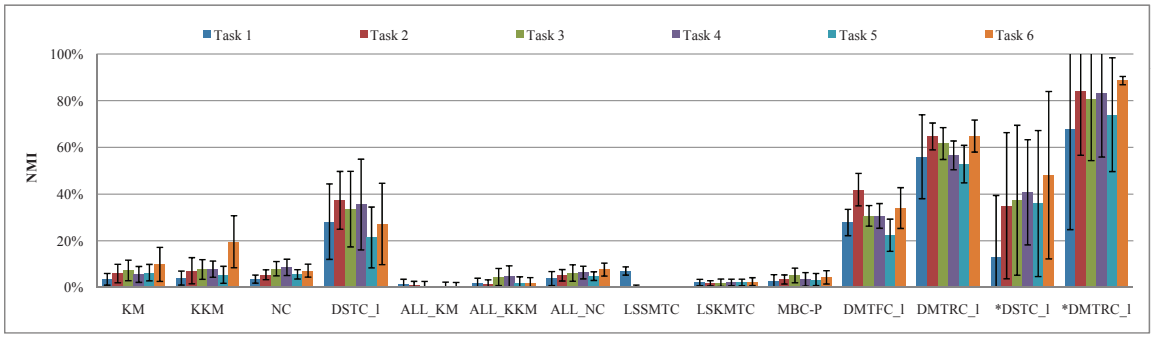


Fig. 10. NMI comparison with respect to each individual task on 5% data of the 20-Newgroups dataset.

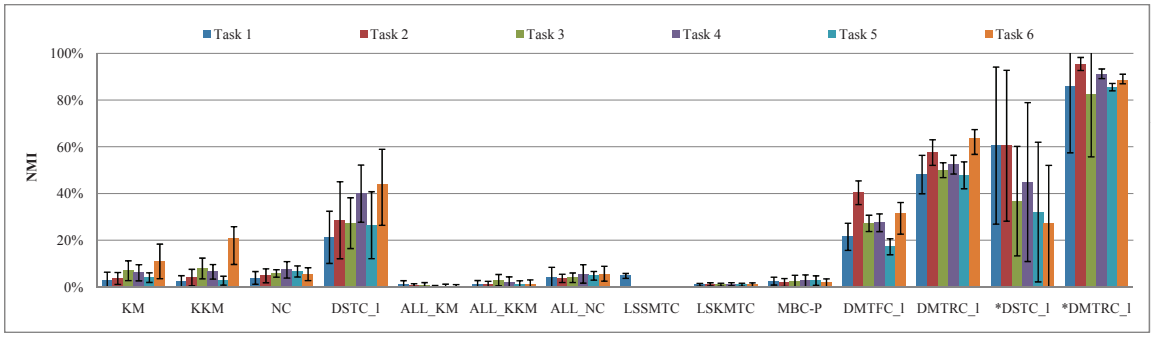


Fig. 11. NMI comparison with respect to each individual task on 10% data of the 20-Newgroups dataset.

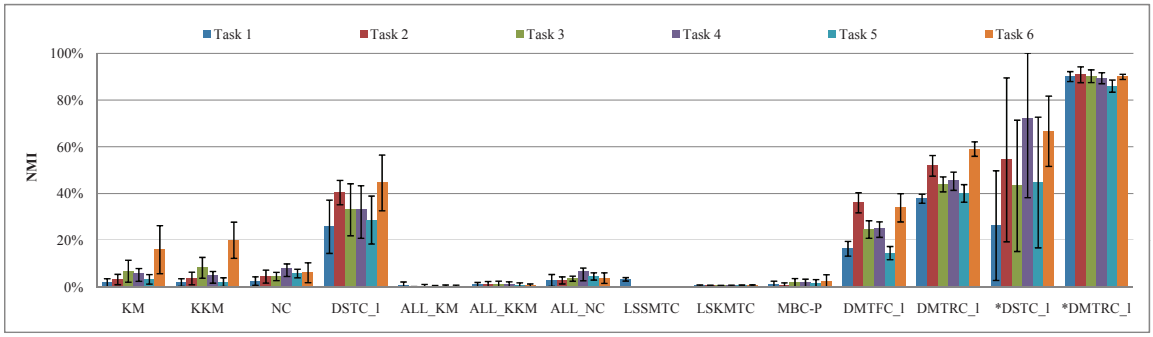


Fig. 12. NMI comparison with respect to each individual task on 20% data of the 20-Newgroups dataset.

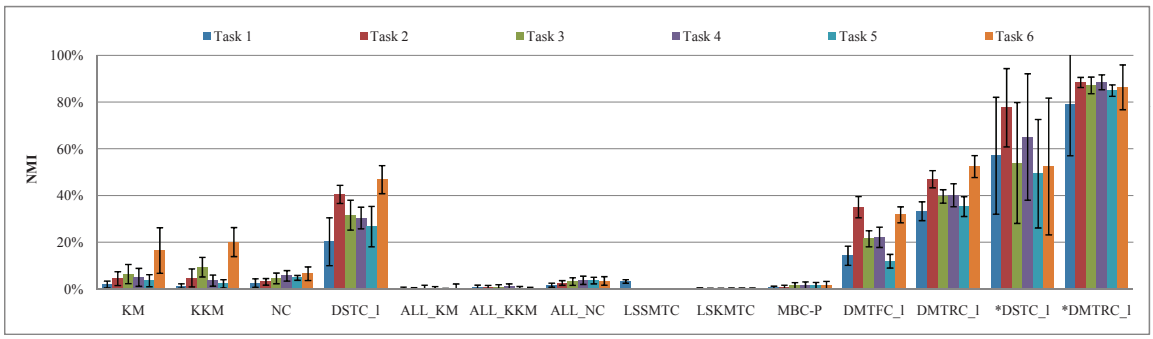


Fig. 13. NMI comparison with respect to each individual task on 40% data of the 20-Newgroups dataset.

## 6 Supplement to the Results of the Multi-Domain Sentiment Dataset

In this section, we will first give the average performance of all tasks in the metrics of NMI, ARI, ACC and the Area under ROC Curve (AUC) respectively. Then, we will give the performance of each individual task with 10%, 30%, and 50% data of the sentiment dataset in the metric of NMI. At last, we will show the CPU time of the methods.

### 6.1 Average Performance of All Tasks in Different Metrics

Figs. 14, 15, 16 and 17 show the average performance on the sentiment dataset in the metrics of NMI, ARI, ACC and AUC respectively.

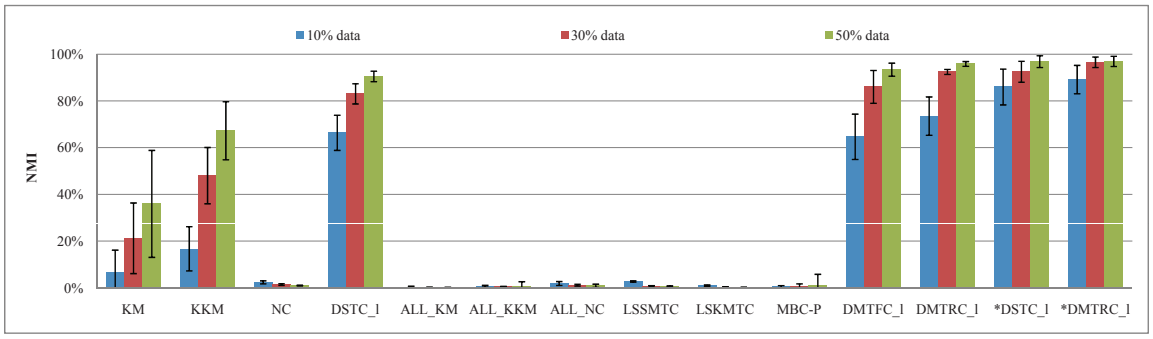


Fig. 14. Comparison of the average NMI of all tasks on the sentiment dataset.

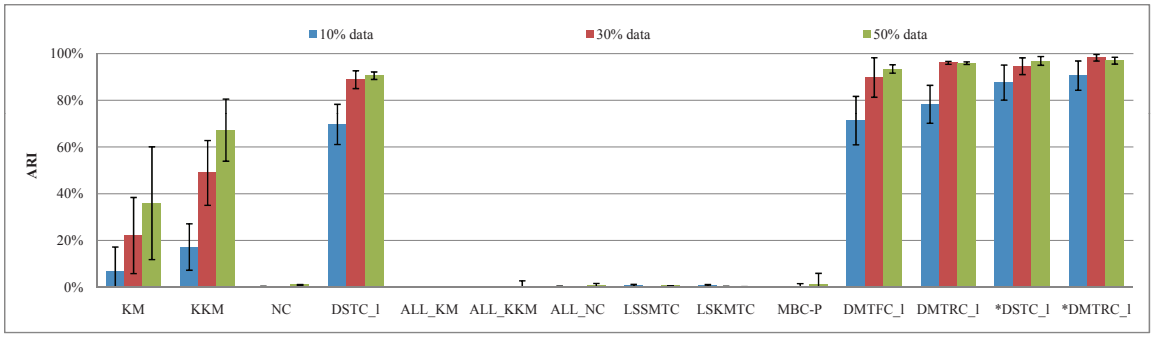


Fig. 15. Comparison of the average ARI of all tasks on the sentiment dataset.

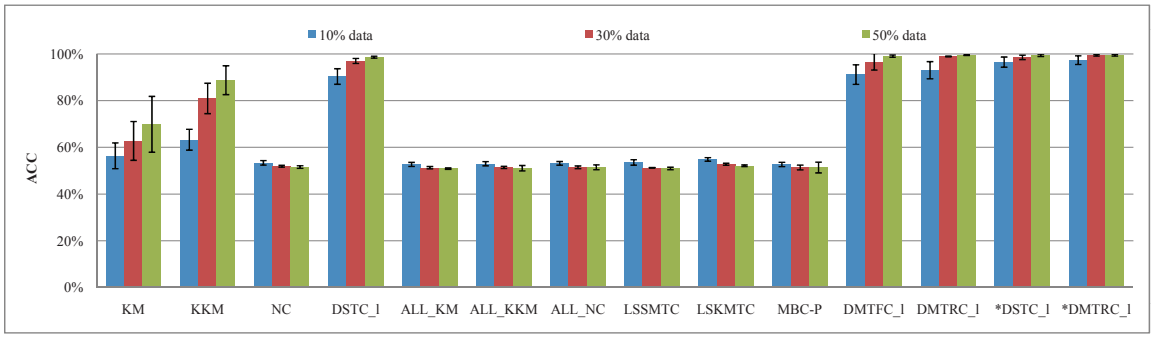


Fig. 16. Comparison of the average ACC of all tasks on the sentiment dataset.

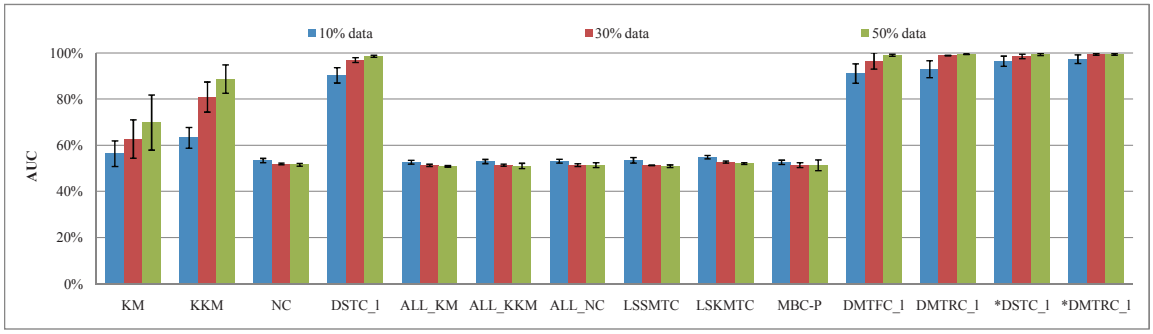


Fig. 17. Comparison of the average AUC of all tasks on the sentiment dataset.

### 6.2 Performance of Each Individual Task in the Metric of NMI

Figs. 18, 19, and 20 show the NMIs of the clustering algorithms with 10%, 30%, and 50% data of the 20-newsgroups dataset in the metric of NMI.

From the figures, we can see that the proposed clustering methods are much more robust than KM and KKM.

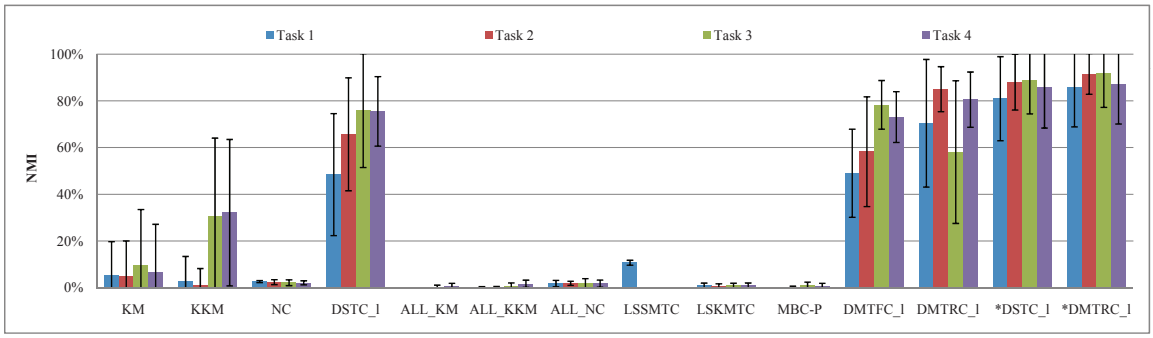


Fig. 18. NMI comparison with respect to each individual task on 10% data of the sentiment dataset.

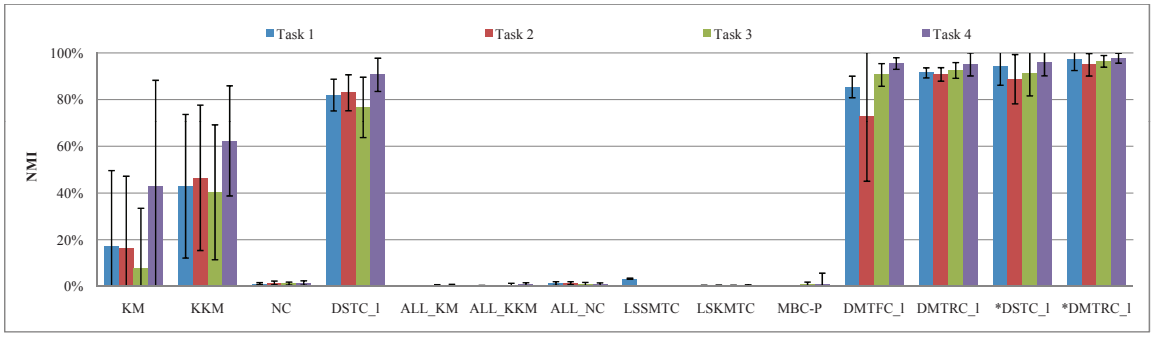


Fig. 19. NMI comparison with respect to each individual task on 30% data of the sentiment dataset.

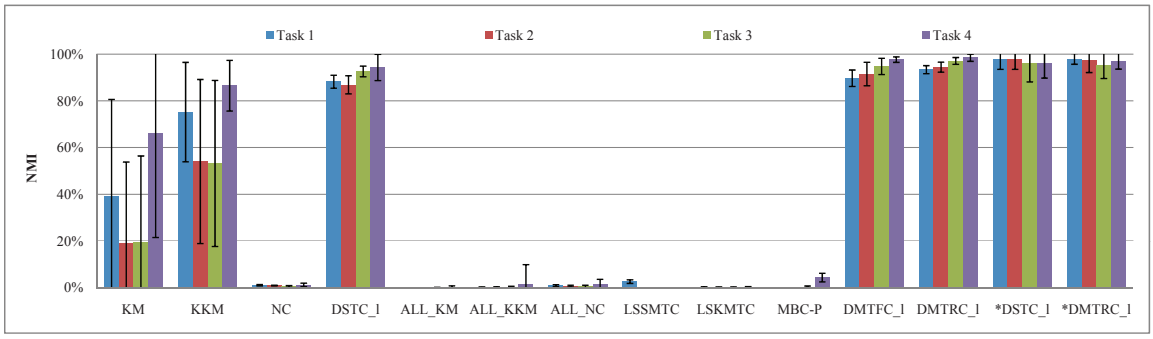


Fig. 20. NMI comparison with respect to each individual task on 50% data of the sentiment dataset.

## REFERENCES

[1] J. E. Kelley, "The cutting-plane method for solving convex programs," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 4, pp. 703–712, 1960.
[2] 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.*, vol. 21, 2009, pp. 1825–1832.
[3] H. Yang, Z. Xu, J. Ye, I. King, and M. R. Lyu, "Efficient sparse generalized multiple kernel learning," *IEEE Trans. Neural Netw.*, no. 3, pp. 433–446, 2011.
[4] X. L. Zhang and J. Wu, "Linearithmic time sparse and convex maximum margin clustering," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 1, no. 99, pp. 1–24, 2012.
[5] C. H. Teo, A. Smola, S. V. N. Vishwanathan, and Q. V. Le, "A scalable modular convex solver for regularized risk minimization," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2007, pp. 727–736.
[6] Y. Zhang and D. Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," in *Proc. 26th Conf. Uncertainty Artif. Intell.*, 2010, pp. 733–742.
[7] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, 2008.
[8] 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*, 2009, pp. 344–351.
[9] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ. Pr., 2004.