Domain Adaption via Feature Selection on Explicit Feature Map

Wan-Yu Deng, Amaury Lendasse, Yew-Soon Ong, Ivor W. Tsang, Lin Chen, Qing-Hua Zheng

***Abstract*—In most domain adaption approaches, all features are used for domain adaption. However, often, not every feature is beneficial for domain adaption. In such cases, incorrectly involving all features might cause the performance to degrade. In other words, to make the model trained on the source domain work well on the target domain, it is desirable to find invariant features for domain adaption rather than using all features. However, invariant features across domains may lie in a higher-order space, instead of in the original feature space. Moreover, the discriminative ability of some invariant features such as shared background information is weak, and needs to be further filtered. So, in this paper, we propose a novel domain adaption algorithm based on an explicit feature map and feature selection. The data is firstly represented by a kernel-induced explicit feature map, such that high-order invariant features can be revealed. Then, by minimizing the marginal distribution difference, conditional distribution difference, and the model error, the invariant discriminative features are effectively selected. This problem is NP hard to be solved, and we propose to relax it and solve it by a cutting plane algorithm. Experimental results on six real-world benchmarks have demonstrated the effectiveness and efficiency of the proposed algorithm, which outperforms many state-of-the-art domain adaption approaches.**

***Index Terms*—Domain adaption, Transfer learning, Feature selection, Distribution distance.**

I. Introduction

Classical supervised learning algorithms assume that the training and testing samples obey the same distribution. However, in many real-world scenarios, this assumption is invalid. For example, in object recognition, the samples may be collected under specific conditions, involving device type, position, orientation and so on. The model trained under one condition can not be directly used for another. Therefore, domain adaption is proposed [1], which aims to exploit the information of the target domain, so that the model trained on the source domain can generalize well on the target domain.

The major issue for domain adaption is how to reduce the difference across domains. There are two kinds of commonly used algorithms [1], which implement domain adaption from instance and feature-based approaches, respectively. Instance-based methods assume that certain parts of the source data can be used in the target domain by instance re-weighting or selection. For example, in [2], some source instances are firstly measured by conditional probability difference, and then, only the instances with small difference are selected. In [3], source instances are ordered by the contribution to the target domain, and then the 'bad' source instances are reduced while the other 'good' instances are encouraged. Feature-based methods [4]–[7] attempt to learn a 'good' feature representation for the target domain, such that the knowledge from the source domain can be transferred to the target domain. For example, the transfer component analysis (TCA) [4] method learns a latent subspace by principle component analysis to reduce the distribution difference across domains. Subspace alignment (SA) [5] represents source and target data by eigenvectors, and then aligns them by a mapping matrix. Hybrid heterogeneous transfer learning (HHTL) [6] attempts to learn a multi-layer representation for the source and target data respectively, and then learns multiple layer-wise mapping matrices to align data. Maximum independence domain adaption (MIDA) [7] aims to learn features which have maximal independence with the domain features so as to reduce the inter-domain discrepancy in distributions. Transfer subspace learning (TSL) [8] minimizes the Bregman divergence between domains in the selected subspace, so it boosts the performance when training and testing samples are not independent and identically distributed. Joint distribution adaption (JDA) [9] aims to jointly adapt both the marginal distribution and conditional distribution in a principled dimensionality reduction procedure, and construct a new feature representation that is effective and robust for substantial distribution difference.

Most previous methods use all features in the learning procedure. However, not all features are useful for domain adaption, and what is more, just some of them cause the difference across domains. If one can identify and remove these features, the difference across domains should be reduced. Therefore, S. Uguroglu et al. [10] proposed a feature selection method, feature-based Maximum Mean Discrepancy (f-MMD) for domain adaption. However, f-MMD is only focused on the marginal distribution. Moreover, it does not consider the discriminative ability and high-order characteristic of the invariant features. So, f-MMD suffers from four major limitations: 1) Finding invariant features can reduce the distribution difference, but, does not ensure that all of them contribute to domain adaption. For example, shared background information contributes very little to the prediction; 2) The shared invariant features may exist in a high-order space, rather than in the original space. Through f-MMD it may be difficult to find such high-order invariant features underlying domains efficiently, due to the lack of high-order explicit representation of the data; 3) f-MMD does not consider conditional distribution, and can

Wan-Yu Deng is with the Xi'an University of Posts & Telecommunications and Shaanxi Key Laboratory of Network Data Analysis and Intelligent Processing, China (e-mail: dengwanyu@126.com). Amaury Lendasse is with the University of Houston, USA; Yew-Soon Ong is with Nanyang Technological University, Singapore; Ivor W. Tsang is with University of Technology Sydney, Australia; Lin Chen is with the Xi'an University of Posts & Telecommunications, China. Qing-Hua Zheng is with Xi'an Jiaotong University, China.

not capture the local structure of the data accurately, especially for multi-class domain adaption problems; 4) Moreover, its optimization process is highly expensive due to the repetitive kernel matrix computation and inefficient optimization procedure. So, f-MMD may be considered unfit for large-scale problems.

In this paper, we propose a novel approach, called ***Explicit-Map based Feature Selection*** (EMFS), for domain adaption. EMFS attempts to, 1) reveal high-order invariant features by explicit feature map, so that the domain difference, when represented by them, can be dramatically reduced; 2) remove non-discriminative features from invariant features, such that the model trained on them becomes more sparse, which may lead to better generalized performance; 3) integrate feature learning and model learning, so that, the conditional distribution difference can be estimated, and the model can be obtained directly.

Our main contribution is on proposing a novel feature selection approach, which selects high-order invariant discriminative features, so as to reduce the domain distance more effectively. In contrast with f-MMD, EMFS can reveal invariant features in high-order space, and further remove non-discriminative features. Moreover, it can estimate and adjust a conditional distribution. This method is proved to be effective in experiments on a wide range of data sets, and outperforms many state-of-the-art domain adaption algorithms. The rest of the paper is organized as follows: In Section II, related work is introduced briefly. In Section III and IV the optimization objective and optimization procedure of the proposed algorithm is described in detail. In Section V, we summarize the experimental studies on the widely used benchmark problems. Section VI summarizes the main conclusions.

II. Related Work

In this section, we present some related work of the proposed algorithm. For clarity, the frequently used notations are summarized in Table I.

TABLE I
NOTATIONS AND DESCRIPTIONS

Notation	Description	Notation	Description
C	#classes	$\mathcal{P}_s(\mathbf{X}_s)$	marginal distribution of $\mathbf{X}_s$
m	#original features	$\mathcal{P}_t(\mathbf{X}_t)$	marginal distribution of $\mathbf{X}_t$
$\mathbf{X}_t$	target instances	$\mathcal{Q}_s(\mathbf{X}_s)$	conditional distribution of $\mathbf{X}_s$
$\mathbf{X}_s$	source instances	M	#explicit feature
n_s	#source instances	n_t	#target instances

A. Explicit Feature Map

The state of the art in terms of explicit feature approximations of kernels is random Fourier transformation [11]. It can provide an easily computable, low-dimensional feature representation for kernels. However, since each random Fourier feature corresponds to an independent random projection, a collection of such features will not, in general, be an orthogonal projection. This implies that, many redundant features are involved. A different option for approximating the kernel is truncated Taylor expansion [15]. The idea is to use the Taylor series expansion to approximate kernels $\kappa(\mathbf{x}, \mathbf{y})$, where each term in the Taylor series can be expressed as a sum of matching monomial in $\mathbf{x}$ and $\mathbf{y}$. Compared to random Fourier features, Taylor-based features can provide a better approximation of the kernel with a fixed computational budget [15]. Therefore, we will take truncated Taylor expansion in our study, although other explicit approximations also can be used here. Moreover, for the kernel type, we will select Gaussian and Polynomial kernel as kernel function, since they are most commonly used and always produce promising performance.

1) Gaussian Kernel Explicit Map: Gaussian kernel implicitly maps the data to an infinite dimensional Hilbert space, so, finding the actual explicit map $\varphi(\cdot)$ of Gaussian kernel is nontrivial. By Taylor expansion, it can be expressed as:

$$\kappa(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}} = e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}} e^{-\frac{\|\mathbf{y}\|^2}{2\sigma^2}} e^{\frac{<\mathbf{x},\mathbf{y}>}{\sigma^2}} \tag{1}$$

The first two factors depend on $\mathbf{x}$ and $\mathbf{y}$ separately, so we only need focus on the third factor:

$$e^{\frac{<\mathbf{x},\mathbf{y}>}{\sigma^2}} = \sum_{k=0}^{\infty} \frac{1}{k!} \Big(\frac{<\mathbf{x}, \mathbf{y}>}{\sigma^2}\Big)^k \tag{2}$$

Therein,

$$<\mathbf{x}, \mathbf{y}>^k = (\sum_{i=1}^{m} x_i y_i)^k = \sum_{\mathbf{a}\in\{1,...,m\}^k} \Big(\prod_{i=1}^{k} x_{a_i}\Big)\Big(\prod_{i=1}^{k} y_{a_i}\Big) \tag{3}$$

where m is the number of features, and $\mathbf{a}$ enumerates over all selections of k coordinates of $\mathbf{x}$. Substituting this back into Eq.(2) and Eq.(1) leads to the following explicit features of the degree k :

$$\varphi_{k,\mathbf{a}}(\mathbf{x}) = e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}} \frac{1}{\sigma^k \sqrt{k!}} \prod_{i=0}^{k} x_{a_i} \tag{4}$$

More specifically, given one degree d, the explicit approximation of Gaussian kernel is:

$$\varphi(\mathbf{x}) = \cup_{k=1}^{d} \{\varphi_{k,\mathbf{a}\in\{1,...,m\}^k}(\mathbf{x})\} \tag{5}$$

which corresponds to truncating the Taylor expansion after the d-th term.

It will generate m^d features for the degree d. However, many features are duplicates, although they are from different permutations of $\mathbf{a}$. Collecting them into a single feature for each distinct monomial, we have $\binom{m+d-1}{d}$ features. Combining all results of degree $k \leq d$, the total number of features is $M = \binom{m+d}{d}$. Computing explicit features will need extra time, however, it only needs to be executed once, and so can be completed as an off-line operation.

2) Polynomial Kernel Explicit Map: Similar to a Gaussian kernel, the Polynomial kernel of degree d

$$\kappa(\mathbf{x}, \mathbf{y}) = (\gamma \mathbf{x}'\mathbf{y} + \rho)^d \tag{6}$$

corresponds to the features containing all monomials of degree $k \leq d$. More specifically, they can be written as:

$$\varphi_{k,\mathbf{a}}(\mathbf{x}) = \sqrt{\binom{d}{k} \rho^{d-k}} \prod_{i=1}^{k} x_{a_i} \ , \quad k = 0, ..., d \tag{7}$$

where $\mathbf{a} \in \{1, ..., m\}^k$ enumerates over all selections of k coordinates in $\mathbf{x}$. After using the multinomial theorem and regrouping [15], the explicit features of Polynomial kernel can be written as:

$$\varphi(\mathbf{x}) = \cup_{k=1}^{d} \{\varphi_{k,\mathbf{a}\in\{1,...,m\}^k}(\mathbf{x})\} \tag{8}$$

For the degree d, it will generate a total of $M = \binom{m+d}{d}$ features. For instance, when d = 2, the explicit features of polynomial kernel can be written as:

$$\begin{aligned}\varphi(\mathbf{x}) = (&x_m^2, ..., x_1^2, \sqrt{2}x_m x_{m-1}, ..., \sqrt{2}x_m x_1,\\ &\sqrt{2}x_{m-1}x_{m-2}, ..., \sqrt{2}x_{m-1}x_1, ...,\\ &\sqrt{2}x_2 x_1, \sqrt{2\rho}x_m, ..., \sqrt{2\rho}x_1, \rho)\end{aligned} \tag{9}$$

It is worth noting that, besides Gaussian and Polynomial kernels, our approach holds for other kernels or multiple kernels.

After explicit map, the source and target domain can be represented as $\{(\varphi(\mathbf{x}_i) \in \mathbf{R}^M, y_i)\}_{i=1}^{n_s}$ and $\{\varphi(\mathbf{x}_i) \in \mathbf{R}^M\}_{i=1}^{n_t}$ respectively. Obviously, the explicit features will increase dramatically *w.r.t* degree d and M. This means that, it needs a very efficient optimization method to solve such high-dimensional feature selection problem.

B. Maximum Mean Discrepancy on Explicit Map

Given samples $\mathbf{X}_s = \{\mathbf{x}_s\}_{s=1}^{n_s}$ and $\mathbf{X}_t = \{\mathbf{x}_t\}_{t=1}^{n_t}$ drawn from two distributions, there exist many criteria such as the Kullback-Leibler (KL) divergence, that can be used to estimate their distance. However, many of these estimators are parametric or require an intermediate density estimate. Recently, a nonparametric distance estimate was designed by embedding distributions in an RKHS [16]. Gretton et al. [14] introduced the maximum mean discrepancy (MMD) for comparing distributions based on the corresponding RKHS distance. Instead of using an implicit feature map, we propose to measure distance on an explicit feature map by MMD. Let $\mathbf{X}_s$ and $\mathbf{X}_t$ obey distributions $\mathcal{P}_s$ and $\mathcal{P}_t$. The empirical estimate of the distance between $\mathcal{P}_s$ and $\mathcal{P}_t$ is $Dist(\mathbf{X}_s, \mathbf{X}_t) = \|\frac{1}{n_s}\sum_{s=1}^{n_s}\varphi(\mathbf{x}_s) - \frac{1}{n_t}\sum_{t=1}^{n_t}\varphi(\mathbf{x}_t)\|_F^2$, where φ is kernel-induced explicit feature map. It can be shown that [16], when the kernel is universal, MMD on the kernel-induced explicit feature map will asymptotically approach zero if and only if the two distributions are the same.

III. Proposed Method

In this work, we focus on the setting that some labeled data are available in a source domain, while only unlabeled data are available in the target domain. We denote the source domain as $\{(\mathbf{x}_s \in \mathbf{R}^{1\times m}, y_s \in \{1, ..., C\})\}_{s=1}^{n_s}$, where $\mathbf{x}_s$ denotes the m-dimensional source-domain instance, y_s indicates the corresponding label. We denote $\mathbf{X}_s = \{\mathbf{x}_s\}_{s=1}^{n_s}$ as the total source instances. Similarly, we denote the unlabeled target data as $\{\mathbf{x}_t\}_{t=1}^{n_t} = \mathbf{X}_t$, where $\mathbf{X}_t \in \mathbf{R}^{n_t\times m}$ are the observed target domain instances. Let $\mathcal{P}_s(\mathbf{X}_s)$ and $\mathcal{P}_t(\mathbf{X}_t)$ be the marginal distributions of $\mathbf{X}_s$ and $\mathbf{X}_t$, $\mathcal{Q}_s(\mathbf{X}_s)$ and $\mathcal{Q}_t(\mathbf{X}_t)$ be the conditional distributions of $\mathbf{X}_s$ and $\mathbf{X}_t$. Our task is then to predict the label $\hat{y}_t$ corresponding to the input $\mathbf{x}_t$ in the target domain under the assumptions $\mathcal{P}_s(\mathbf{x}_s) \neq \mathcal{P}_t(\mathbf{x}_t)$ and $\mathcal{Q}_s(y_s|\mathbf{x}_s) \neq \mathcal{Q}_t(y_t|\mathbf{x}_t)$.

A. Objectives

Most domain adaption methods assume that $\mathcal{P}_s \neq \mathcal{P}_t$, but $\mathcal{Q}_s = \mathcal{Q}_t$. However, in many real-world applications, the conditional probability may also change across domains due to noisy or dynamic factors underlying the observed data [4]. In this paper, we use the weaker assumption that $\mathcal{P}_s \neq \mathcal{P}_t$, but there exists a binary indicator $\mathbf{d} \in \{0, 1\}^M$, such that $P(\varphi(\mathbf{x}_s) \odot \mathbf{d}) \approx P(\varphi(\mathbf{x}_t) \odot \mathbf{d})$ and $P(y_s|\varphi(\mathbf{x}_s) \odot \mathbf{d}) \approx P(y_t|\varphi(\mathbf{x}_t) \odot \mathbf{d})$. A key issue is how to find this indicator $\mathbf{d}$. Since we have no labeled data in the target domain, $\mathbf{d}$ cannot be learned by directly minimizing the distance between $P(y_s|\varphi(\mathbf{x}_s) \odot \mathbf{d})$ and $P(y_t|\varphi(\mathbf{x}_t) \odot \mathbf{d})$. Here, we propose to learn $\mathbf{d}$ and a linear classifier f with weight $\mathbf{W}$ jointly, such that: 1) the distance between the marginal distributions $P(\varphi(\mathbf{x}_s) \odot \mathbf{d})$ and $P(\varphi(\mathbf{x}_t) \odot \mathbf{d})$ is small, 2) $\varphi(\mathbf{x}_s) \odot \mathbf{d}$ and $\varphi(\mathbf{x}_t) \odot \mathbf{d}$ preserve important properties of $\varphi(X_s)$ and $\varphi(X_t)$, such that, $\mathbf{d}$ and $\mathbf{W}$ satisfy $P(y_s|\varphi(\mathbf{x}_s) \odot \mathbf{d}) \approx P(y_t|\varphi(\mathbf{x}_t) \odot \mathbf{d})$, and 3) the learned classifier $f = (\varphi(\mathbf{x}) \odot \mathbf{d})\mathbf{W}$ can be well used to make predictions on the target data $\varphi(\mathbf{x}_t)$. We believe that domain adaption under this assumption is more realistic, though also more challenging.

In summary, the goal is to find *invariant* and *discriminative* features from the explicit feature map, and jointly learn an adapted classifier on the selected features. It needs $\{\mathbf{d}, \mathbf{W}\}$ to satisfy three desirable properties: 1) minimize the marginal distribution difference; 2) minimize the conditional distribution difference; and 3) minimize the empirical error on the labeled data.

1) Minimize marginal distribution difference: Given an explicit-map representation $\varphi(\mathbf{x})$ induced by a kernel, the objective to minimize marginal distribution difference by feature selection can be written as follows:

$$\min_{\mathbf{d}\in\mathcal{D}} \ \ell_m(\mathbf{d}) \equiv \left\| \frac{1}{n_s}\sum_{\mathbf{x}_s\in\mathbf{X}_s} \varphi(\mathbf{x}_s) \odot \mathbf{d} - \frac{1}{n_t}\sum_{\mathbf{x}_t\in\mathbf{X}_t} \varphi(\mathbf{x}_t) \odot \mathbf{d} \right\|_F^2 \tag{10}$$

where $\|\cdot\|_F^2$ is a squared Frobenius norm, $\mathbf{d} \in \{0, 1\}^M$ is a binary indicator vector whose entries are 1 for the selected features and 0 otherwise, $\mathcal{D} = \{\mathbf{d} | \mathbf{d} \in \{0, 1\}^M, \|\mathbf{d}\|_0 \leq B\}$ is the domain of $\mathbf{d}$, and B is the maximum number of selected features.

Only reducing the marginal distribution difference does not ensure that the conditional distributions difference will also be small. Therefore, it needs to be explicitly reduced as follows.

2) Minimize conditional distribution difference: Since $\mathcal{Q}_t(y_t|\mathbf{x}_t)$ of the target domain is unknown, we can not directly compare the conditional distributions. Some recent works started to match the conditional distributions by kernel density estimation [17] and co-training [18]. However, they all need some labelled target domain samples, and thus cannot address our problem.

Similar to [4], we propose to predict the labels for the target data by the linear classifier $f(\mathbf{x}) = (\varphi(\mathbf{x}) \odot \mathbf{d})\mathbf{W}$, trained on the source data. Moreover, since the posterior probabilities $\mathcal{Q}_s(y_s|\mathbf{x}_s)$ and $\mathcal{Q}_t(y_t|\mathbf{x}_t)$ are complicated, we turn to the class-conditional distributions $\mathcal{Q}_s(\mathbf{x}_s|y_s = c)$ and $\mathcal{Q}_t(\mathbf{x}_t|y_t = c)$ instead. We can measure the conditional distribution dis-

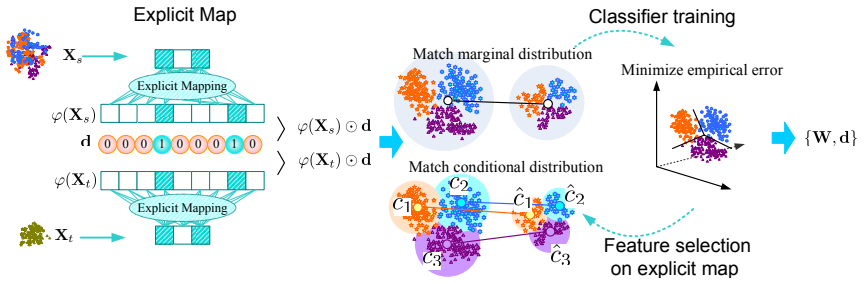


Fig. 1. In EMFS, the data is firstly represented by explicit feature map. Then, feature selection and model learning are jointly optimized.

tance between domains for every label $c \in \{1, ..., C\}$, and minimize the conditional distribution difference as follows:

$$\min_{\mathbf{d}\in\mathcal{D}} \ \ell_c(\mathbf{d}) \equiv \left\| \frac{1}{n_s^{(c)}} \sum_{\mathbf{x}_s \in \mathbf{X}_s^{(c)}} \varphi(\mathbf{x}_s)\odot\mathbf{d} - \frac{1}{n_t^{(c)}} \sum_{\mathbf{x}_t \in \mathbf{X}_t^{(c)}} \varphi(\mathbf{x}_t)\odot\mathbf{d} \right\|_F^2 \tag{11}$$

where $\mathbf{X}_s^{(c)} = \{\mathbf{x}_s | \mathbf{x}_s \in \mathbf{X}_s \wedge y_s = c\}$ denotes source data with label c, and $n_s^{(c)} = |\mathbf{X}_s^{(c)}|$. Correspondingly, $\mathbf{X}_t^{(c)} = \{\mathbf{x}_t | \mathbf{x}_t \in \mathbf{X}_t \wedge \hat{y}_t = c\}$ are target instance with label c, where y_t is the pseudo label of $\mathbf{x}_t$, and $n_t^{(c)} = |\mathbf{X}_t^{(c)}|$.

3) Minimize the empirical error: As discussed above, the selected features not only should reduce the distribution distance, but also be discriminative for label prediction. To find discriminative features, it needs to find the features who minimize the empirical error. In order to unify binary, multi-class and regression as one unified form, One-vs-All coding of ECOC (Error-Correcting Output Codes) [19] is introduced. That is, if $y_s = c$, the expected ECOC coding is $\mathbf{y}_s = [0, .., 0, \overset{c}{1}, 0, ..., 0]_{1\times C}$ where c-th entry is '1' while others zeros. Since the target domain has no labels, we only minimize the empirical error on the source domain:

$$\begin{aligned} \min_{\mathbf{d}\in\mathcal{D}} \min_{\mathbf{W}} \quad & \ell_f(\mathbf{W}, \mathbf{d}) \equiv \sum_{\mathbf{x}_s \in \mathbf{X}_s} \frac{1}{2} \|\boldsymbol{\xi}_s\|_F^2 + \frac{\gamma}{2} \|\mathbf{W}\|_F^2 \\ & \text{s.t. } \boldsymbol{\xi}_s = (\varphi(\mathbf{x}_s) \odot \mathbf{d})\mathbf{W} - \mathbf{y}_s, s = 1, ..., n_s \end{aligned} \tag{12}$$

The pseudo label can be predicted by

$$\hat{y} = \arg \max_{c\in\{1,2,...,C\}} f_c(\mathbf{x}) \tag{13}$$

where f_c is the c-th entry of $\mathbf{f}(\mathbf{x}) = (\varphi(\mathbf{x}) \odot \mathbf{d})\mathbf{W}$

It's worth noting that, although many pseudo labels may be incorrect, they still can be leveraged to match the conditional distributions. The reason is that the distributions are matched by exploiting the adequate statistics rather than the density estimates. In this way, the predictor can be improved gradually along with iterative optimizations. This argument will be verified in the experiments.

By combining all the objectives, the final optimization problem can be written as

$$\begin{aligned} \min_{\mathbf{d}\in\mathcal{D}} \min_{\mathbf{W}} \quad & \ell(\mathbf{W}, \mathbf{d}) \equiv \ell_f(\mathbf{W}, \mathbf{d}) + \ell_m(\mathbf{d}) + \ell_c(\mathbf{d}) \\ & \text{s.t. } \boldsymbol{\xi}_s = (\varphi(\mathbf{x}_s) \odot \mathbf{d})\mathbf{W} - \mathbf{y}_s, s = 1, ..., n_s \end{aligned} \tag{14}$$

B. Optimization Procedure

The proposed formulation selects features naturally with the desired cardinality. This is much more efficient than the sparsity induced methods. However, this problem is NP-hard to solve due to the combinatorial integral constraints on $\mathbf{d}$. To address it, it is necessary to make some transformations and relaxations. It is not difficult to find that, the inner minimization problem with a fixed $\mathbf{d}$ can be solved equivalently in its dual. By introducing $\boldsymbol{\alpha} \in \mathbf{R}^{n_s\times C}$, the dual variable to the constraint $\boldsymbol{\xi}_i = (\varphi(\mathbf{x}_i) \odot \mathbf{d})\mathbf{W} - \mathbf{y}_i$, we can solve the inner regression problem in its dual. Specifically, the Lagrangian function of the inner regression problem is

$$\begin{aligned} \ell(\mathbf{W}, \boldsymbol{\Xi}, \boldsymbol{\alpha}, \mathbf{d}) = & \frac{1}{2}\|\boldsymbol{\Xi}\|_F^2 + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 + tr\left(\boldsymbol{\alpha}(\boldsymbol{\Xi} - \boldsymbol{\Phi}_s \overset{\diamond}{\mathbf{d}} \mathbf{W} + \mathbf{Y}_s)'\right) \\ & + \ell_m(\mathbf{d}) + \ell_c(\mathbf{d}) \end{aligned} \tag{15}$$

where $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1', ..., \boldsymbol{\xi}_{n_s}']'$, $\boldsymbol{\Phi}_s = \{\varphi(\mathbf{x}_s)\}_{s=1}^{n_s}$, and $\overset{\diamond}{\mathbf{d}} = \text{diag}(\mathbf{d})$ which denotes the diagonal matrix whose diagonal entries correspond to $\mathbf{d}$.

Setting the derivatives of $\ell_f(\mathbf{W}, \boldsymbol{\Xi}, \boldsymbol{\alpha}, \mathbf{d})$ *w.r.t.* $\mathbf{W}$ and $\boldsymbol{\Xi}$ to zero, the Karush-Kuhn-Tucker (KKT) conditions can be obtained:

$$\gamma\mathbf{W} = \overset{\diamond}{\mathbf{d}}\boldsymbol{\Phi}_s'\boldsymbol{\alpha}, \quad \boldsymbol{\alpha} = -\boldsymbol{\Xi} \tag{16}$$

By substituting Eq.16 into Eq.15, it can be converted into the following dual formulation:

$$\min_{\mathbf{d}} \max_{\boldsymbol{\alpha}\in\mathbf{R}^{n_s\times C}} \ell(\boldsymbol{\alpha}, \mathbf{d}) \tag{17}$$

where

$$\ell(\boldsymbol{\alpha}, \mathbf{d}) = tr(\boldsymbol{\alpha}\mathbf{Y}_s') - \frac{1}{2} tr\left(\boldsymbol{\alpha}(\frac{1}{\gamma}\boldsymbol{\Phi}_s \overset{\diamond}{\mathbf{d}} \boldsymbol{\Phi}_s' + \mathbf{I})\boldsymbol{\alpha}'\right) + \ell_m(\mathbf{d}) + \ell_c(\mathbf{d}) \tag{18}$$

However, this problem is still a mixed integer programming (MIP) problem, which is computationally expensive in general. Following [26], we introduce a mild convex relaxation for our formulation. According to the minimax inequality [27], when interchanging the order of $\min_{\mathbf{d}\in\mathcal{D}}$ and $\max_{\alpha\in\mathbf{R}^{n_s\times C}}$, then the saddle-point problem can be lower-bounded by

$$\min_{\mathbf{d}\in\mathcal{D}} \max_{\boldsymbol{\alpha}\in\mathbf{R}^{n_s\times C}} \ell(\boldsymbol{\alpha}, \mathbf{d}) \geq \max_{\boldsymbol{\alpha}\in\mathbf{R}^{n_s\times C}} \min_{\mathbf{d}\in\mathcal{D}} \ell(\boldsymbol{\alpha}, \mathbf{d}) \tag{19}$$

Algorithm 1 Explicit-Map based Feature Selection (EMFS)

Input: labelled source data $\{(\mathbf{x}_i, t_i)\}_{i=1}^{n_s}$, unlabeled target data $\{\mathbf{x}_j\}_{j=1}^{n_t}$, #selected features B in each iteration.
Output: the adapted classifier

Represent the data by explicit map: $\{(\varphi(\mathbf{x}_i), y_i)\}_{i=1}^{n_s}$ and $\{\varphi(\mathbf{x}_j)\}_{j=1}^{n_t}$
Initialize $\mathcal{C} = \phi$, $\boldsymbol{\alpha} = \mathbf{Y}_s$, $p = 1$;
while not convergence **do**
  update $\mathbf{d}^p$ with given $\boldsymbol{\alpha}$ and $\{\hat{y}_j\}_{j=1}^{n_t}$ by **Algorithm 2**
  update $\mathcal{C}$ by $\mathcal{C} = \mathcal{C} \cup \{\mathbf{d}^p\}$
  update p by $p = p + 1$;
  solve the reduced subproblem by **Algorithm 4** to get $\widehat{\mathcal{W}} = [\widehat{\mathbf{W}}^1, ..., \widehat{\mathbf{W}}^{\prime}]$
  update $\boldsymbol{\alpha} = \mathbf{Y}_s - \sum_{p=1}^{T} \boldsymbol{\Phi}_s \overset{\diamond}{\mathbf{d}^p} \widehat{\mathbf{W}}^p$
  update pseudo target labels $\{\hat{y}_j\}_{j=1}^{n_t}$ by (34)
return $\widehat{\mathcal{W}}$ and $\mathcal{C}$

And furthermore, this relaxed problem can be transformed into a convex QCQP problem by introducing an additional parameter $\theta \in \mathbf{R}$,

$$\begin{aligned} \max_{\boldsymbol{\alpha} \in \mathbf{R}^{n_s \times C}, \theta \in \mathbf{R}} \quad & \theta \\ \text{s.t.} \quad & \theta \leq \ell(\boldsymbol{\alpha}, \mathbf{d}), \forall \mathbf{d} \in \mathcal{D} \end{aligned} \tag{20}$$

The constraint domain $\mathcal{D}$ contains a combinatorial number of $\mathbf{d}$s, making a combinatorial number of optimization problem involved. However, only a few of them are active since it only needs to select a small number of features. So, we turn to cutting plane algorithm [20] to iteratively find the most active constraint and add it to the active constraint set $\mathcal{C}$, which is initialized to an empty set ϕ. $\mathcal{C}$ is always a subset of $\mathcal{D}$, i.e., $\mathcal{C} \subseteq \mathcal{D}$. Given the updated set $\mathcal{C}$, we solve the following reduced QCQP subproblem,

$$\begin{aligned} \max_{\boldsymbol{\alpha} \in \mathcal{A}, \theta \in \mathbf{R}} \quad & \theta \\ \text{s.t.} \quad & \theta \leq \ell(\boldsymbol{\alpha}, \mathbf{d}), \forall \mathbf{d} \in \mathcal{C} \end{aligned} \tag{21}$$

As $|\mathcal{C}|$ is very small, one can efficiently solve Eq.21, and obtain a new $\boldsymbol{\alpha}$ to update the active constraint set $\mathcal{C}$ via the worst-case analysis [21]. The whole procedure iterates until the stopping condition is achieved. The proposed algorithm is described in Algorithm 1, which mainly involves three operations: feature selection, model updating and pseudo label prediction

1) Feature Selection Procedure: Given $\boldsymbol{\alpha}$ and the pseudo labels $\{\hat{y}_i\}_{i=1}^{n_t}$ of target data $\{\varphi(\mathbf{x}_i)\}_{i=1}^{n_t}$, the feature selection in p-th iteration need us solve:

$$\begin{aligned} \mathbf{d}^p &= \arg\min_{\mathbf{d} \in \mathcal{D}} \ell(\boldsymbol{\alpha}, \mathbf{d}) \\ &= \arg\min_{\mathbf{d} \in \mathcal{D}} -\|\overset{\diamond}{\mathbf{d}} \boldsymbol{\Phi}_s^T \boldsymbol{\alpha}\|_F^2 + \ell_m(\mathbf{d}) + \ell_c(\mathbf{d}) \end{aligned} \tag{22}$$

Let $\mathbf{A} = \boldsymbol{\Phi}_s^T \boldsymbol{\alpha}$, $\mathbf{m}_s = \frac{1}{n_s} \sum_{\mathbf{x}_s \in \mathbf{X}_s} \varphi(\mathbf{x}_s)$, $\mathbf{m}_t = \frac{1}{n_t} \sum_{\mathbf{x}_t \in \mathbf{X}_t} \varphi(\mathbf{x}_t)$, $\mathbf{m}_s^c = \frac{1}{n_s^{(c)}} \sum_{\mathbf{x}_s \in \mathbf{X}_s^{(c)}} \varphi(\mathbf{x}_s)$ and $\mathbf{m}_t^c = \frac{1}{n_t^{(c)}} \sum_{\mathbf{x}_t \in \mathbf{X}_t^{(c)}} \varphi(\mathbf{x}_t)$, then, the score for the j-th feature can be calculated as follows:

$$\mathbf{s}(j) = \mathbf{s}_m f(j) + \mathbf{s}_c(j) - \mathbf{s}_f(j) \tag{23}$$

Algorithm 2 Feature Selection Procedure

Input: source data $\{\varphi(\mathbf{x}_i), y_i\}_{i=1}^{n_s}$, unlabeled target data $\{\varphi(\mathbf{x}_i), \hat{y}_i\}_{i=1}^{n_t}$, dual variable $\boldsymbol{\alpha}$ B selected features in each iteration, and the selection vector $\mathbf{d}$.
Output: the most active constrain

Initialize $\mathbf{d} = \mathbf{0}$
Compute $\mathbf{s}(j) = \mathbf{s}_m(j) + \mathbf{s}_c(j) - \mathbf{s}_f(j)$, $\forall j = 1, .., M$;
Set B entries of $\mathbf{d}$ w.r.t. the top B values of $\mathbf{s}$ to 1.
return $\mathbf{d}$

where

$$\mathbf{s}_m(j) = (\mathbf{m}_s(j) - \mathbf{m}_t(j))^2 \tag{24}$$

$$\mathbf{s}_c(j) = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} (\mathbf{m}_s^{(c)}(j) - \mathbf{m}_t^{(c)}(j))^2 \tag{25}$$

$$\mathbf{s}_f(j) = \sum_{i=1}^{C} \mathbf{A}(j, i)^2 \tag{26}$$

The optimization problem becomes:

$$\mathbf{d}^p = \arg\max_{\mathbf{d} \in \mathcal{D}} \sum_{j=1}^{D} \mathbf{s}(j) d_j \tag{27}$$

Apparently, this problem can be efficiently addressed by setting the d_j to 1 of the top-B number of $\mathbf{s}(j)$s and the rests to 0. In other words, the most active constraint can be identified by choosing the features with the B smallest values in $\mathbf{s}$. This procedure is summarized in Algorithm 2. Once the most active constraint $\mathbf{d}^p$ is obtained, it is then added to the active constraint set $\mathcal{C} = \mathcal{C} \cup \{\mathbf{d}^p\}$.

2) Prediction Model Optimization: After updating $\mathcal{C}$, the subproblem in Eq.21 with constraints $\mathcal{C}$ is then solved. Since $|\mathcal{C}|$ is small, the problem is easy to be solved by sub-gradient methods [21]. However, directly solving it *w.r.t.* the dual variables $\boldsymbol{\alpha}$ is very expensive, especially when n_s is large. Suppose there are T constraints in $\mathcal{C}$, i.e. $T = |\mathcal{C}|$. Even if there are a large amount of features, at most TB features are selected by $\mathcal{C}$ (where $TB \ll M$). Based on this observation, the subproblem might be solved more efficiently *w.r.t.* the primal variables $\mathbf{W}$. The subproblem can be equivalently modelled as a squared group-LASSO problem:

$$\min_{[\mathbf{W}_1, ..., \mathbf{W}_T]} \frac{\gamma}{2} \left(\sum_{p=1}^{T} \|\mathbf{W}^p\|_F \right)^2 + \frac{1}{2} \|\boldsymbol{\Xi}\|_F^2 \tag{28}$$

where $\boldsymbol{\Xi} = \mathbf{Y}_s - \sum_{p=1}^{T} \boldsymbol{\Phi}_s \overset{\diamond}{\mathbf{d}^p} \mathbf{W}^p$ denotes the residual matrix, and $\mathbf{W}^p$ denotes the weight matrix defined on the features indicated by $\mathbf{d}^p$. The dual variable $\boldsymbol{\alpha}$ can be recovered by $\boldsymbol{\alpha} = \boldsymbol{\xi}$, which is required to find the new most active constraint.

For convenience, let $\mathcal{W} = [\mathbf{W}^1, ..., \mathbf{W}^T] \in \mathbf{R}^{TB \times C}$, $g(\mathcal{W}) = \frac{\gamma}{2} \left(\sum_{p=1}^{T} \|\mathbf{W}^p\|_F \right)^2$ and $h(\mathcal{W}) = \frac{1}{2} \|\boldsymbol{\Xi}\|_F^2$, Eq.28 can be rewritten as:

$$\mathcal{F}(\mathcal{W}) = \frac{\gamma}{2} \left(\sum_{p=1}^{T} \|\mathbf{W}^p\|_F \right)^2 + \frac{1}{2} \|\boldsymbol{\Xi}\|_F^2 \tag{29}$$

Algorithm 3 Moreau Projection

Input: $\mathcal{G} = [\mathbf{G}^1, ..., \mathbf{G}^T]$
Output: $\widehat{\mathcal{W}}$

1: Calculate $u_p = \|\mathbf{G}^p\|_F$ for all $p \in \{1, ..., T\}$
2: Sort $\mathbf{u}$ to obtain $\widehat{\mathbf{u}}$ such that $\hat{u}_1 \geq ... \geq \hat{u}_T$
3: Find $\rho = \max\left\{p | \hat{u}_p - \frac{1}{p+\tau}\sum_{i=1}^{p}\hat{u}_i > 0, p = 1, ..., T\right\}$
4: Calculate the threshold value $\zeta = \frac{1}{p+\tau}\sum_{i=1}^{\rho}\hat{u}_i$
5: Compute by
6: **return** $\widehat{\mathcal{W}}$

Similar to [22], we propose to solve the problem by accelerated proximal gradient method (APG), which iteratively minimizes the following quadratic approximation,

$$\begin{aligned}\widetilde{\mathcal{F}}(\mathcal{W}, \mathcal{W}_k) &= h(\mathcal{W}_k) + \langle \nabla h, \mathcal{W} - \mathcal{W}_k \rangle + \frac{\tau}{2}\|\mathcal{W} - \mathcal{W}_k\|_F^2 + g(\mathcal{W}) \\ &= \frac{\tau}{2}\|\mathcal{W} - \mathcal{G}\|_F^2 + g(\mathcal{W}) + h(\mathcal{W}_k) - \frac{1}{2\tau}\|\nabla h\|_F^2\end{aligned} \tag{30}$$

where ∇h denotes the gradient of h at point $\mathcal{W}_k$, $\tau > 0$ is the Lipschitz constant of $h(\mathcal{W})$, and $\mathcal{G} = \mathcal{W}_k - \frac{1}{\tau}\nabla h = [\mathbf{G}^1, ..., \mathbf{G}^T]$.

Since $h(\mathcal{W}_k) - \frac{1}{2\tau}\|\nabla h\|_F^2$ is constant *w.r.t.* $\mathcal{W}$, it only needs to solve the following projection problem,

$$\min_{\mathcal{W}} \frac{\tau}{2}\|\mathcal{W} - \mathcal{G}\|_F^2 + g(\mathcal{W}) \tag{31}$$

This problem has a unique global closed-form solution, that can be computed by Moreau Projection [22] as described in Algorithm 3.

Denote the optimal solution to is $\widehat{\mathcal{W}} = [\widehat{\mathbf{W}}^1, ..., \widehat{\mathbf{W}}^T]$. Then, its p-th term, $\widehat{\mathbf{W}}^p$, can be calculated by

$$\widehat{\mathbf{W}}^p = \begin{cases} \frac{\|\mathbf{G}^p\|_F - \zeta}{\|\mathbf{G}^p\|_F}\mathbf{G}^p & if\ \|\mathbf{G}^p\|_F - \zeta > 0 \\ 0 & otherwise \end{cases} \tag{32}$$

where $p \in \{1, 2, ..., T\}$.

The overall APG algorithm for solving the problem is summarized in Algorithm 4. More details and the convergence derivation of APG can be referred to [22].

3) Prediction of target pseudo labels: Given the updated model $\widehat{\mathcal{W}} = [\widehat{\mathbf{W}}^1, ..., \widehat{\mathbf{W}}^T]$ and corresponding feature indicator vectors $\mathcal{C} = [\mathbf{d}^1, ..., \mathbf{d}^T]$, the C-dimensional prediction output vector for the target data $\mathbf{x}$ can be obtained by:

$$\mathbf{f}(\mathbf{x}) = \sum_{p=1}^{T}(\varphi(\mathbf{x}) \odot \mathbf{d}^p)\,\widehat{\mathbf{W}}^p \in \mathbb{R}^C \tag{33}$$

And then, the pseudo label $\hat{y}$ can be predicted by

$$\hat{y} = \arg\max_{c \in \{1,...,C\}} f_c(\mathbf{x}) \tag{34}$$

where $f_c(\mathbf{x})$ is c-th entry of the output vector $\mathbf{f}(\mathbf{x}) \in \mathbf{R}^C$.

C. Computational Issues

The gradient computation in the APG algorithm is *w.r.t.* the selected features only. It takes $\mathcal{O}(TBn_s)$ cost in general where $TB \ll M$. Moreover, the score computation is very simple

Algorithm 4 Prediction Model Optimization

Initialization: Initialize the Lipschitz constant $L_p = L_{p-1}$, $L_0 = L_{max} = \lambda_{max}(\mathbf{A}\mathbf{A}^T)$ and set $\mathcal{W}^{-1} = \mathcal{W}^0$ by warm start, $\tau_0 = L_p$, $\mu \in (0, 1)$, parameter $\delta^{-1} = \delta^0 = 1$, and $k = 0$

1: Set $\mathcal{V}^k = \mathcal{W}^k + \frac{\delta^{k-1}-1}{\delta^k}(\mathcal{W}^k - \mathcal{W}^{k-1})$
2: $\tau = \mu^2\tau_k$
3: **repeat**
4: Set $\mathcal{G} = \mathcal{V}^k - \frac{1}{\tau}\nabla h(\mathcal{V}^k)$
5: Compute $\widehat{\mathcal{W}}$ by Algorithm 3
6: **if** $\mathcal{F}(\widehat{\mathcal{W}}) \leq \widetilde{\mathcal{F}}(\widehat{\mathcal{W}}, \mathcal{V}^k)$ **then**
7: Set $\tau_k = \tau$, stop, break
8: **else**
9: $\tau = \min\{\mu^{-1}\tau, L_{max}\}$
10: **until** $\mathcal{F}(\widehat{\mathcal{W}}) \leq \widetilde{\mathcal{F}}(\widehat{\mathcal{W}}, \mathcal{V}^k)$
11: Set $\mathcal{W}^{k+1} = \widehat{\mathcal{W}}$
12: Let $\delta^{k+1} = \frac{1+\sqrt{1+4(\delta^k)^2}}{2}$, and $k = k + 1$
13: Quit if the stopping condition is achieved. Otherwise, go to Step 1)
14: Let $L_p = \tau_k$ and return.

TABLE II
STATISTICS OF THE SIX BENCHMARK DATASETS

Dataset	Type	#Samples	#Features	#Classes	Subsets
USPS	Digit	1800	256	10	USPS
MNIST	Digit	2000	256	10	MNIST
COIL20	Object	1440	1024	20	COIL1, OIL2
PIE	Face	11554	1024	68	PIE1,...,PIE5
Office	Object	1410	800	10	A, W, D
Caltech	Object	1123	800	10	C

column-wise sum operation, which only takes $\mathcal{O}((n_s + n_t))$ cost. Totally, the algorithm will takes $\mathcal{O}(TBn_s + (n_s + n_t))$ cost. It is only linear relation to $TB \leq M$ and $n_s + n_t$, and has nothing to do with M. Therefore, it is very efficient for high-dimensional problems.

IV. Experimental Study

In this section, we conduct extensive experiments on six widely adopted benchmark datasets USPS, MNIST, COIL20, PIE, Office, and Caltech to evaluate the EMFS approach. All the experiments are conducted on a machine with the configurations: Win7, MATLAB 2017a, Intel i7 6700, 4 GHz CPU (8 cores) and 64G RAM.

A. Datasets

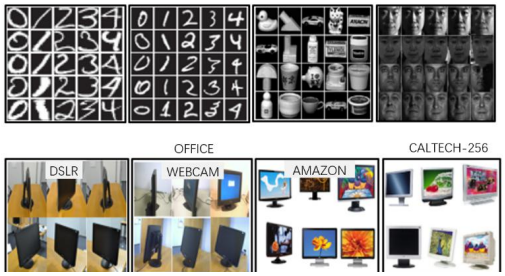


Fig. 2. Examples of six benchmark datasets

USPS has a training set with 7291 images and a test set with 2007 images of size 16×16.

MNIST consists of 60,000 training images and 10,000 testing images of size 28×28. They share 10 digital categories but have different size. Each image is rescaled to 16×16, and represented by a 256 pixels vector. Fig.2 shows that USPS obeys a distribution very different to MNIST. Following [9], one domain adaption dataset USPS→MNIST is formed by selecting 1,800 random images from USPS as source domain, and selecting 2,000 random images from MNIST as target domain. The source-target sequence is exchanged to form another data set MNIST→USPS.

COIL20 consists of 1,440 images of 20 objects. Each time the object rotates 5 degrees, an image is acquired so that each object has 72 images. Each image has 32×32 pixels, and each pixel has 256 gray levels. To construct domain adaption datasets, COIL20 is parted into two sets C1 and C2: C1 comprises the images in the orientations of $[0°, 85°] \cup [185°, 265°]$; C2 comprises the images in the orientations of $[90°, 175°] \cup [270°, 355°]$. In such manner, C1 and C2 will be subject to different distributions. The dataset C1→C2 is constructed by using C1 as the source data, and C2 as the target data. Switching source/target pair, another dataset C2→C1 is formed.

PIE [23] is a face database with 68 individuals, which has 41,368 images of size 32×32. The face images were captured by 13 cameras in different locations and 21 flashes of different illuminations and/or expressions. Five subsets, PIE1 (C05, left pose), P2 (C07, upward pose), P3 (C09, downward pose), P4 (C27, frontal pose), P5 (C29, right pose), are chosen. Two different subsets are selected as the source and target data respectively. Then, 5×4=20 domain adaption datasets can be constructed: P1→P2, P1→P3, ..., P5→P4. In such manner, the source and target data will be subject to very different distributions.

Office [24] has 4,652 images and 31 categories. It consists of 3 real-world domains, Amazon, Webcam, and DSLR.

Caltech-256 [25] contains 256 categories and 30,607 images. In experiments, we adopt the public Office+Caltech datasets released by Gong et al. [21]. In specific, by randomly selecting two different domains as the source domain and target domain respectively, 4×3=12 domain adaption datasets are constructed: A→W, A→D, A→C, ..., C →D.

B. Baseline Algorithms

EMFS are verified and compared with eight state-of-the-art baseline algorithms as follows:

No transfer:

- 1-Nearest Neighbor Classifier (NN)
- Principal Component Analysis (PCA)+NN

Transfer learning

- Geodesic Flow Kernel (GFK) [24]+NN
- Transfer Component Analysis (TCA) [4]+NN
- Transfer Subspace Learning (TSL) [8]+NN
- Joint Distribution Adaption(JDA) [9]+NN
- Maximum Independence Domain Adaption (MIDA) [7]+NN
- Feature Selection with MMD (f-MMD) [10]+NN

TABLE III
EXPLICIT FEATURES OF THE SIX BENCHMARK DATASETS

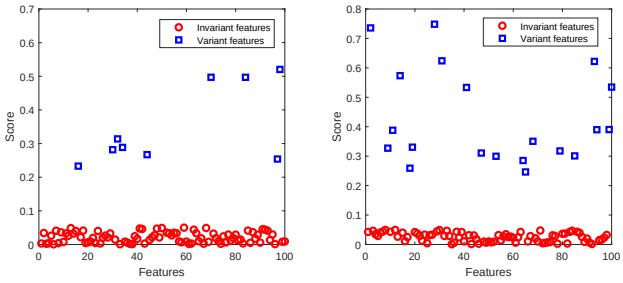
Dataset	#Original Feat.	#Gauss.Feat.	#Poly.Feat.
USPS	256	33153	33153
MNIST	256	33153	33153
COIL20	1024	525825	525825
PIE	1024	525825	525825
Office	800	321201	321201
Caltech-256	800	321201	321201

C. Parameter Settings

All baseline algorithms are trained on source domain, and tested on target domain. PCA, TSL, TCA, JDA, MIDA, f-MMD are executed as a dimensionality reduction or feature selection procedure on all data before NN. For the proposed algorithm, EMFS and EMFS+NN are both verified. In EMFS+NN, EMFS is executed as feature selection procedure before NN similar to baseline algorithms.

For the hyper-parameter, the degree of Gaussian and Polynomial kernel, we adopt the most widely used setting, $d = 2$. The statistic information of the datasets after explicit map is shown in Table III. Since the source and target domain follow different distributions, the optimal parameters can not be tuned by cross validation. Thus, the performance of all methods is estimated in the parameter space, and the best results are reported. EMFS involves the parameters: regularization parameter γ, and #selected features B in each iteration. We set $\gamma \in \{0.01, 0.1, 1, 10, 100\}$ and $B = 200$. For PCA, TSL, TCA, JDA and MIDA, we set #bases of the subspace $k \in [10, 20, ..., 200]$, and regularization parameter $\lambda \in \{0.01, 0.1, 1, 10, 100\}$. For f-MMD, the weight threshold value is set to $\lambda \in \{0.1, 0.2, 0.3, 0.4\}$. We use classification accuracy on target data as the evaluation metric.

D. Feature Identification Ability



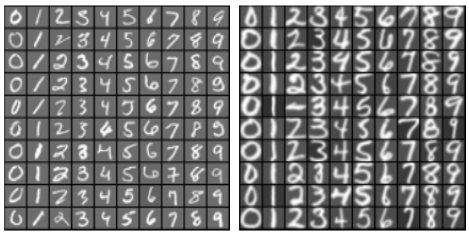
(a) The number of variant features $m_v = 10$ (b) The number of variant features $m_v = 20$

Fig. 3. Feature identification ability. Red circles illustrate invariant features, blue squares illustrate variant features. As shown in each subfigure, scores (scaled into [0,1]) for variant features are significantly higher than the invariant features

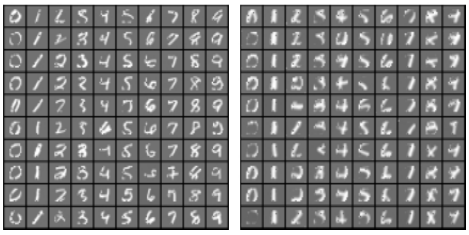
The purpose of the experiments in this section is to demonstrate whether EMFS can identify invariant/variant features across domains. One synthetic dataset is generated by the procedure [10]. Specifically, given n, the number of samples from each domain, m, the number of features, m_v, the number of variant features, and $(m - m_v)$, the number of invariant

features. For two domains, $(m-m_v)$ invariant features are sampled from $(m-m_v)$ a normal distribution with zero mean and unit variance. For the first domain, m_v variant features are sampled from m_v randomly picked distributions with zero mean and unit variance. For the second domain, these dimensions are sampled from the same m_v distributions but with linear shift 0.2 in sample mean. To produce class labels, m dimensional weight vector, $\mathbf{A} \in \mathbf{R}^{m\times 1}$ is drawn from the standard uniform distribution. Class labels are then achieved by a sign function to the sample $\mathbf{x}$, i.e. $y = \text{sign}(\mathbf{xA})$.

We generate 2 groups of domain adaption datasets with 10 and 20 variant dimensions by the above procedure, where 100 instances from each domain with m =100 features. Each subfigure in Fig.3 shows the scaled score of each feature. It can be observed that, the features which are assigned significantly larger weights by our algorithm, are indeed variant features.



(a) Original MNIST (b) Original USPS



(c) Selected MNIST (d) Selected USPS

Fig. 4. Digital image visualization before and after EMFS on original features. Some digital images of MNIST and USPS before/after feature selection. As shown in the figures, EMFS can identify invariant and variant features and remove variant ones effectively.

To show the identification result of our algorithm more directly, we show some digital images of MNIST and USPS dataset before and after our algorithm, respectively in Fig.4. As evident from Fig.4, the invariant pixel features of MNIST and USPS can be successfully identified.

E. Accuracy and Runtime

Tab. IV shows the classification accuracies, archived by all algorithms on 4 tasks, including 36 pairs of 'source→target' datasets in total. NN performs very poorly on 36 datasets, and only achieves the average accuracy of 37.4%. This validates that, EMFS effectively discover the invariant and discriminative features, and also demonstrates the benefits of modeling on such features instead of all features. Secondly, GFK achieves satisfactory performance on Office+Caltech datasets, but performs poorly on the other datasets. The reason is that, GFK 'requires a sufficiently small subspace to ensure smooth transmission of different subspaces along the geodesic stream, which, however, may lead to inaccurate representation of data. Thirdly, EMFS achieves much better accuracy than TCA. The reason is that TCA doesn't reduce the conditional distribution difference explicitly. Another limitation is that, all features, no matter variant and invariant, all are used in subspace learning. EMFS overcomes these limitations and only use the invariant and discriminative features as exchange pathway. Fourthly, EMFS outperforms TSL, which uses the kernel density estimation to measure the marginal distribution difference. Since TSL can better reduce the marginal distribution difference than TCA, it achieves better results than TCA. However, it does not explicitly reduce the conditional distribution difference. Lastly, EMFS also outperforms JDA, where subspace is learned on all features. Moreover, we test the performance of EMFS+NN, where EMFS is only used as feature selection by adding a NN classifier. We can find that, EMFS+NN is a little worse than EMFS. This indicates that, jointly learning the model and invariant features is helpful to build a more precise model. However, EMFS+NN is better than other baseline algorithms. This indicates that, EMFS can effectively discover the invariant discriminative features across domains.

TABLE IV
ACCURACY (%) OF EMFS AND OTHER BASELINE METHODS

Data	No Transfer		Transfer									
									EMFS+NN		EMFS	
	NN	PCA	f-MMD	MIDA	GFK	TCA	TSL	JDA	Poly	Gauss	Poly	Gauss
U→M	44.7	44.9	50.3	51.2	46.4	51.0	53.7	59.7	61.4	61.8	61.7	**62.2**
M→U	65.9	66.2	58.2	56.1	67.2	56.2	66.0	67.3	68.6	69.2	69.5	**69.8**
AVG	55.3	55.6	54.3	53.6	56.8	53.6	59.8	63.5	65.0	65.5	65.6	**66.0**
C1→C2	83.6	84.7	88.6	87.4	72.5	88.4	88.0	89.3	91.8	92.0	92.1	**92.2**
C2→C1	82.8	84.0	87.5	88.1	74.2	85.8	87.9	88.5	90.5	90.9	91.3	**91.6**
AVG	83.2	84.4	88.0	87.8	73.4	87.1	87.9	88.9	91.1	91.4	91.7	**91.9**
P1→P2	26.1	24.8	41.2	43.3	26.1	40.7	44.0	58.8	61.5	61.8	61.8	**62.1**
P1→P3	26.6	25.1	39.5	43.3	27.3	41.8	47.5	54.2	57.9	58.8	59.1	**59.7**
P1→P4	30.7	29.2	61.2	58.2	31.1	59.6	62.7	84.5	86.4	86.8	**87.3**	87.2
P1→P5	16.7	16.3	35.8	30.1	17.6	29.3	36.1	49.8	52.5	52.6	53.1	**53.2**
P2→P1	24.5	24.2	45.2	46.1	25.2	41.8	46.2	57.6	58.1	59.2	58.9	**59.2**
P2→P3	46.6	45.5	52.1	52.4	47.4	51.4	57.6	62.9	64.3	64.5	65.3	**66.1**
P2→P4	54.1	53.3	65.2	63.5	54.2	64.7	71.4	75.8	77.8	77.9	77.8	**77.9**
P2→P5	26.5	25.4	32.1	34.8	27.1	33.7	35.6	39.9	44.1	44.3	45.2	**45.8**
P3→P1	21.3	20.9	35.9	33.5	21.8	34.7	36.9	50.9	53.3	53.8	54.1	**54.7**
P3→P2	41.0	40.4	48.1	48.1	43.2	47.7	47.0	57.9	59.6	59.8	60.5	**61.1**
P3→P4	46.5	46.1	49.3	55.1	46.4	56.2	59.5	68.5	70.2	70.6	70.7	**71.3**
P3→P5	26.2	25.3	35.6	32.9	26.8	33.1	36.3	39.9	41.6	41.9	42.1	**42.5**
P4→P1	32.9	31.9	56.3	59.8	34.2	55.6	63.6	80.6	82.5	82.7	83.2	**83.3**
P4→P2	62.7	60.9	66.8	66.4	62.9	67.8	72.7	82.6	85.3	85.6	85.3	**85.6**
P4→P3	73.2	72.2	75.3	76.1	73.3	75.8	83.5	87.3	87.7	88.2	88.7	**88.9**
P4→P5	37.2	35.1	41.2	45.1	37.4	40.2	44.8	54.7	56.8	57.2	56.8	**57.2**
P5→P1	18.5	18.8	24.8	30.6	20.3	26.9	33.3	46.5	49.4	49.4	50.3	**50.4**
P5→P2	24.2	23.4	32.3	28.9	24.6	29.9	34.1	42.1	45.1	45.1	45.6	**45.8**
P5→P3	28.3	27.2	30.2	31.3	28.5	29.9	36.6	53.3	55.8	55.9	56.2	**56.5**
P5→P4	31.2	30.3	32.9	35.8	31.3	33.6	38.7	57.0	59.5	59.6	60.1	**61.6**
AVG	34.8	33.8	45.1	45.7	46.6	44.7	49.4	60.2	62.5	62.8	63.1	**63.5**
C→A	23.7	36.9	40.2	38.5	41.0	38.2	44.5	44.8	48.0	48.5	48.0	**48.5**
C→W	25.7	32.5	38.5	37.1	40.7	38.6	34.2	41.7	42.8	42.8	43.3	**44.1**
C→D	25.5	38.2	42.2	41.9	38.8	41.4	43.3	45.2	51.7	51.8	52.2	**52.6**
A→C	26.0	37.7	38.3	38.8	40.2	37.7	37.6	39.4	40.0	40.0	40.2	**41.4**
A→W	29.8	35.6	34.7	37.5	38.9	37.6	33.9	37.9	42.7	43.6	42.7	**43.6**
A→D	25.5	27.4	34.1	32.2	36.3	33.1	26.1	39.5	48.3	48.8	49.0	**49.6**
W→C	19.8	26.3	29.8	31.1	30.7	29.3	29.8	31.2	32.5	32.8	33.1	**33.5**
W→A	22.9	31.0	29.6	29.4	29.7	30.0	30.3	32.8	36.2	36.2	36.5	**36.7**
W→D	59.2	77.1	84.2	83.6	80.9	87.2	87.3	**89.2**	87.1	87.3	87.2	87.9
D→C	26.2	29.6	31.5	31.2	30.3	31.7	28.5	31.5	31.1	31.5	31.1	**32.4**
D→A	28.5	32.0	33.7	33.5	32.0	32.15	27.6	33.1	34.1	34.6	34.3	**35.3**
D→W	63.4	75.9	87.1	87.8	75.6	86.1	85.4	89.5	90.1	90.0	90.2	**90.6**
AVG	31.6	39.8	43.7	43.5	42.9	43.6	42.4	46.3	48.7	48.9	49.0	**49.7**
AVG(ALL)	37.4	39.8	57.8	57.6	54.9	57.2	59.8	64.7	66.8	67.1	67.4	**67.8**

TABLE V
RUNTIMES OF ALL ALGORITHMS

Method	Runtime (s)	Method	Runtime (s)
NN	4.85	TSL	1789
PCA	2.63	JDA	46.32
f-MMD	15046	MIDA	5.9
GFK	4.58	EMFS-Poly	6.2
TCA	3.80	EMFS-Gaus	6.24

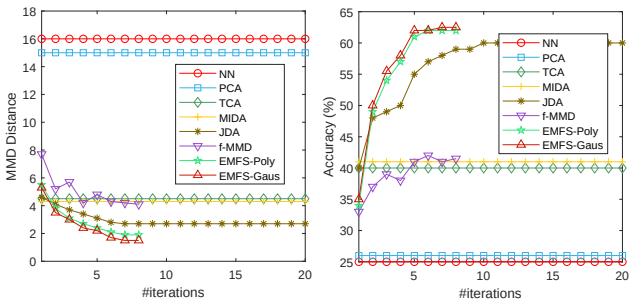
Table V shows the running time of all algorithms. It can

be observed that, although the dimension become very high after explicit feature map, EMFS can efficiently find invariant discriminative features. It is more efficient than MIDA, JDA, TSL and EMFS, and comparable with TCA and GFK. The reason lies that, in EMFS, only small part of selected features are involved in optimization.Since f-MMD involves a large number of kernel matrix computations, and all features are need used in the optimization, it takes the longest training time.

F. Performance Affected by Feature Types

In these four domain adaption tasks, the source and target domain vary from different aspects. Specifically, As shown in Fig.2, for MNIST and USPS, two domains have different object size, where the size of digits in USPS is larger than MNIST. For COIL, two domains have different rotation directions and degrees. PIE are multi-views of face images, captured under different conditions including locations and illuminations. Office and Caltech have different object appearance. Moreover, there's a lot of variation in the dimensionality, where, the dimensionality of every task is 33153, 525825, 525825 and 321201 respectively. Different dimensionality and variations lead that the difficulty levels of these domain adaptation tasks are different. For example, MNIST *vs.* USPS task is relatively easy due to the low dimensionality and simple variation across domains, while Office and Caltech is relative difficult since the dimensionality is very high and appearance change is more complex. However, although the difficulty levels of these tasks are different, EMFS achieves **66.0%**, **91.9%**, **63.5%** and **49.7%** respectively, which are much better than the best result **63.5%**, **88.9%**, **60.2%** and **46.3%** of other eight algorithms. This indicates that, EMFS is very stable for different domain adaption tasks, and consistently show the similar performance.

G. Distribution Distance Verification



(a) MMD distance w.r.t #iterations (b) Accuracy w.r.t #iterations

Fig. 5. Effectiveness verification: MMD distance and classification accuracy on the P1→ P2 dataset

The effectiveness of EMFS is further evaluated by comparing the distribution distance. All algorithms are executed on P1→P2 under optimal parameters. Then, the distribution distance of each algorithm is calculated on the resultant embedding or feature subset. For obtaining the true distance, the ground-truth target labels are used instead of the pseudo labels. The distribution distance and corresponding accuracy are illustrated in Fig.5 (a) and Fig.5 (b) respectively. It shows that: 1) Without feature representation or selection, the distribution distance of NN is the largest; 2) PCA only can reduce the distribution distance slightly, thus it helps a little with domain adaption; 3) TCA, MIDA, and f-MMD can significantly reduce the distribution distance by reducing the marginal distribution difference, so it performs better than PCA; 4) JDA can reduce both marginal and conditional distributions explicitly, so as to learn an effective subspace embedding and achieve better accuracy than TCA; 5) EMFS not only can reduce the marginal and conditional distribution difference, but also can remove those variant and non-discriminative features, thus enhance the domain adaption performance and achieve the best classification accuracy.

V. Conclusion

In this paper, we proposed a kernel-induced explicit feature selection method for domain adaption, called EMFS. EMFS represents the data by explicit feature map, to reveal the high-order invariant features, such that the distribution difference between two domains can be reduced efficiently. Moreover, By filtering non-discriminative features and matching conditional distribution, EMFS can achieve better generalized performance. Wide experiments on different applications demonstrate that, EMFS outperforms many other state-of-art algorithms such as TCA, TSL, MIDA, f-MMD and JDA. However, current EMFS is only studied for homogenous domain adaption setting, where two domains have the same feature space. In the future work, we will extend this method to the heterogeneous domain adaption.

Acknowledgment

This work is supported by National Science Foundation of China Grant No. 61572399, 61721002, 61532015, 61532004, 61472315; National Key Research and Development Program of China with grant number 2016YFB1000903; Shaanxi New Star of Science & Technology Grant No.2013KJXX-29; New Star Team of Xi'an University of Posts & Telecommunications; Provincial Key Disciplines Construction Fund of General Institutions of Higher Education in Shaanxi; the Data Science and Artificial Intelligence Center (DSAIR) at the Nanyang Technological University; ASTAR Thematic Strategic Research Programme (TSRP) Grant No. 1121720013; the Computational Intelligence Research Laboratory at NTU; ARC Future Fellowship FT130100746; ARC Linkage Project LP150100671; and ARC Discovery Project DP180100106.

References

[1] Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, vol.22, no.10, pp.1345-1359,2010

[2] J. Jiang and C. Zhai, Instance weighting for domain adaption in NLP, In ACL 2007, 2007, pp. 264-271.

[3] E. W. Xiang, B. Cao, D. H. Hu, and Q. Yang, Bridging domains using world wide knowledge for transfer learning, IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 6, pp. 770-783, 2010.

[4] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, Domain adaption via Transfer Component Analysis, IEEE Transactions on Neural Networks, vol. 22, no. 2, pp. 199-210, Feb. 2011.

[5] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, Unsupervised Visual Domain adaption Using Subspace Alignment, in 2013 IEEE International Conference on Computer Vision, 2013, pp. 2960-2967.
[6] J. T. Zhou, S. J. Pan, I. W. Tsang, and Y. Yan, Hybrid Heterogeneous Transfer Learning Through Deep Learning, in Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 2213 -2219.
[7] K. Yan, L. Kou, and D. Zhang, Learning Domain-Invariant Subspace Using Domain Features and Independence Maximization, IEEE Transactions on Cybernetics, vol. PP, no. 99, pp. 1-12, 2017.
[8] S. Si, D. Tao, and B. Geng, Bregman Divergence-Based Regularization for Transfer Subspace Learning, IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 7, pp. 929-942, Jul. 2010.
[9] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, Transfer Feature Learning with Joint Distribution adaption, in 2013 IEEE International Conference on Computer Vision, 2013, pp. 2200 -2207.
[10] S. Uguroglu and J. Carbonell, Feature selection for transfer learning, in Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 430 -442.
[11] Z. Hu, M. Lin, and C. Zhang, Dependent Online Kernel Learning With Constant Number of Random Fourier Features, IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 10, pp. 2464-2476, 2015.
[12] P. Drineas and M. W. Mahoney, On the Nystrom Method for Approximating a Gram Matrix for Improved Kernel-Based Learning, J. Mach. Learn. Res., vol. 6, pp. 2153-2175, 2005.
[13] T. Yang, Y. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou, Nystrom Method vs Random Fourier Features: A Theoretical and Empirical Comparison, in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 476-484.
[14] Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Scholkopf, and Alexander J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. In ISMB, pages 49-57, Fortaleza, Brazil, 2006.
[15] A. Cotter, J. Keshet, and N. Srebro, Explicit Approximations of the Gaussian Kernel, arXiv:1109.4603 [cs], Sep. 2011.
[16] A. J. Smola, A. Gretton, L. Song, and B. Scholkopf, A Hilbert space embedding for distributions, in Proceedings of the 18th International Conference on Algorithmic Learning Theory, Sendai, Japan, Oct. 2007, pp. 13-31.
[17] B. Quanz, J. Huan, and M. Mishra, Knowledge Transfer with Low-Quality Data: A Feature Extraction Issue, IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 10, pp. 1789-1802, Oct. 2012.
[18] M. Chen, K. Q. Weinberger, and J. C. Blitzer, Co-training for Domain adaption, in Proceedings of the 24th International Conference on Neural Information Processing Systems, USA, 2011, pp. 2456-2464.
[19] K. Liu, S. Wei, Y. Zhao, Z. Zhu, Y. Wei, and C. Xu, Accumulated reconstruction error vector (AREV): a semantic representation for cross-media retrieval, Multimedia Tools and Applications, pp. 1-16, 2014.
[20] A. Mutapcic and S. Boyd, Cutting-set Methods for Robust Convex Optimization with Pessimizing Oracles, Optimization Methods Software, vol. 24, no. 3, pp. 381-406, 2009.
[21] Q. Gu, Z. Li, and J. Han, Generalized Fisher Score for Feature Selection, in Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Arlington, Virginia, United States, 2011, pp. 266-273.
[22] M. Tan, I. W. Tsang, and L. Wang, Towards Ultrahigh Dimensional Feature Selection for Big Data, J. Mach. Learn. Res., vol. 15, no. 1, pp. 1371-1429, 2014.
[23] T. Sim, S. Baker, and M. Bsat, The CMU Pose, Illumination, and Expression (PIE) Database, in Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, Washington, DC, USA, 2002, p. 53-.
[24] K. Grauman, Geodesic Flow Kernel for Unsupervised Domain adaption, in Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 2012, pp. 2066-2073.
[25] G. Griffin, A. Holub, and P. Perona, Caltech-256 object category dataset. Technical report, Caltech, 2007. 5, Technical report, 2007.
[26] Li Y F, Tsang I W, Kwok T Y, et al. Tighter and Convex Maximum Margin Clustering[J]. Aistats B, 2011, 5:344-351.
[27] Kim, S. J., Boyd, S. A minimax theorem with applications to machine learning, signal processing, and finance. 2007, IEEE Conference on Decision and Control Vol.19, pp.751-758.

Wan-Yu Deng Received the BS degree in 2001, the MS degree in 2004 from Northwest Polytechnical University, China, and the Ph.D degree in the Department of Computer Science and Technology in 2010 from Xi'an Jiaotong University, China. He is Professor of the School of Computer Science, Xi'an University of Posts & Telecommunications, China. His research interests include machine learning, big data analysis and domain adaption.

Amaury Lendasse Amaury Lendasse was born in 1972, in Belgium. He received a M.S. degree in Mechanical Engineering from the Universite Catholique de Louvain (Belgium) in 1996, a M.S. in Control in 1997 and a Ph.D. He is now an Associate Professor at the University of Houston (USA). His research includes Big Data, time series prediction, noise variance estimation, determination of missing values in temporal databases, and functional neural networks

Yew-Soon Ong received the Ph.D. degree in University of Southampton, in 2003. He is Professor and Chair of the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interest in computational intelligence spans across memetic computing, evolutionary design and machine learning. He is an Associate Editor of the IEEE Computational Intelligence Magazine, IEEE Transactions Neural Networks and Learning Systems, IEEE Transactions on Evolutionary Computation, and many others.

Ivor Wai-Hung Tsang Ivor Wai-Hung Tsang received the PhD degree in computer science from the Hong Kong University of Science and Technology, in 2007. He is an Australian Future fellow and associate professor in University of Technology, Sydney. He received the prestigious IEEE Transactions on Neural Networks Outstanding 2004 Paper Award in 2006, the 2014 IEEE Transactions on Multimedia Prized Paper Award, and a number of best paper awards from reputable international conferences.

Lin Chen received the BS degree in Computer Science and Technology in 1999 from Shaanxi Normal University, the MS degree in Software Technology and Theory in 2005 from Northwest Polytechnical University, China. She is a associate professor in Xi'an University of Posts and Telecommunications, China. Her research interests include collaborative filtering and personalized service

Qing-Hua Zheng Received the BS degree in computer software in 1990, the MS degree in computer organization and architecture in 1993, and the PhD degree in system engineering in 1997, all from Xi'an Jiaotong University, China. He is a professor in Xi'an Jiaotong University, and serves as the vice-president of Xi'an Jiaotong University. His research areas include multimedia distance education, intelligent e-learning theory and algorithm.