Shared Structure Learning for Multiple Tasks with Multiple Views

Xin Jin[1,2], Fuzhen Zhuang[1], Shuhui Wang[1], Qing He[1], and Zhongzhi Shi[1]

[1] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
{jinx,zhuangfz,heq,shizz}@ics.ict.ac.cn,
wangshuhui@ict.ac.cn

Abstract. Real-world problems usually exhibit dual-heterogeneity, i.e., every task in the problem has features from multiple views, and multiple tasks are related with each other through one or more shared views. To solve these multi-task problems with multiple views, we propose a shared structure learning framework, which can learn shared predictive structures on common views from multiple related tasks, and use the consistency among different views to improve the performance. An alternating optimization algorithm is derived to solve the proposed framework. Moreover, the computation load can be dealt with locally in each task during the optimization, through only sharing some statistics, which significantly reduces the time complexity and space complexity. Experimental studies on four real-world data sets demonstrate that our framework significantly outperforms the state-of-the-art baselines.

Keywords: Multi-task Learning, Multi-view Learning, Alternating Optimization.

1 Introduction

In many practical situations, people need to solve a number of related tasks, and multi-task learning (MTL) [5–7, 20] is a good choice for these problems. It learns multiple related tasks together so as to improve the performance of each task relative to learning them separately. Besides, many problems contain different "kinds" of information, that is, they include multi-view data. Multi-view learning (MVL) [3, 8, 19] can make better use of these different views and get improved results. However, many real-world problems exhibit dual-heterogeneity [14]. To be specific, a single learning task might have features in multiple views (i.e., feature heterogeneity); different learning tasks might be related with each other through one or more shared views (features) (i.e., task heterogeneity). One example is the web page classification problem. If people want to identify whether the web pages from different universities are course home pages, then classifying each university can be seen as a task. Meanwhile, every web page has different kinds of features, one kind is the content of the web page, and the

H. Blockeel et al. (Eds.): ECML PKDD 2013, Part II, LNAI 8189, pp. 353–368, 2013.

other kind is the anchor text attached to hyperlinks pointing to this page, from other pages on the web. Such problem type is Multi-task learning with Multiple Views (MTMV). However, traditional multi-task learning or multi-view learning methods are not very suitable as they cannot use all the information contained in the problem.

In supervised learning, given a labeled training set and *hypothesis space* $\mathcal{H}$, the goal of *empirical risk minimization* (ERM) is to find a predictor $f \in \mathcal{H}$ that minimizes empirical error. The error of the best predictor learned from finite sample is called the *estimation error*. The error of using a restricted $\mathcal{H}$ is often referred to as the *approximation error*, from the structure risk minimization theory [21]. One needs to select the size of $\mathcal{H}$ to balance the trade-off between approximation error and estimation error. This is typically done through model selection, which learns a set of predictors from a set of candidate hypothesis spaces $\mathcal{H}_\theta$, and then chooses the best one. When multiple related tasks are given, learning the structure parameter Θ in the predictor space becomes easier [1, 6]. Also, different tasks can share information through the structure parameter Θ, i.e., for each task t, $f_t \in \mathcal{H}_{t,\Theta}$. MTMV learning methods that can make full use of information contained in multiple tasks and multiple views are proposed [14, 25]. The transductive algorithm *IteM*^{2} [14] can only deal with nonnegative feature values. *regMVMT* algorithm [25] shares the label information among different tasks by minimizing the difference of the prediction models for different tasks. Without prior knowledge, simply restricting all the tasks are similar seems inappropriate. This paper assumes that multiple tasks share a common predictive structure, and they can do model selection collectively. Compared to other methods to learn good predictive structures, such as data-manifold methods based on graph structure, our method can learn some underlying predictive functional structures in hypothesis space, which can characterize better predictors.

To facilitate information sharing among different tasks on multi-view representation, in this paper, we propose an efficient inductive convex shared structure learning for MTMV problem (*CSL-MTMV*). Our method learns shared predictive structures on hypothesis spaces from multiple related tasks that have common views; consequently, all tasks can share information through the shared structures. In this way, the strict assumption in the previous MTMV methods that all the tasks should be similar can be discarded. We assumed that the underlying structure is a shared low-dimensional subspace, and a linear form of feature map is considered for simplicity. Furthermore, it uses the prediction consistency among different views to improve the performance. Besides, some tasks may not have all the views in many real applications. To deal with missing views, a direct extension of our algorithm is provided. Our method is more flexible than previous inductive MTMV algorithm *regMVMT* [25]. Specifically, *regMVMT* can be seen as a special case of *CSL-MTMV*, which means our approach is more generalized and has a potential to get better results. In addition, different from *regMVMT*, our method decouples different tasks during the model optimization, which significantly reduces the time complexity and space complexity. Therefore, our method is more scalable for problems with large number of tasks.

The rest of this paper is organized as follows. A brief review of related work is given in Section 2. The MTMV problem definition and some preliminary works are presented in Section 3. Our shared structure learning framework for MTMV problem and a convex relaxation algorithm are described in Section 4. To demonstrate the effectiveness of our algorithm, some experimental results are shown in Section 5. Conclusion is provided in Section 6.

2 Related Work

Currently, there are only a few researches on multi-task problem with multi-view data (MTMV). The traditional multi-task learning and multi-view learning methods also provide some insights for the MTMV problem. In the following, a brief description of these methods is given.

Multi-Task Learning (MTL). Multi-task learning conducts multiple related learning tasks simultaneously so that the label information in one task can be used for other tasks. The earliest MTL method [5] learns a shared hidden layer representation for different tasks. Supposing that all the tasks are similar, a regularization formulation is proposed for MTL [11]. MTL can be modeled by stochastic process methods, such as [20, 24]. Multi-task feature learning learns a low-dimensional representation which is shared across a set of multiple related tasks [2, 15]. To deal with outlier tasks, a robust multi-task learning algorithm is proposed [7]. The methods to learn predictive structures on hypothesis spaces from multiple learning tasks are also proposed [1, 6].

Multi-View Learning (MVL). The basic idea of MVL is making use of the consistency among different views to achieve better performance. One of the earliest works on multi-view learning is co-training algorithm [3], which uses one view's predictor to enlarge the training set for other views. Nigam and Ghani compared co-training, EM and co-EM methods, and showed that co-EM algorithm is the best among the three approaches [17]. Some improvements of co-training algorithm are also proposed [16, 23]. Other methods are based on co-regularization framework. Sindhwani et al. [18] proposed a learning framework for multi-view regularization. *SVM-2K* [12] is a method which uses kernels for two views learning. Sindhwani and Rosenberg [19] constructed a single Reproducing Kernel Hilbert Spaces (RKHSs) with a data-dependent "co-regularization" norm that reduces MVL to standard supervised learning. Chen et al. [8] presented a large-margin learning framework to discover a predictive latent subspace representation shared by multiple views.

Multi-Task Learning with Multiple Views (MTMV). He and Lawrence [14] proposed a graph-based framework which takes full advantage of information among multiple tasks and multiple views, and an iterative algorithm ($IteM^2$) is proposed to optimize the model. The framework is transductive which cannot predict the unseen samples. It can only deal with problems with nonnegative feature values. *regMVMT* [25] uses co-regularization to obtain functions that are consistent with each other on the unlabeled samples for different views.

Across different tasks, additional regularization functions are utilized to ensure the learned functions are similar. The assumption that all the tasks are similar to each other may not be appropriate. Different tasks are coupled in the computation process of *regMVMT* algorithm, making the model becoming more complex and requires more memory to store the data.

3 Preliminaries

3.1 MTMV Problem Definition

Notations. In this paper, $[m : n]$ $(n > m)$ denotes a set of integers in the range of m to n inclusively. Let $\mathbb{S}_+$ be the subset of positive semidefinite matrices. Denote $A \preceq B$ if and only if $B - A$ is positive semidefinite. Let $\mathrm{tr}(X)$ be the trace of X, and X^{-1} be the inverse of matrix X. $\|\cdot\|$ denotes ℓ_2 norm of a vector. Unless specified otherwise, all vectors are column vectors.

In this part, a formal introduction of MTMV problem is given. The problem definition is very similar to [14, 25]. Suppose that the problem includes T tasks and V views in total. Also, N labeled and M unlabeled data samples are given. Usually, the labeled examples are insufficient while the unlabeled samples are abundant, i.e. $M \gg N$. For each task $t \in [1 : T]$, there are n_t and m_t labeled and unlabeled examples, thus we have $N = \sum_t n_t$ and $M = \sum_t m_t$. Let d_v be the number of features in the view $v \in [1 : V]$, and denote $D = \sum_v d_v$.

The feature matrix $X_t^v \in \mathbb{R}^{n_t \times d_v}$ is used to denote the labeled samples in task t for view v, the corresponding unlabeled examples is denoted $P_t^v \in \mathbb{R}^{m_t \times d_v}$. Let $y_t \in \{1, -1\}^{n_t \times 1}$ be the label vector of the labeled examples in the task t. $X_t = (X_t^1, X_t^2, \ldots, X_t^V)$, and $P_t = (P_t^1, P_t^2, \ldots, P_t^V)$ are concatenated feature matrices of the labeled and unlabeled examples for task t, respectively. It is common that in some applications not all tasks have features available from all the V views, so an indicator matrix $I_{id} \in \{1, 0\}^{T \times V}$ is used to mark which view is missing from which task, i.e. $I_{id}(t, v) = 0$ if the task t does not contain v-th view, and $= 1$ otherwise. This notation can only handle "structured" missing views [25] in the sense that if a view is present in a task, it is present in all the samples in the task; if a view is missing from a task, it is missing in all the samples in the task. Throughout the paper we use subscripts to denote tasks and superscripts to denote views. So the goal of this paper is to leverage the label information from all the tasks to help classify the unlabeled examples in each task, as well as use the consistency among different views of a single task to improve the performance.

3.2 Shared Structure Learning for MTL

Shared structure learning has been successfully used in single view multi-task learning (MTL) problems [1, 6], that is, $V = 1$ in the MTMV problem described in Section 3.1. In MTL, suppose the dimension of the feature space is d, and the objective is to learn linear predictors $f_t(x) = u_t^\top x$, for $t \in [1 : T]$, where u_t is

the weight vector for the t-th task. For shared structure learning, it is assumed that the underlying structure is a shared low-dimensional subspace, and a linear form of feature map is considered for simplicity. The predictors $\{f_t\}_{t=1}^{T}$ can be learned simultaneously by exploiting a shared feature space. Formally, the prediction function f_t can be expressed as:

$$f_t(x) = u_t^\top x = w_t^\top x + z_t^\top \Theta x \tag{1}$$

where the structure parameter Θ takes the form of an $h \times d$ matrix with orthonormal rows, i.e., $\Theta\Theta^\top = I$.

In [6], an improved alternating structure optimization (iASO) formulation is given:

$$\min_{\{u_t, z_t\}, \Theta\Theta^\top = I} \sum_{t=1}^{T} \left(\frac{1}{n_t} \sum_{i=1}^{n_t} L\left(f_t(x_{t,i}), y_{t,i}\right) + g_t(u_t, z_t, \Theta) \right) \tag{2}$$

where $g_t(u_t, z_t, \Theta)$ is the regularization function defined as:

$$g_t(u_t, z_t, \Theta) = \alpha \|u_t - \Theta^\top z_t\|^2 + \beta \|u_t\|^2. \tag{3}$$

The regularization function in Eq.(3) controls the task relatedness (via the first component) as well as the complexity of the predictor functions (via the second component) as commonly used in traditional regularized risk minimization formulation for supervised learning. α and β are pre-specified coefficients, indicating the importance of the corresponding regularization component. This formulation provides the foundation for our MTMV learning methods.

4 Shared Structure Learning for MTMV Problem

4.1 Shared Structure Learning Framework for MTMV Problem

A straightforward way to use the single view multi-task learning (MTL) methods described in Section 3.2 is as follows. First, the prediction model for each view data is learned individually, so the MTMV problem can be divided into V MTL problems. Then, a model for each view v in each task t is acquired, represented by $f_t^v(x_t^v)$ with the following formulation:

$$f_t^v(x_t^v) = u_t^{v\top} x_t^v = w_t^{v\top} x_t^v + z_t^{v\top} \Theta^v x_t^v, \tag{4}$$

where u_t^v, w_t^v and z_t^v have similar meanings as in Eq.(1), structure parameter Θ^v represents the low-dimensional feature map for view v across different tasks. The basic assumption underlying multi-view learning for a single task is that the multiple views are conditionally independent and the predictive model of each view can be used to make predictions on data examples, then the final models are obtained according to these models. Without prior knowledge on which view contributes more to the final models than other views, it is often assumed that all views contribute equally, as described in [19]. The final model for task t in

MTMV problem is obtained by averaging the prediction results from all view functions as follows:

$$f_t(x_t) = \frac{1}{V}\sum_{v=1}^{V} f_t^v(x_t^v), \tag{5}$$

where $x_t = [{x_t^1}^\top, {x_t^2}^\top, \ldots, {x_t^V}^\top]^\top$ is the concatenated feature vector of the samples for task t.

However, in MTMV problem, it is worthwhile to make better use of the information contained in different views, not just only decompose into separate MTL problems. The models built on each single view will agree with one another as much as possible on unlabeled examples. Co-regularization is a technique to enforce such model agreement on unlabeled examples. Adding this into the model, we obtain the following formulation :

$$\min_{\{u_t^v, z_t^v, \Theta^v\}, \Theta^v({\Theta^v}^\top)=I} \sum_{t=1}^{T}\sum_{v=1}^{V}\left(\frac{1}{n_t}\sum_{i=1}^{n_t} L\left(f_t^v(x_{t,i}^v), y_{t,i}\right) + g_t^v(u_t^v, z_t^v, \Theta^v) + \gamma\frac{1}{m_t}\sum_{j=1}^{m_t}\sum_{v'\neq v}\left(f_t^{v'}(p_{t,j}^{v'}) - f_t^v(p_{t,j}^v)\right)^2\right) \tag{6}$$

where L is the empirical loss function, $x_{t,i}^v$ is the feature representation for the v-th view of i-th labeled sample in task t, $p_{t,j}^v$ ($p_{t,j}^{v'}$) is the feature representation for the v-th (v'-th) view of j-th unlabeled sample in task t, $g_t^v(u_t^v, z_t^v, \Theta^v)$ is the regularization function defined as:

$$g_t^v(u_t^v, z_t^v, \Theta^v) = \alpha\|u_t^v - {\Theta^v}^\top z_t^v\|^2 + \beta\|u_t^v\|^2, \tag{7}$$

where the structure parameter Θ^v is a $h \times d_v$ matrix. The regularization function in Eq.(7) controls the task relatedness as well as the complexity of the predictor models. So, the optimization problem described in Eq.(6) can take advantage of multiple views and multiple tasks simultaneously.

4.2 A Relaxed Convex Formulation

The problem in Eq.(6) is non-convex and difficult to solve due to its orthonormal constraints and the regularization in terms of u_t^v, z_t^v and Θ^v (suppose L is convex loss function). Converting it into a convex formulation is desirable. The optimal $\{z_t^v\}$ for the problem in Eq.(6) can be expressed as $z_t^v = \Theta^v u_t^v$. Let $U^v = [u_1^v, u_2^v, \ldots, u_T^v] \in \mathbb{R}^{d_v \times T}$ and $Z^v = [z_1^v, z_2^v, \ldots, z_T^v] \in \mathbb{R}^{h \times T}$, so $Z^v = \Theta^v U^v$. Then we denote:

$$G_0(U^v, \Theta^v) = \min_{Z^v} \sum_{t=1}^{T} g_t^v(u_t^v, z_t^v, \Theta^v) = \alpha\mathrm{tr}\left({U^v}^\top((1+\eta)I - {\Theta^v}^\top\Theta^v)U^v\right) \tag{8}$$

where $\eta = \beta/\alpha > 0$. Eq.(8) can be reformulated into an equivalent form given by

$$G_1(U^v, \Theta^v) = \alpha\eta(1+\eta)\mathrm{tr}\left({U^v}^\top(\eta I + {\Theta^v}^\top\Theta^v)^{-1}U^v\right). \tag{9}$$

The orthonormality constraint on Θ^v is non-convex, which makes the optimization problem non-convex. One method is to relax the feasible domain of it into a convex set. Let $M^v = \Theta^{v\top}\Theta^v$, using a similar derivation as in [6], the feasible domain of the optimization problem can be relaxed into a convex set, and a convex formulation of the problem in Eq.(6) can be defined as follows:

$$\min_{\{u_t^v, M^v\}} \sum_{t=1}^{T}\sum_{v=1}^{V}\left(\frac{1}{n_t}\sum_{i=1}^{n_t} L\left(f_t^v(x_{t,i}^v), y_{t,i}\right) + \gamma\frac{1}{m_t}\sum_{j=1}^{m_t}\sum_{v'\neq v}\left(f_t^{v'}(p_{t,j}^{v'}) - f_t^v(p_{t,j}^v)\right)^2\right)$$

$$+\sum_{v=1}^{V} G_2(U^v, M^v), \qquad \text{subject to}: \quad \mathrm{tr}(M^v) = h, M^v \preceq I, M^v \in \mathbb{S}_+, \tag{10}$$

where $G_2(U^v, M^v)$ is defined as:

$$G_2(U^v, M^v) = \alpha\eta(1+\eta)\mathrm{tr}\left(U^{v\top}(\eta I + M^v)^{-1}U^v\right). \tag{11}$$

Note that the problem in Eq.(10) is a convex relaxation of that in Eq.(6). The optimal Θ^v to Eq.(6) can be approximated using the top h eigenvectors (corresponding to the largest h eigenvalues) of the optimal M^v computed from Eq.(10).

4.3 Convex Shared Structure Learning Algorithm

The optimization problem in Eq.(10) is convex, so the globally optimal solution can be obtained. In this section, a convex shared structure learning algorithm for MTMV problem (*CSL-MTMV*) is presented. In *CSL-MTMV* algorithm, the two optimization variables are optimized alternately, that is, one variable is fixed, while the other one can be optimized according to the fixed one. The methods are described in the following, and the final algorithm is in Algorithm 1.

Computation of $\{U^v\}$ for Given $\{M^v\}$. In Eq.(10), if $\{M^v\}$ are given, it can be easily found that the computation of u_t^v for different tasks can be decoupled, that is, different tasks' weight vectors can be computed separately. Suppose the least square loss function is used where:

$$L\left(f_t^v(x_{t,i}^v), y_{t,i}\right) = (u_t^{v\top}x_{t,i}^v - y_{t,i})^2. \tag{12}$$

We denote the objective function in Eq.(10) as F, and the derivative regarding to each u_t^v is:

$$\begin{aligned}\frac{\partial F}{\partial u_t^v} =& \frac{2}{n_t}\sum_{i=1}^{n_t}(u_t^{v\top}x_{t,i}^v - y_{t,i})x_{t,i}^v + \gamma\frac{2}{m_t}\sum_{j=1}^{m_t}\sum_{v'\neq v}\left(u_t^{v\top}p_{t,j}^v - u_t^{v'\top}p_{t,j}^{v'}\right)p_{t,j}^v \\ &+ 2\alpha\eta(1+\eta)(\eta I + M^v)^{-1}u_t^v\end{aligned} \tag{13}$$

For convenience, the following notations are given:

$$\begin{aligned} A_t^v &= \frac{2}{n_t} X_t^{v\top} X_t^v + \gamma \frac{2}{m_t}(V-1) P_t^{v\top} P_t^v + 2\alpha\eta(1+\eta)(\eta I + M^v)^{-1} \\ B_t^{vv'} &= -\gamma \frac{2}{m_t} P_t^{v\top} P_t^{v'}, \quad C_t^v = \frac{2}{n_t} X_t^{v\top} y_t \end{aligned} \tag{14}$$

where X_t^v, P_t^v and y_t are described in Section 3.1. By setting Eq.(13) to zero and rearranging the terms, the following equation can be obtained:

$$A_t^v u_t^v + \sum_{v' \neq v} B_t^{vv'} u_t^{v'} = C_t^v. \tag{15}$$

From Eq.(15), u_t^v is correlated with other $u_t^{v'}$ for the same task t, i.e., the views of the same task are correlated. u_t^v and $u_{t'}^v$ from different tasks are not correlated. Therefore, the u_t^v of different tasks can be computed separately, while the different views for the same task must be solved together. Note that such an equation can be obtained for each view v in task t. By combining these equations, the following linear equation system can be obtained for each task t:

$$\mathcal{L}_t \mathcal{W}_t = \mathcal{R}_t \tag{16}$$

where $\mathcal{L}_t \in \mathbb{R}^{D\times D}$ is a symmetric block matrix with $V \times V$ blocks. The specific forms of the symbols in Eq.(16) are as follows:

$$\begin{aligned} \mathcal{L}_t &= \begin{bmatrix} A_t^1 & B_t^{12} & \cdots & B_t^{1V} \\ B_t^{21} & A_t^2 & \cdots & B_t^{2V} \\ \vdots & \vdots & \ddots & \vdots \\ B_t^{V1} & B_t^{V2} & \cdots & A_t^V \end{bmatrix} \\ \mathcal{W}_t = \mathrm{Vec}\left([u_t^1, u_t^2, \cdots, u_t^V]\right), &\quad \mathcal{R}_t = \mathrm{Vec}\left([C_t^1, C_t^2, \cdots, C_t^V]\right) \end{aligned} \tag{17}$$

where Vec() denotes the function stacking the column vectors in a matrix to a single column vector. For each task t, an equation system described in Eq.(16) is constructed and solved. The optimal solution of $\{u_t^v\}$ can be easily obtained by left multiplication of the (pseudo-) inverse of matrix $\mathcal{L}_t$.

Computation of $\{M^v\}$ for Given $\{U^v\}$. For given $\{U^v\}$, in Eq.(10), different M^v are not correlated, they can be computed separately. For each view v, the following problem can be obtained:

$$\min_{M^v} \mathrm{tr}\left(U^{v\top}(\eta I + M^v)^{-1} U^v\right), \text{subject to} : \mathrm{tr}(M^v) = h, M^v \preceq I, M^v \in \mathbb{S}_+ \tag{18}$$

This problem is a semidefinite program (SDP), where direct optimization is computationally expensive. An efficient approach to solve it is described in the following. Let $U^v = P_1 \Sigma P_2^\top$ be its singular value decomposition (SVD), where $P_1 \in \mathbb{R}^{d_v \times d_v}$ and $P_2 \in \mathbb{R}^{T\times T}$ are column-wise orthogonal, and $\mathrm{rank}(U^v) = q$. In

general, $q \leq T \leq d_v$, we also suppose that the dimension h of the shared feature space for the T tasks satisfies $h \leq q$. Then,

$$\Sigma = \text{diag}(\sigma_1, \cdots, \sigma_T) \in R^{d_v \times T},\ \sigma_1 \geq \cdots \geq \sigma_q > 0 = \sigma_{q+1} = \cdots = \sigma_T. \quad (19)$$

Consider the following optimization problem:

$$\min_{\gamma_i} \sum_{i=1}^{q} \frac{\sigma_i^2}{\eta + \gamma_i}, \qquad \text{subject to} : \sum_{i=1}^{q} \gamma_i = h, 0 \leq \gamma_i \leq 1, \forall i, \quad (20)$$

where $\{\sigma_i\}$ are the singular values of U^v defined in Eq.(19), this optimization problem is convex [4]. The problem in Eq.(20) can be solved via many existing algorithms such as the projected gradient descent method [4].

Chen et al. [6] show that how to transform the SDP problem in Eq.(18) into the convex optimization problem in Eq.(20). Specifically, let $\{\gamma_i^*\}$ be optimal to Eq.(20) and denote $\Lambda^* = diag(\gamma_1^*, \cdots, \gamma_q^*, 0) \in \mathbb{R}^{d_v \times d_v}$. Let $P_1 \in \mathbb{R}^{d_v \times d_v}$ be orthogonal consisting of the left singular vectors of U^v. Then $M^{v*} = P_1 \Lambda^* P_1^\top$ is an optimal solution to Eq.(18). In addition, by solving the problem in Eq.(20) we obtain the same optimal solution and objective value as Eq.(18).

Algorithm 1. Convex shared structure learning algorithm for MTMV problem (*CSL-MTMV*)

Input:
$\{y_t\}_{t=1}^T$, $\{X_t\}_{t=1}^T$, $\{P_t\}_{t=1}^T$, α, β, γ, h
Output:
$\{U^v\}_{v=1}^V$, $\{Z^v\}_{v=1}^V$, $\{\Theta^v\}_{v=1}^V$
Method:
Initialize $\{M^v\}_{v=1}^V$ that satisfy the constraints in Eq.(18);
repeat
  for $t = 1$ to T **do**
    Construct $A_t^v, B_t^{vv'}, C_t^v$ defined in Eq.(14);
    Construct $\mathcal{L}_t, \mathcal{R}_t$ defined in Eq.(17);
    Compute $\mathcal{W}_t = \mathcal{L}_t^{-1} \mathcal{R}_t$;
  end for
  for $v = 1$ to V **do**
    Compute the SVD of $U^v = P_1 \Sigma P_2^\top$;
    Compute the optimal values of $\{\gamma_i^*\}$ for problem in Eq.(20);
    Denote $\Lambda^* = diag(\gamma_1^*, \cdots, \gamma_q^*, 0)$, and compute $M^v = P_1 \Lambda^* P_1^\top$;
  end for
until convergence criterion is satisfied.
For each v, construct Θ^v using the top h eigenvectors of M^v;
Compute $Z^v = \Theta^v U^v$;
return $\{U^v\}_{v=1}^V$, $\{Z^v\}_{v=1}^V$, $\{\Theta^v\}_{v=1}^V$.

4.4 Dealing with Missing-View Data

In the previous sections, we only consider the ideal case that all tasks in a data set have complete data. When incomplete data is involved in the MTMV learning, the problem becomes more challenging. We aim to handle the case of "structured" missing views as described in [25]. That is, if a view is missing from a task, it is missing in all the samples in the task. Of course, partially observed views (i.e. some views are missing only in a part of samples in a task) are more difficult to deal with, which is beyond the scope of this paper.

In our MTMV learning framework, it is easy dealing with structured missing views. Let $V_t \leq V$ denote the real number of views contained in task t and $T_v \leq T$ denote the number of tasks contain view v. When computing $\{U^v\}$ for given $\{M^v\}$, if view v is missing from task t, the variables related to view v in Eq.(16) are all useless, including u_t^v in $\mathcal{W}_t$, C_t^v in $\mathcal{R}_t$, $B_t^{vv'}$ in $\mathcal{L}_t$, and the v-th block row and block column in matrix $\mathcal{L}_t$. After removing these variables, and replace V, T using V_t, T_v in the corresponding equations, a problem with smaller size can be obtained:

$$\mathcal{L}'_t \mathcal{W}'_t = \mathcal{R}'_t \tag{21}$$

When computing $\{M^v\}$ for given $\{U^v\}$, if view v is missing from task t, then in Eq.(18), the t-th column of matrix $\{U^v\}$ (i.e. u_t^v) does not exist. After removing this column and replace V, T using V_t, T_v in the corresponding equations, a similar optimization problem can be obtained.

Furthermore, if for a view v, $T_v = 1$, i.e., there is only one task that contains view v, the algorithm can still be improved. As stated above, the shared structure among multiple tasks is learned based on the relationships of these tasks, if only one task exists for a view, then there is no need to learn the shared low dimensional feature space for this view. Specifically, if only task t contains view v, then the prediction model for this view is as follows:

$$\bar{f}_t^v(x_t^v) = {u_t^v}^\top x_t^v \tag{22}$$

In the optimization problem in Eq.(6), for this view, the regularization function $g_t^v(u_t^v, z_t^v, \Theta^v)$ is replaced with $\bar{g}_t^v(u_t^v) = \beta \|u_t^v\|^2$. After some direct derivation, it can be found that the A_t^v in Eq.(17) should have the new form as:

$$\bar{A}_t^v = \frac{2}{n_t} {X_t^v}^\top X_t^v + 2\beta I \tag{23}$$

For this view, there is no need to compute M^v or Θ^v in every iteration.

4.5 Complexity Analysis of the Algorithm

To analyze the complexity of *CSL-MTMV* algorithm, we consider the worst case that all the tasks in the problem have features from all the views. In the algorithm, we need to construct $\mathcal{L}_t, \mathcal{W}_t, \mathcal{R}_t$ defined in Eq.(17), compute the inverse of matrix $\mathcal{L}_t$, and compute $\{M^v\}$. It can be found that the speed bottleneck is computation of T inverse of matrices $\mathcal{L}_t$, where the time complexity is $\mathbf{O}(TD^3)$.

The space requirement of the algorithm mainly depends on the size of matrix $\mathcal{L}_t$ with $\mathbf{O}(D^2)$. The time complexity of *regMVMT* algorithm [25] is $\mathbf{O}(T^3D^3)$ and space complexity is $\mathbf{O}(TD^2 + T(T-1)D)$. It can be easily found that through decoupling different tasks in the computation process, *CSL-MTMV* can significantly reduce the time and space complexity.

4.6 Relationship with *regMVMT* Algorithm

The *regMVMT* algorithm [25] can be seen as a special case of our algorithm. Specifically, in Eq.(6), we set $\Theta^v = I$, $z_t^v = \frac{1}{T_v}\sum_{i=1}^{T_v} u_t^v$, and do not use the weighting factors $\frac{1}{m_t}$ and $\frac{1}{n_t}$ to compensate for the tasks with different sample numbers. With this setting, our model is transformed into the *regMVMT* problem definition in Eq.(5) in [25]. Therefore, the problem formulation in this paper is more generalized and flexible, which is able to find good solutions with more chance. In fact, *regMVMT* requires that the model parameters of all the tasks are similar, which is too rigorous for problems with outlier tasks. In this paper, the common structures between different tasks are learned and different tasks share information using these structures. Compared with other state-of-the-art methods, such as data-manifold methods based on graph structure, our method can learn some underlying predictive functional structures in hypothesis space, which better characterizes a set of good predictors.

5 Experiments

In this section, we conduct the experiments on four real-world data sets to validate the effectiveness of the proposed algorithm *CSL-MTMV*.

5.1 Data Sets

All the four data sets have multiple tasks with multiple views, and some statistics of them are summarized in Table 1, where N_p and N_n denote the number of positive and negative samples in each task, respectively. The first two data sets are with complete views, and the rest two are with missing views.

- The first one is the NUS-WIDE Object web image database [9] where each image is annotated by objects such as "boat","bird", and etc. We take block-wise color moments as one view and the rest features as the other one. In this data set, we remove the images associated with zero or only one object, and those tasks with too few positive or negative examples. Finally, a two-view data set with 11 tasks are obtained.
- The second one is the Leaves data set [13]. It includes leaves from one hundred plant species that are divided into 32 different genuses, and 16 samples of leaves for each plant species are presented. For each sample, a shape descriptor, fine scale margin and texture histogram are given. By selecting one species from each of the 32 different genuses, 32 tasks with three views are obtained.

- The third one is constructed from 20 Newsgroups[1], which includes 20 categories. 200 documents are randomly selected from each category. For each task, the documents from one category are regarded as positive samples, and from another different category are negative ones. We take the words appearing in all 20 tasks as the common view, and the words existing only in each task as specified view. Finally, we construct 20 tasks with totally 21 views, while each task with 19 views missing. The *tf-idf* weighting scheme is adopted, and the principal component analysis [22] is used to reduce the dimension of features to 300 for each view.
- The last one is NIST Topic Detection and Tracking (TDT2) corpus [10]. In this data set, only the largest 20 categories are selected, and for the categories containing more than 200 documents, we randomly selected 200 documents from each category. The tasks and views are similarly constructed as 20 Newsgroups. We also have 20 tasks with totally 21 views, and each task with 19 views missing.

Table 1. Description of the data sets

Data set	T	V	N_p	N_n	View Missing?
NUS-WIDE Object	11	2	$310 \sim 1220$	$2438 \sim 3348$	No
leaves	32	3	16	496	No
20 Newsgroups	20	21	200	200	Yes
TDT2	20	21	$98 \sim 200$	200	Yes

5.2 Baselines

We compare *CSL-MTMV* with the following baselines, which can handle multi-task problems with multiple views:

• ***IteM*^{2}:** *IteM*^{2} algorithm [14] is a transductive algorithm, and it can only handle nonnegative feature values. When applying *IteM*^{2} algorithm to some of our data sets that have negative feature values, we add a positive constant to the feature values to guarantee its nonnegativity.

• ***regMVMT*:** *regMVMT* algorithm [25] is an inductive algorithm, which assumes all tasks should be similar to achieve good performance.

5.3 Experiment Setting and Evaluation Metric

Experiment Setting. In each data set, we randomly select n labeled samples and m unlabeled samples for each task as training set. The value of n is set according to the complexity of the learning problem, and m is generally $2 \sim 4$ times of n. We apply five-fold cross validation on the training set to optimize

[1] `http://people.csail.mit.edu/jrennie/20Newsgroups/`

the parameters for the algorithms *CSL-MTMV* (including α, β and γ.) and *reg-MVMT* (including λ, μ and γ). The parameters of *IteM*^{2} are set the same as their original paper. For *CSL-MTMV*, the number of iteration is set to 20, and number of dimensionality h as $\lfloor (T-1)/5 \rfloor \times 5$ in our experiments.

Evaluation Metric. The F_1 measure is adopted to evaluate all the algorithms, since the *accuracy* measure may be vulnerable to the class unbalance, which just exists in some of our data sets. Let tp, fp and fn denote the numbers of true positive samples, false positive samples and false negative samples, respectively, then $Precision = tp/(tp + fp)$, $Recall = tp/(tp + fn)$.

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{24}$$

Each experiment is repeated 10 times, and each time we randomly select n labeled samples and m unlabeled samples for each task as training set. Finally, the average value of F_1 is recorded.

5.4 Experiment Results

Learning with Complete-View Data. The first two data sets in Table 1 are with complete views.

For NUS-WIDE Object data set, different number of labeled samples are chosen as training set to test the performance of these algorithms, i.e., the number of labeled samples are selected in the range $[100, 700]$ with interval of 100. All the results are shown in Table 2, which can be observed that, the value of F_1 increases with the increasing of the number of labeled samples, and *CSL-MTMV* achieves the best results under various cases.

For the leaves data set, there are only 16 positive samples for each task, so the number of labeled samples is fixed as 50, among which the number of positive samples is set to {1,2,3,4,5,6,7} separately. The experiment results are shown in Table 3, where the first line gives the numbers of positive samples. Similar results

Table 2. Experimental results on NUS-WIDE Object data set

samples #	100	200	300	400	500	600	700
IteM^2	0.1539	0.1529	0.1526	0.1534	0.1546	0.1522	0.1512
regMVMT	0.3695	0.3822	0.3875	0.3918	0.4036	0.4102	0.4159
CSL-MTMV	**0.3930**	**0.4075**	**0.4104**	**0.4178**	**0.4193**	**0.4211**	**0.4263**

Table 3. Experimental results on leaves data set

positive samples #	1	2	3	4	5	6	7
IteM^2	0.0289	0.0341	0.0397	0.0390	0.0373	0.0371	0.0392
regMVMT	0.0598	0.0981	0.1611	0.2637	0.3573	0.4623	0.5644
CSL-MTMV	**0.0802**	**0.1072**	**0.1905**	**0.3017**	**0.4045**	**0.5229**	**0.6128**

can be obtained, i.e., *CSL-MTMV* performs the best under different numbers of labeled positive samples.

Learning with Missing-View Data. In real-world problems, some tasks may not share all the views, so the problems with missing views are also considered. In the experiments, the last two data sets, 20 Newsgroups and TDT2, are with missing views.

Different number of labeled samples are also selected as training set to test the performance of these compared algorithms. The number is sampled in the range $[10, 70]$ with interval of 10, and the results are recorded in Tables 4 and 5. We can observe the similar results as the first two data sets. Again, *CSL-MTMV* gives the best performance.

Table 4. Experimental results on 20 Newsgroups data set

samples #	10	20	30	40	50	60	70
IteM^2	0.4880	0.4879	0.4912	0.4776	0.4866	0.5068	0.5247
regMVMT	0.8570	0.9144	0.9330	0.9500	0.9566	0.9629	0.9651
CSL-MTMV	**0.8733**	**0.9256**	**0.9406**	**0.9540**	**0.9597**	**0.9652**	**0.9667**

Table 5. Experimental results on TDT2 data set

samples #	10	20	30	40	50	60	70
IteM^2	0.4922	0.4897	0.5142	0.5101	0.5159	0.5069	0.5160
regMVMT	0.9742	0.9903	0.9930	0.9941	0.9949	0.9947	0.9947
CSL-MTMV	**0.9825**	**0.9936**	**0.9946**	**0.9956**	**0.9957**	**0.9962**	**0.9958**

It is worth mentioning that, we find *IteM*^{2} can not perform well on these four data sets. We conjecture there may be two reasons, 1) *IteM*^{2} can only handle the data sets with non-negative values of features. 2) *IteM*^{2} assumes the test set should have the same proportion of positive samples as the training set, which might also degrade classification performance.

6 Conclusions

To deal with the MTMV problems, a shared structure learning framework called *CSL-MTMV* is proposed in this paper, in which both the shared predictive structure among multiple tasks and prediction consistence among different views within a single task are considered. We also convert the optimization problem to a convex one, and develop an alternating optimization algorithm to solve it. The algorithm can decouple different tasks in the computation process, which significantly reduces the time complexity and space complexity. Moreover, *CSL-MTMV* is a general framework, since the recently proposed algorithm *regMVMT* can be regarded as a special case of ours. The experiments on four real-world data sets demonstrate the effectiveness of the proposed framework.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (No. 61175052, 61203297, 60933004, 61035003), National High-tech R&D Program of China (863 Program) (No. 2013AA01A606, 2012AA011003), and National Program on Key Basic Research Project (973 Program) (No. 2013CB329502).

References

1. Ando, R.K., Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. Journal of Machine Learning Research 6, 01 (2005)
2. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. In: Advances in Neural Information Processing Systems, Vancouver, BC, Canada, pp. 41–48 (2007)
3. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, pp. 92–100. ACM, New York (1998)
4. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press (2004)
5. Caruana, R.: Multitask learning. Machine Learning 28(1), 41–75 (1997)
6. Chen, J., Tang, L., Liu, J., Ye, J.: A convex formulation for learning shared structures from multiple tasks. In: Proceedings of the 26th International Conference on Machine Learning, ICML 2009, Montreal, QC, Canada, pp. 137–144 (2009)
7. Chen, J., Zhou, J., Ye, J.: Integrating low-rank and group-sparse structures for robust multi-task learning. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, United States, pp. 42–50 (2011)
8. Chen, N., Zhu, J., Xing, E.P.: Predictive subspace learning for multi-view data: A large margin approach. In: Annual Conference on Neural Information Processing Systems 2010, NIPS 2010, Vancouver, BC, Canada (2010)
9. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: A real-world web image database from national university of singapore. In: CIVR 2009 - Proceedings of the ACM International Conference on Image and Video Retrieval, Santorini Island, Greece, pp. 368–375 (2009)
10. Cieri, C., Strassel, S., Graff, D., Martey, N., Rennert, K., Liberman, M.: Corpora for topic detection and tracking. In: Allan, J. (ed.) Topic Detection and Tracking, pp. 33–66. Kluwer Academic Publishers, Norwell (2002)
11. Evgeniou, T., Pontil, M.: Regularized multi-task learning. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, United States, pp. 109–117 (2004)
12. Farquhar, J.D., Hardoon, D.R., Meng, H., Shawe-Taylor, J., Szedmak, S.: Two view learning: Svm-2k, theory and practice. In: Advances in Neural Information Processing Systems, Vancouver, BC, Canada, pp. 355–362 (2005)
13. Frank, A., Asuncion, A.: UCI machine learning repository (2013), `http://archive.ics.uci.edu/ml`
14. He, J., Lawrence, R.: A graph-based framework for multi-task multi-view learning. In: Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, WA, United States, pp. 25–32 (2011)
15. Jalali, A., Ravikumar, P., Sanghavi, S., Ruan, C.: A dirty model for multi-task learning. In: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010, Vancouver, BC, Canada (2010)

16. Muslea, I., Minton, S., Knoblock, C.A.: Active + Semi-supervised Learning = Robust Multi-View Learning. In: International Conference on Machine Learning, pp. 435–442 (2002)
17. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: International Conference on Information and Knowledge Management, pp. 86–93 (2000)
18. Sindhwani, V., Niyogi, P., Belkin, M.: A Co-Regularization Approach to Semi-supervised Learning with Multiple Views. In: Workshop on Learning with Multiple Views, International Conference on Machine Learning (2005)
19. Sindhwani, V., Rosenberg, D.S.: An rkhs for multi-view learning and manifold co-regularization. In: Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, pp. 976–983 (2008)
20. Skolidis, G., Sanguinetti, G.: Bayesian multitask classification with gaussian process priors. IEEE Transactions on Neural Networks 22(12), 2011–2021 (2011)
21. Vapnik, V.: The nature of statistical learning theory. Springer (1999)
22. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemometrics and Intelligent Laboratory Systems 2(1), 37–52 (1987)
23. Yu, S., Krishnapuram, B., Rosales, R., Bharat Rao, R.: Bayesian co-training. Journal of Machine Learning Research 12, 2649–2680 (2011)
24. Yu, S., Tresp, V., Yu, K.: Robust multi-task learning with t-processes. In: Twenty-Fourth International Conference on Machine Learning, Corvalis, OR, United States, vol. 227, pp. 1103–1110 (2007)
25. Zhang, J., Huan, J.: Inductive multi-task learning with multiple view data. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, pp. 543–551 (2012)