“© 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Heterogeneous Feature Space based Task Selection Machine for Unsupervised Transfer Learning

Shan Xue[a,b], Jie Lu[a], Guangquan Zhang[a],
[a]Decision Systems & E-Service Intelligence Research Lab, Centre for Quantum Computation & Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology Sydney
shan.xue@student.uts.edu.au, jie.lu@uts.edu.au, guangquan.zhang@uts.edu.au

Li Xiong[b]

[b]School of Management
Shanghai University
Shanghai, China
xiongli8@shu.edu.cn

***Abstract*—Transfer learning techniques try to transfer knowledge from previous tasks to a new target task with either fewer training data or less training than traditional machine learning techniques. Since transfer learning cares more about relatedness between tasks and their domains, it is useful for handling massive data, which are not labeled, to overcome distribution and feature space gaps, respectively. In this paper, we propose a new task selection algorithm in an unsupervised transfer learning domain, called as Task Selection Machine (TSM). It goes with a key technical problem, i.e., feature mapping for heterogeneous feature spaces. An extended feature method is applied to feature mapping algorithm. Also, TSM training algorithm, which is main contribution for this paper, relies on feature mapping. Meanwhile, the proposed TSM finally meets the unsupervised transfer learning requirements and solves the unsupervised multi-task transfer learning issues conversely.**

Keywords-heterogeneous feature space; transfer learning; unsupervised learnig; multi-task learning; feature mapping

I. INTRODUCTION

The key problem of transfer learning [1] is task relatedness [2]. Determining whether the tasks are related or not which the information will be transferred among is very important or how they are correlated. And decide transfer or not information between two tasks, how many information to transfer and how the information will be transferred. Machine learning methods that enable any kind of communication between different tasks are performing transfer. The task from which the knowledge is extracted is called the *source task* and the novel task to which it is applied is the *target task* [3]. Literature in the field reports methods for performing transfer in two distinct types, functional and representational. In *functional transfer*, learning in the source and target happens simultaneously and it exploits implicit pressures from additional training patterns, via shared or common internal representations. In *representational transfer*, source and target learning occurs separately in time and an explicit representation is transferred from the source to the target. It cares most about learning the target task only.

Moreover, most methods of transfer learning implicitly assume that the source and target tasks are somehow related to each other. When, for example, the source task concerns training on female-only speech whilst the target task is to recognize speech from males only. In addition, most existing transfer learning algorithms assume that the feature spaces between the source and target domains are the same. However, in practice, it is useful to transfer knowledge across domains or tasks that have different feature spaces that is the so-called heterogeneous transfer learning.

The main objective for multi-task learning is to successfully learn all tasks with efficiency, e.g. via feature selection learning or instance learning [4], whose disadvantage is learner re-builds when new task comes. Transfer learning helps speeding up learning. However, it's not easy to tell target task from source tasks if we expect reducing computational complexity and when data are all unlabeled, i.e., to achieve unsupervised learning.

Traditional SVM methods [5] and Domain Transfer SVM (DTSVM) [6] are under a common supervised or semi-supervised training framework. In this paper, we are motivated by Domain Selection Machine (DSM) proposed for semi-supervised transfer learning [7] to try SVM based technique on multi-task learning, especially in unsupervised transfer learning research.

Therefore, the main contribution of this paper is we proposed a Task Selection Machine (TSM) for unsupervised multi-task transfer learning. While classical SVM is a supervised learning method for solving classification and regression problems, feature spaces in unsupervised transfer learning always differ. In order to make SVM based TSM for unsupervised transfer learning adaptive, a special process for feature mapping is defined in this paper too.

The rest of the paper is organized as follows. In section II, related works are reviewed. In section III, we detailed propose a novel unsupervised multi-task selection algorithm called Task Selection Machine (TSM) and explain the related properties. In Section IV concludes the paper and outlines some directions for future study.

II. RELATED WORK

Multi-task learning aims at learning jointly over N available sets, leading to a symmetric share of information [8]. This is particularly useful when each task has few data. The multi-task framework supposes that all the sets share the same feature space but present slightly different domains. Traditionally, one either assume that the set of labels for all the tasks are the same or that it is possible to access to an oracle mapping function that aligns the classes. Many techniques for multi-task learning have been published in machine learning domains especially based on some applications such as nature language processing, computer vision and image processing. Most of the works suppose multiple binary tasks and only few attempts have been done in the multiclass case without label correspondences [9].

One of the proposed algorithms considered multiclass cases and focused on multiple sources and a single target with domain shift and partially overlapping label sets, $\mathcal{Y}^s \cap \mathcal{Y}^t \neq \emptyset$ for all $(i, j) \in \{1, \cdots, N\}$ [8]. The difference in the domains can be caused by both different distributions $P^i(X) \neq P^j(X)$ and different feature spaces $\mathcal{X}^i \neq \mathcal{X}^j$. This algorithm was designed for extracting general information from all the sources, i.e., in multi-task fashion, and to use it when learning on a new target with a general advantage both on the known categories and on new ones, which is in domain adaptation and in transfer learning, respectively. Different from the classical multi-task learning, it broke the symmetry adding a transfer part to a target problem. Meanwhile, it overcame the transfer learning problem of evaluating the task relatedness leveraging on the possibility to extract a common useful knowledge from multiple sources. Therefore, when we set multi-task learning in a transfer learning domain, we can go beyond domain adaptation which does not cover the case of completely new classes in the target task.

However, because of multiple sources with eventually different features, previous hypothesis of relying on a flat average knowledge is not helpful in the case of tasks with partially overlapping label sets.

So far, the most common criterion to measure the distribution similarity is a nonparametric distances metric named maximum mean discrepancy (MMD) [10].

$$Disk_k(\mathcal{D}^{\mathrm{S}}, \mathcal{D}^{\mathrm{T}}) = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \phi(\boldsymbol{x}_i^S) - \frac{1}{n_T} \sum_{i=1}^{n_T} \phi(\boldsymbol{x}_i^T) \right\|^2, \quad (1)$$

where $\phi(\cdot)$ is the feature space mapping function. In worst case, even when you don't have similar tasks to transfer individuals from, genetic algorithm can be used to have performance improvement in a genetic algorithm task. In such case target task can be used as source task [2].

Feature selection helps in understanding data, reducing computation requirement, reducing the effect of curse of dimensionality and improving the predictor performance [11]. Feature selection techniques can be classified as wrapper methods and filter methods. Different assessment functions result in different feature selection criteria. Filter and wrapper techniques can be supervised as well as unsupervised. How to evaluate the significance of features becomes a key issue in feature selection. For example, ranking methods [12] are one of filter methods since they are applied before classification and clustering to filter out the less relevant variables. It is used due to their simplicity and good success is reported for practical applications. In filter method, the feature that has no influence on the cluster can be discarded. Ranking based filter method is proven to be computationally light, help avoid overfitting, and work well for certain datasets. To deal with to high dimensional data sets with structured input and structured output (SISO), where the SI means the input features are structured and the SO means the tasks are structured, a structured feature selection method in task relationship inference for multi-task learning was proposed in [13]. This work investigated a completely ignored problem in multi-task learning with SISO data: the interplay of structured feature selection and task relationship modeling.

Unlike supervised feature selection, in unsupervised feature selection the cluster label information is unavailable to guide the selection of minimal feature subset. Early unsupervised feature selection methods mainly use some evaluation indices to evaluate each individual feature or feature subset, and then select the top K features or the best feature subset. These indices evaluate the clustering performance, redundancy, information loss, sample similarity or manifold structure. These methods, however, are computationally expensive in searching. To reduce the computational cost, a feature clustering method is proposed in [14] to find the representative features based on feature similarity without searching. Motivated by the success of low-rank representation in subspace clustering, Zhu et. al. (2015) proposed a regularized self-representation (RSR) model for unsupervised feature selection [15], which is a simple yet very effective unsupervised feature selection method by exploiting the self-representation ability of features.

There are still few studies about feature selection focus on unsupervised transfer learning domain. Tran et al. (2013) introduced an efficient algorithm for ranking and selecting representation knowledge from a Restricted Boltzmann Machine (RBM) trained on a source domain to be transferred onto a target domain [16]. Self-taught learning [17], on the other hand, applies sparse coding to transfer the representations learned from source domain onto the target domain. Like self-taught, we are interested in unsupervised transfer learning using cross-task features.

Different from the other applications, in unsupervised feature based task selection, our goal is to identify a representative feature sets so that all the features can be well reconstructed to present tasks. Therefore, the crucial problems for this kind of research lie on unsupervised feature mapping, so that we can present heterogeneous tasks by corresponding feature spaces. In this paper, we try to do unsupervised feature mapping in an unsupervised transfer

learning framework to achieve computational complexity, efficiency, costs and adaptations.

III. An Unsupervised Transfer Learning Algorithm for Multi-task Selection Machine

A. Preliminary Concepts

In this section, we describe some fundamental information and measurements.

We denote the transpose of vector/matrix by using the superscript $'$. We also define $\mathbf{0}_n, \mathbf{1}_n \in \mathbb{R}^n$ as the column vectors of all zero and all ones, respectively. Moreover, we denote $\circ$ as an operator between two vectors, for example $\boldsymbol{a} \circ \boldsymbol{b} = (a_1 b_1, \cdots, a_n b_n)'$.

Let $X \in \mathbb{R}^{n \times m}$ be a data matrix, n be the number of tasks and define $\mathbb{N} = \{1, \cdots, n\}$, m be the maximum number of feature space $\mathbb{M} = \{1, \cdots, m\}$ and feature space for each sample differs. For each task $i \in \mathbb{N}$, we give the n samples $(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n) \in \mathbb{R}^{n \times m} \times \mathbb{R}$, where y_i is the label of $\boldsymbol{x}_i$. Each sample, $\boldsymbol{x}_i \to \boldsymbol{v}_i = \{v_{ij}|_{j=1}^{n_f}\}$, is denoted by a set of detectable features $\boldsymbol{v}_i$, where each feature space is defined by n_f, $n_f \in \mathbb{M}$. Based on this data structure, we wish to estimate $\mathbb{R}^{n \times m} \to \mathbb{R}$, which is hard in classical machine learning domain for the difficulty on feature selection.

B. Problem Definition

To deal with this unlabeled multiple tasks with different feature spaces problem, we introduce unsupervised transfer learning framework by denoting $\mathcal{T}^T = \{(\boldsymbol{x}_i^T, y_i^T)|_{i=1}^{n_T}\}$be the target task and $\mathcal{T}^S = \{(\boldsymbol{x}_i^S, y_i^S)|_{i=1}^{n_S}\}$ be the source tasks, where n_T is the number of $\mathcal{T}^T$, n_S is the number of $\mathcal{T}^S$. $\mathcal{D}^T$ and $\mathcal{D}^S$ contain the knowledge of $\mathcal{T}^T$ and $\mathcal{T}^S$ separately. In this issue, unsupervised transfer learning aims to help improve the learning of the target predictive function $f_t(\cdot)$ in $\mathcal{D}^T$ using the knowledge in $\mathcal{D}^S$ and $\mathcal{D}^T$, where $\mathcal{D}^T$ is different from $\mathcal{D}^S$ and labels y_i^T and y_i^S are not observable.

The underlying assumptions in this paper are:

- In transfer learning framework, $n_T \ll n_S$; and
- The functions $f^i(\cdot)$ for feature selection are related so that they all share a small set of features.

For the i-th sample $\boldsymbol{x}_i$, feature mapping (see Figure 1 and Algorithm 1) based sample clustering for each task is represented as

$$f^i(\boldsymbol{x}_i) = \sum_{j=1}^{n_f} a_j h_i(\boldsymbol{x}_i) = \sum_{j=1}^{n_f} a_j h_i(x_{ij}), \tag{2}$$

where $h_i(\cdot): \mathbb{R}^m \to \mathbb{R}$ are the features and $a_j \in \mathbb{R}$ are the regression parameters. For simplicity, we focus on non-linear features, that is, $h_i(\boldsymbol{x}_i) = \langle \varphi(\boldsymbol{v}), \varphi(\boldsymbol{x}_i) \rangle$, where $\boldsymbol{v} = \{v_j\}, v_j \in \mathbb{R}^m$. In addition, we assume that the vectors v_j are orthonormal. The function $f^i(\cdot)$ are non-linear as well, that is $f^i(\boldsymbol{x}_i) = \langle \varphi(\boldsymbol{u}), \varphi(\boldsymbol{x}_i) \rangle$, where $\varphi(\boldsymbol{u}) = \sum_j a_j \varphi(\boldsymbol{v}), \boldsymbol{u} = \{u_j\}, u_j \in \mathbb{R}^m$.

Algorithm 1 Multi-Task Feature Mapping

Input:
Unlabeled task samples $\{\boldsymbol{x}_i|_{i=1}^n\}$.
Output:
Feature spaces $\boldsymbol{v}_i$ for samples $\boldsymbol{x}_i$;
A feature vector $\boldsymbol{v} = \{v_j|_{j=1}^{n_f}\}$.
1: **Initialization:** Feature space $\boldsymbol{v} \leftarrow null$, $a_j \leftarrow 1$ and $i \leftarrow 1$.
2: **While** $i \le n$ **do**
3: $h_i(\boldsymbol{x}_i) \leftarrow \text{map } \boldsymbol{v}_i \text{ from } \boldsymbol{x}_i$.
4: $\boldsymbol{v}^{(i)} \leftarrow \boldsymbol{v}^{(i-1)} \cup \boldsymbol{v}_i$.
5: $i \leftarrow i + 1$.
6: **End While**
7: **Return** $\{\boldsymbol{v}_i|_{i=1}^n\}$ and $\boldsymbol{v} \leftarrow \boldsymbol{v}^{(i)}$.

We then present the unsupervised mutli-task selection algorithm (Algorithm 3). An objective TSM is proposed as follows and is described in Figure 2 as well,

$$\begin{aligned} \min_{\boldsymbol{s}, \boldsymbol{w}, \boldsymbol{\beta}, b, \boldsymbol{f}^i} & \frac{1}{2}(\|\boldsymbol{w}\|^2 + \lambda\|\boldsymbol{\beta}\|^2) \\ & + \frac{\theta}{2}\sum_{s=1}^{n_T} s_s \sum_{i=1}^{n_T} (f^i(\boldsymbol{x}_i^T) - f^s(\boldsymbol{x}_s^S))^2 \\ & + C\sum_{i=1}^{n_T} \ell_\epsilon(f^i(\boldsymbol{x}_i^T) - f_t(\boldsymbol{x}_i^T)), \end{aligned} \tag{3}$$

$$s.t. \sum_{s=1}^{n_S} s_s \ge 1,\ s_s \in \{0,1\}, \tag{4}$$

where $\ell_\epsilon(\cdot)$ is the ϵ-insensitive loss function,

$$\ell_\epsilon(\mathrm{a}) = \begin{cases} |a| - \epsilon, & if\,|a| > \epsilon \\ 0, & otherwise \end{cases} \tag{5}$$

$\boldsymbol{f}^i = (f^1(\boldsymbol{x}_1^T), \cdots, f^{n_T}(\boldsymbol{x}_{n_T}^T))'$ is a vector of decision values of unlabeled target task samples from the target cluster $f^i(\cdot)$, and $\lambda, \theta, C > 0$ are the regularization parameters.

C. Problem Solution

Solving Eq. (3) and Eq. (4), we present the feasible set of $\boldsymbol{s}$ as $\mathcal{M} = \{\boldsymbol{s}|\mathbf{1}_S'\boldsymbol{s} \ge 1, \boldsymbol{s} \in \{0,1\}^S\}$. We define $\tilde{\boldsymbol{w}} = (\boldsymbol{w}', \sqrt{\lambda}\boldsymbol{\beta}')'$ and $\tilde{\varphi}(\boldsymbol{x}_i^T) = (\varphi'(\boldsymbol{x}_i^T), \frac{1}{\sqrt{\lambda}}(\boldsymbol{s} \circ \boldsymbol{f}^s(\boldsymbol{x}_s^S))')'$, where $\boldsymbol{f}^s = (f^1(\boldsymbol{x}_s^S), \cdots, f^{ns}(\boldsymbol{x}_{n_S}^S))'$ is a vector of decision values

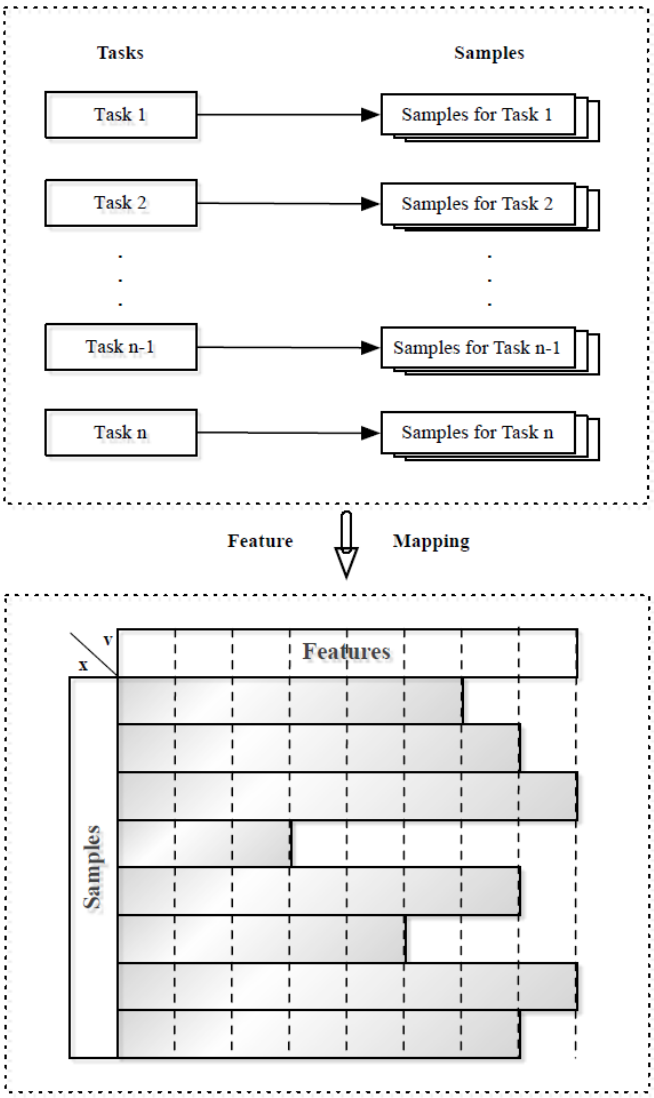

Figure 1. Feature Mapping.

of unlabeled source task samples from the source clusters $f^s(\cdot)$. Since ϵ-insensitive loss ℓ_ϵ is non-smooth, we transform the loss on the unlabeled target samples $\boldsymbol{x}_i^T$ in Eq. (3) into constraints in which the slake variables ξ_i and ξ_i^* are introduced. Then, we rewrite the optimization problem in Eq. (3) with condition Eq. (4) as follows,

$$\min_{\substack{\boldsymbol{s}\in\mathcal{M},\widetilde{\boldsymbol{w}},\\ b,\boldsymbol{f}^i,\xi_i,\xi_i^*}} \frac{1}{2}\|\widetilde{\boldsymbol{w}}\|^2 + \frac{\theta}{2}\sum_{i,s=1}^{n_T} s_s\left\|\boldsymbol{f}^i-\boldsymbol{f}^s\right\|^2 + C\sum_{i=1}^{n_T}(\xi_i+\xi_i^*) \tag{6}$$

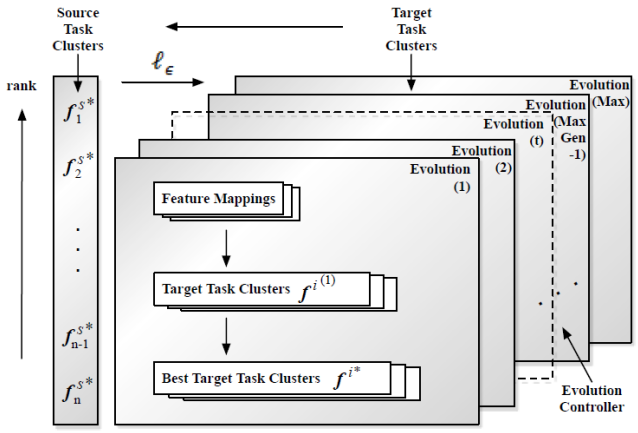

Figure 2. Task Selection Machine (TSM).

$$s.t. \quad \widetilde{\boldsymbol{w}}'\tilde{\varphi}(\boldsymbol{x}_i^T)+b-f^i(\boldsymbol{x}_i^T)\le\epsilon+\xi_i,\ \ \xi_i\ge 0 \tag{7}$$

$$f^i(\boldsymbol{x}_i^T)-\widetilde{\boldsymbol{w}}'\tilde{\varphi}(\boldsymbol{x}_i^T)-b\le\epsilon+\xi_i^*,\ \ \xi_i^*\ge 0 \tag{8}$$

We now switch the problem in Eq. (6) to a primal Lagrangian formulation, by introducing the dual variables $\boldsymbol{\alpha}=(\alpha_1,\cdots,\alpha_{n_T})'$ and $\boldsymbol{\alpha}^*=(\alpha_i^*,\cdots,\alpha_{n_T}^*)'$ for the constrains in Eq. (7) and (8), respectively.

$$\begin{aligned}
&L_P(\widetilde{\boldsymbol{w}},b,\boldsymbol{f}^i,\boldsymbol{\xi},\boldsymbol{\xi}^*,\boldsymbol{\alpha},\boldsymbol{\alpha}^*)\\
&=\frac{1}{2}\|\widetilde{\boldsymbol{w}}\|^2+\frac{\theta}{2}\sum_{i,s=1}^{n_T}s_s\left\|\boldsymbol{f}^i-\boldsymbol{f}^s\right\|^2+C\sum_{i=1}^{n_T}(\xi_i+\xi_i^*)\\
&\quad-\sum_{i=1}^{n_T}\alpha_i f^i(\boldsymbol{x}_i^T)(\widetilde{\boldsymbol{w}}'\tilde{\varphi}(\boldsymbol{x}_i^T)+b-\epsilon-\xi_i+1)\\
&\quad+\sum_{i=1}^{n_T}\alpha_i^* f^i(\boldsymbol{x}_i^T)(\widetilde{\boldsymbol{w}}'\tilde{\varphi}(\boldsymbol{x}_i^T)+b+\epsilon+\xi_i^*-1)\\
&=\frac{1}{2}\|\widetilde{\boldsymbol{w}}\|^2+\frac{\theta}{2}\sum_{i,s=1}^{n_T}s_s\left\|\boldsymbol{f}^i-\boldsymbol{f}^s\right\|^2+C\sum_{i=1}^{n_T}(\xi_i+\xi_i^*)\\
&\quad-\sum_{i=1}^{n_T}(\alpha_i-\alpha_i^*)f^i(\boldsymbol{x}_i^T)(\widetilde{\boldsymbol{w}}'\tilde{\varphi}(\boldsymbol{x}_i^T)+b)\\
&\quad-\sum_{i=1}^{n_T}(\alpha_i+\alpha_i^*)+\epsilon\sum_{i=1}^{n_T}(\alpha_i+\alpha_i^*)\\
&\quad+\sum_{i=1}^{n_T}(\alpha_i\xi_i+\alpha_i^*\xi_i^*)
\end{aligned} \tag{9}$$

Algorithm 2 Source Cluster Ranking

Input:
Source task feature clusters $\boldsymbol{f}^s$.
Output:
The rank-cluster vector $\hat{\boldsymbol{r}}_{n_S}$ for $\boldsymbol{s}$.
1: $\boldsymbol{r}_{n_S} \leftarrow$ Rank $\boldsymbol{f}^s$ as numbers from n_S (best) to 1 (worst).
2: **for all** r_s in $\boldsymbol{r}_{n_S}$ **do**
3: **If** $r_s \geq n_S - n_T + 1$**then**
4: $\hat{r}_s = 1$
5: **Else**
6: $\hat{r}_s = 0$
7: **End If**
8: **End for**
9: **Return** $\hat{\boldsymbol{r}}_{n_S} \leftarrow \{\hat{r}_s\}$.

Then, requiring that the gradient of L_P with respect to $\widetilde{\boldsymbol{w}}, b, \boldsymbol{\xi}$ and $\boldsymbol{\xi}^*$ vanish, we obtain the Karush-Kuhn-Tucker (KKT) conditions for the primal Lagrangian as,

$$\widetilde{\boldsymbol{w}} = \sum_{i=1}^{n_T} (\alpha_i - \alpha_i^*) f^i(\boldsymbol{x}_i^T) \tilde{\varphi}(\boldsymbol{x}_i^T), \tag{10}$$

$$\sum_{i=1}^{n_T} (\alpha_i - \alpha_i^*) f^i(\boldsymbol{x}_i^T) = 0, \tag{11}$$

$$\sum_{i=1}^{n_T} \alpha_i = \sum_{i=1}^{n_T} \alpha_i^* = C. \tag{12}$$

Meanwhile, we have a relationship between $f_t(\boldsymbol{x}_i^T)$ and $\boldsymbol{f}^i$ in this optimization problem,

$$\sum_{i=1}^{n_T} (\alpha_i - \alpha_i^*) f_t(\boldsymbol{x}_i^T) = \theta \sum_{i,s=1}^{n_T} s_s(\boldsymbol{f}^i - \boldsymbol{f}^s). \tag{13}$$

Since these are equality constraints in the dual formulation, we can substitute them into Eq. (9) to give a dual problem,

$$L_D(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*)$$

$$= -\frac{1}{2} \sum_{i,j=1}^{n_T} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) f^i(\boldsymbol{x}_i^T) f^i(\boldsymbol{x}_j^T) \tilde{\varphi}(\boldsymbol{x}_i^T) \tilde{\varphi}(\boldsymbol{x}_j^T)$$

$$+(1-\epsilon) \sum_{i=1}^{n_T} (\alpha_i - \alpha_i^*), \tag{14}$$

where $\|\widetilde{\boldsymbol{w}}\|^2 = \widetilde{\boldsymbol{w}}'\widetilde{\boldsymbol{w}} = \sum_{i,j=1}^{n_T} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) f^i(\boldsymbol{x}_i^T) f^i(\boldsymbol{x}_j^T) \tilde{\varphi}(\boldsymbol{x}_i^T) \tilde{\varphi}(\boldsymbol{x}_j^T)$, the condition of L_D is $\sum_{i=1}^{n_T} (\alpha_i - \alpha_i^*) f^i(\boldsymbol{x}_i^T) = 0$, and $\boldsymbol{0}_{n_T} \leq \boldsymbol{\alpha} - \boldsymbol{\alpha}^* \leq C\boldsymbol{1}_{n_T}$.

From Eq. (6) to Eq. (14), an optimization problem is converted as

$$\min_{\boldsymbol{s}, \boldsymbol{f}^i} \max_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} -\frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)' \boldsymbol{K} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) - \epsilon \boldsymbol{1}'_{n_T} (\boldsymbol{\alpha} + \boldsymbol{\alpha}^*)$$

$$+ \min_{\boldsymbol{s}, \boldsymbol{f}^i} \frac{\theta}{2} \hat{\boldsymbol{r}}'_{n_S} \boldsymbol{s} \boldsymbol{B}' \boldsymbol{B}, \tag{15}$$

$$s.t. \quad \boldsymbol{1}'_{n_T} \boldsymbol{\alpha} = \boldsymbol{1}'_{n_T} \boldsymbol{\alpha}^* \tag{16}$$

$$\boldsymbol{0}_{n_T} \leq \boldsymbol{\alpha} - \boldsymbol{\alpha}^* \leq C \boldsymbol{1}_{n_T}' \tag{17}$$

where $\hat{\boldsymbol{r}}_{n_S} = \{\hat{r}_p|_{p=1}^{n_S}, \sum_p \hat{r}_p = n_T\}$ is a $0/1$ vector valued by $\boldsymbol{f}^s$'s ranking, that is, mark the 1 *to* n_T –th best $\boldsymbol{f}^s$ as 1 and 0 otherwise, $\boldsymbol{B}$ consists of the bias from each target cluster $\boldsymbol{f}^i$ and source cluster $\boldsymbol{f}^s$, and $\boldsymbol{K}$ is denoted as a kernel matrix with each entry as

$$\boldsymbol{K}(\boldsymbol{x}_i^T, \boldsymbol{x}_j^T) = f^i(\boldsymbol{x}_i^T)' \tilde{\varphi}(\boldsymbol{x}_i^T)' \tilde{\varphi}(\boldsymbol{x}_j^T) f^i(\boldsymbol{x}_i^T). \tag{18}$$

Algorithm 3 Task Selection Machine (TSM)

Input:
Unlabeled target task samples $\{\boldsymbol{x}_i^T|_{i=1}^{n_T}\}$;
Source clusters $\boldsymbol{f}^s$;
Maximum Evolution Generation $MaxGen$.
Output:
The target task clusters $\mathcal{F} = \{\boldsymbol{f}^i|_{i=1}^{n_T}\}$.
1: **Initialization:** $t \leftarrow 1, \boldsymbol{s}_t \leftarrow \boldsymbol{1}_{n_S}$ and $\mathcal{F} \leftarrow null$.
2: **While** $i \leq n_T$ **do**
3: $\boldsymbol{K} \leftarrow$ Via Eq. (18).
4: $\hat{\boldsymbol{r}} \leftarrow$Calculate from **Algorithm 2**.
5: $\boldsymbol{B} \leftarrow$ Bias between $\boldsymbol{f}^i$ and $\boldsymbol{f}^s$.
6: **While** $t \leq MaxGen$ **do**
7: $t \leftarrow 1$
8: $g^{(t)}(\boldsymbol{s}) \leftarrow$ Calculate using $\boldsymbol{\alpha}^{(t)}, \boldsymbol{\alpha}^{*(t)}$.
9: **If** $\max_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*}$ in Eq. (15) decreases **then** break
10: **End While**
11: $\boldsymbol{f}^{i^*} \leftarrow$Sort out the best target cluster who minimise Eq. (15).
12: $\mathcal{F} \leftarrow$ Add $\boldsymbol{f}^{i^*}$.
13: **Update** $\boldsymbol{s}, \boldsymbol{f}^s, \boldsymbol{f}^i$ and $\tilde{\varphi}$ **by** transfer the $\boldsymbol{f}^{i^*}$'s target task as a source task.
14: $i \leftarrow i - 1$.
15: **End While**
16: **Return** $\mathcal{F}$.

In Algorithm 3, when $\boldsymbol{s}$ fixed in Eq. (15) for each evolution, we solve the optimization problem for $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$ until $\max_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} -\frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)'\boldsymbol{K}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) - \epsilon \mathbf{1}'_{n_T}(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*)$ decreases. Observing that $\min_{\boldsymbol{s}, \boldsymbol{f}^i} \frac{\theta}{2}\hat{\boldsymbol{r}}'_{n_S}\boldsymbol{s}\boldsymbol{B}'\boldsymbol{B}$ is general optimization problem which also related to Algorithm 1, we directly using evolution algorithm to get $\min_{\boldsymbol{s}, \boldsymbol{f}^i}$.

With the learned dual variables $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$, we use the target decision function in Eq. (19) to predict new target tasks.

$$f_t(\boldsymbol{x}_i^T) = \sum_{s=1}^{n_S} s_s \beta_s f^s(\boldsymbol{x}_i^S) + \boldsymbol{w}'\varphi(\boldsymbol{x}_i^T) + b, \qquad (19)$$

where $\sum_{s=1}^{n_S} s_s \beta_s f^s(\boldsymbol{x}_i^S)$ is a weighted combination of source clusters based on multiple features selection, $\varphi(\cdot)$ is a feature mapping function that map $\boldsymbol{x}_i$'s features into $\varphi(\boldsymbol{x}_i)$, $\boldsymbol{w}$ is a weight vector, and b is a bias term.

IV. Conclusions and Future Work

In this paper, we proposed a novel algorithm on Task Selection Machine (TSM) and solved the feature mapping problem by extending feature selection methods and also be trained in TSM. The scenario for this algorithm is general in real world situations that the disordered knowledge chosen by feature mapping will be finally transferred to acquirable knowledge. Currently, our work is set in unsupervised multiple tasks transfer learning. Through distribute clusters for each task, we make detection on relatedness between target tasks and source tasks, so that target tasks who first be trained are selected. The advantage for distinguishing clusters is avoiding sample distribution differences.

We can therefore claim that the proposed TSM can solve the problems of feature mapping for TSM in heterogeneous feature space which is mainly based on an unsupervised transfer learning framework. In future work, we are interested in learning and transferring high-level features for a specific domain, and considering some mutli-distribution situations.

Acknowledgment

This work is supported by the Australian Research Council (ARC) under discovery grant DP140101366, Shanghai Education Commission (No. 14ZS085), and Education Ministry of China (No. 12YJA630158).

References

[1] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer Learning using Computational Intelligence: A Survey," Knowledge-Based Systems, 2015.

[2] B. Koçer and A. Arslan, "Genetic transfer learning," Expert Systems with Applications, vol. 37, pp. 6997-7002, 2010.

[3] M. Kohli, G. D. Magoulas, and M. S. Thomas, "Transfer learning across heterogeneous tasks using behavioural genetic principles," in Computational Intelligence (UKCI), 2013 13th UK Workshop on, 2013, pp. 151-158.

[4] C. Wang, J. Lu, and G. Zhang, "Integration of ontology data through learning instance matching," in Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on, 2006, pp. 536-539.

[5] L. Duan, D. Xu, and S.-F. Chang, "Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach," in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 2012, pp. 1338-1345.

[6] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank, "Domain transfer svm for video concept detection," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 2009, pp. 1375-1381.

[7] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, pp. 770-787, 2010.

[8] T. Tommasi, N. Quadrianto, B. Caputo, and C. H. Lampert, "Beyond dataset bias: Multi-task unaligned shared knowledge transfer," in Computer Vision–ACCV 2012, ed: Springer, 2013, pp. 1-15.

[9] N. Quadrianto, J. Petterson, T. S. Caetano, A. J. Smola, and S. Vishwanathan, "Multitask learning without label correspondences," in Advances in Neural Information Processing Systems, 2010, pp. 1957-1965.

[10] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," 2014.

[11] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," Computers & Electrical Engineering, vol. 40, pp. 16-28, 2014.

[12] L. Laporte, R. Flamary, S. Canu, S. Déjean, and J. Mothe, "Nonconvex regularizations for feature selection in ranking with sparse svm," Neural Networks and Learning Systems, IEEE Transactions on, vol. 25, pp. 1118-1130, 2014.

[13] H. Fei and J. Huan, "Structured feature selection and task relationship inference for multi-task learning," Knowledge and information systems, vol. 35, pp. 345-364, 2013.

[14] P. Mitra, C. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," IEEE transactions on pattern analysis and machine intelligence, vol. 24, pp. 301-312, 2002.

[15] P. Zhu, W. Zuo, L. Zhang, Q. Hu, and S. C. Shiu, "Unsupervised feature selection by regularized self-representation," Pattern Recognition, vol. 48, pp. 438-446, 2015.

[16] S. N. Tran and A. d. A. Garcez, "Adaptive Feature Ranking for Unsupervised Transfer Learning," arXiv preprint arXiv:1312.6190, 2013.

[17] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in Proceedings of the 24th international conference on Machine learning, 2007, pp. 759-766.