A Graph-based Framework for Multi-Task Multi-View Learning

Jingrui He JINGRUHE@US.IBM.COM
Rick Lawrence RICKLAWR@US.IBM.COM
IBM T.J. Watson Research Center

Abstract

Many real-world problems exhibit *dual-heterogeneity*. A single learning task might have features in multiple views (i.e., *feature heterogeneity*); multiple learning tasks might be related with each other through one or more shared views (i.e., *task heterogeneity*). Existing multi-task learning or multi-view learning algorithms only capture one type of heterogeneity.

In this paper, we introduce Multi-Task Multi-View (M^2TV) learning for such complicated learning problems with both feature heterogeneity and task heterogeneity. We propose a graph-based framework ($GraM^2$) to take full advantage of the dual-heterogeneous nature. Our framework has a natural connection to Reproducing Kernel Hilbert Space (RKHS). Furthermore, we propose an iterative algorithm ($IteM^2$) for $GraM^2$ framework, and analyze its *optimality, convergence and time complexity*. Experimental results on various real data sets demonstrate its effectiveness.

1. Introduction

Many real-world problems exhibit *dual-heterogeneity*. 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*). For example, sentiment classification for movie reviews and for political blog posts are two related tasks. They both have the word features. However, political blog posts may have additional features based on the social network of the blog users. Another example is multi-lingual web image annotation, where images collected from Chinese web sites and English web sites both have content-based features, and they also have task-specific features, i.e., surrounding texts in Chinese and English respectively. (See Figure 1 for an illustrative example.) Neither multi-task learning nor multi-view learning is optimal for such a complicated learning problem. Existing multi-task learning explores the relatedness with other tasks, but disregards the consistency among different views of a single task; whereas existing multi-view learning ignores the label information from other related tasks.

To the best of our knowledge, there does not exist an effective learning method to fully explore both the feature heterogeneity and the task heterogeneity simultaneously. This is partially due to the fact that existing multi-task learning and multi-view learning algorithms adopt quite different methodologies (see Section 2 for a brief review). It is not clear how to seamlessly bridge them together to enjoy the best of both worlds.

To address such challenges, in this paper, we introduce a novel Multi-Task Multi-View (M^2TV) learning problem. On one hand, it uses the label information from related tasks to make up for the lack of labeled data in a single task; on the other hand, it uses the consistency among different views to improve the performance. It is tailored for the above complicated dual-heterogeneous problems where multiple related tasks have both shared and task-specific views (features), since it makes full use of the available information.

For M^2TV learning, we propose a graph-based framework ($GraM^2$). Within each task, we construct a bipartite graph for each view, modeling the relationship between the examples and the features in this view. The consistency among different views is obtained by requiring them to produce the same classification function, which is commonly used in multi-view learning. Across different tasks, we establish their relationship by imposing the similarity constraint on the common views. Furthermore, an iterative algorithm ($IteM^2$) is proposed to solve the $GraM^2$ framework. We conduct theoretical analysis as well as empirical evaluations to demonstrate the effectiveness of our method.

The main contributions of this paper can be summarized as follows.

1. *Problem Definition*: we introduce a novel problem named Multi-Task Multi-View learning (M^2TV), where multiple related tasks have both shared and task-specific views.

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

2. *Framework*: we propose a graph-based framework ($GraM^2$) for M^2TV learning. We show that our framework ($GraM^2$) has a natural connection to Reproducing Kernel Hilbert Space (RKHS).
3. *Algorithm*: we propose an effective algorithm ($IteM^2$) for the $GraM^2$ framework. We show that $IteM^2$ converges to an optimal solution in a scalable way.

The rest of this paper is organized as follows. We first review related work in Section 2. In Section 3, we introduce the problem definition. Then we propose the graph-based framework ($GraM^2$) in Section 4, followed by an analysis of the RKHS. The $IteM^2$ algorithm is presented and analyzed in Section 5. To demonstrate the effectiveness of $IteM^2$, we show some experimental results in Section 6. Finally, we conclude in Section 7.

2. Related Work

As mentioned before, many real-world problems exhibit dual-heterogeneity. Most existing works only explore one type of heterogeneity, such as multi-task learning and multi-view learning.

Multi-View Learning. The basic idea of multi-view learning is to make use of the consistency among different views to achieve better performance. One of the earliest work in multi-view learning is (Blum & Mitchell, 1998), where the authors propose the co-training algorithm for problems where the examples are described by two distinct views. The authors in (Nigam & Ghani, 2000) further analyze the performance of co-training when certain assumptions are violated. More recent work in multi-view learning include: SVM-2K algorithm proposed in (Farquhar et al., 2005), which combines KCCA with SVM in an optimization framework; CoMR algorithm proposed in (Sindhwani & Rosenberg, 2008), which is based on an RKHS with a data-dependent 'co-regularization' norm; the large-margin framework for multi-view data (Chen et al., 2010), which is based on an undirected latent space Markov network, to name a few. In M^2TV learning, we also perform multi-view learning within a single task. In addition, we are able to use the label information from other related tasks, which is particularly useful when the number of labeled examples in a single task is very small.

Multi-Task Learning. In multi-task learning, people model task relatedness in various ways. Some researchers assume that the function parameters for different tasks are similar, such as the kernel methods for learning multiple tasks (Evgeniou et al., 2005), the semi-supervised multi-task learning framework (Liu et al., 2007), the clustered multi-task learning algorithm (Jacob et al., 2008), etc. Some researchers assume that different tasks share a common representation / structure, such as the multi-task feature learning algorithm based on 1-norm (Argyriou et al., 2008), the ASO algorithm (Ando & Zhang, 2005) and its improved version, the CASO algorithm (Chen et al., 2009), the mt-lmnn algorithm (Parameswaran & Weinberger, 2010), etc. In M^2TV learning, we also perform multi-task learning via the common views shared by different tasks. In addition, we are able to leverage the consistency among different views of a single task to achieve better performance.

Other Related Work. There are some existing works which try to explore multiple types of heterogeneity. Although successful in themselves, their problem settings are fundamentally different from ours. For example, in (Cavallanti et al., 2010), the authors study linear algorithms for online multi-task multi-view learning. However, their settings are named multi-view because examples from different tasks come from different feature spaces, and the features of a single task do *not* form different views; whereas in our settings, the features of a single task form different views, some of which are shared across different tasks, some of which are not. Therefore, if applied in our settings, their algorithm can not make use of the consistency among different views of the same task. In (Zhao & Hoi, 2010), the authors study online transfer learning both in a homogeneous domain and across heterogeneous domains. However, in this paper, we are interested in multiple tasks instead of a single target task (domain). Furthermore, they assume that the feature space of the source domain is a subset of that of the target domain whereas we can address more general settings.

3. M^2TV: Problem Definition

In this section, we formally introduce our M^2TV learning. Suppose that we have T tasks and V views in total. Each task has V_i views, $1 \leq V_i \leq V$, $i = 1, \ldots, T$. Each view corresponds to a type of feature, e.g., bag of words, linkage among the examples, etc. For the i^{th} task and the k^{th} view, there are d_{ik} features. Let S_{ij} denote the set of indices of common views shared by the i^{th} and j^{th} tasks. $S_{ii} = \phi$. For example, $S_{12} = \{1\}$ means that Task 1 and Task 2 share the first view. If $1 \in S_{12}$, and $1 \in S_{13}$, then $1 \in S_{23}$.

Notice that existing multi-task learning and multi-view learning are special cases of our M^2TV learning. To be specific, if $V_i = 1$, and $S_{ij} = \{1\}$, $i, j = 1, \ldots, T$, $i \neq j$, the problem is reduced to multi-task learning; if $T = 1$, the problem is reduced to multi-view learning.

For the i^{th} task, we have n_i examples, which are de-

noted $\mathbb{X}_i = \{x_{i1}, \ldots, x_{in_i}\} \subset \mathbb{R}^{\sum_{k=1}^{V_i} d_{ik}}$. In this paper, we assume that the features are non-negative, e.g., word frequency in document classification[1]. Without loss of generality, suppose that the first m_i examples are labeled $y_{i1}, \ldots, y_{im_i}$, which are either 1 or -1. Note that m_i is usually very small compared with n_i. So our goal is to leverage the label information from all the tasks to help classify the unlabeled examples in each task, as well as to use the consistency among different views of a single task to improve the performance.

4. $GraM^2$: A Graph-based Framework

In this section, we present our graph-based framework ($GraM^2$) for M^2TV learning. We first present its objective function. Then we show how it can be reduced to standard supervised learning via an RKHS.

4.1. Objective Function

In our $GraM^2$ framework, we have two types of functions. One is defined on the examples. To be specific, for the i^{th} task, define function $g_i(\cdot)$, which takes values on $x_{i1}, \ldots, x_{in_i}$. $g_i(\cdot) > 0$ indicates a positive class label whereas $g_i(\cdot) < 0$ indicates a negative class label. The other one is defined on the features. To be specific, for the i^{th} task and the k^{th} view, define function $f_{ik}(\cdot)$, which takes values on the features in this view. $f_{ik}(\cdot)$ helps determine the class label of an example having such features. Take sentiment classification as an example. $f_{ik}(\cdot) > 0$ indicates positive polarity of a word whereas $f_{ik}(\cdot) < 0$ indicates negative polarity. The polarity of all the words in a document together will determine the sentiment of the document. Furthermore, if $|f_{ik}(\cdot)|$ is large, then the corresponding word often has strong polarity; on the other hand, if $|f_{ik}(\cdot)|$ is small, then the corresponding word has weak polarity, which may even have conflicting polarity in different context.

For the i^{th} task and the k^{th} view, we construct a bipartite graph $G_{ik} = \{N_{ik}, E_{ik}\}$ where N_{ik} is the set of nodes and E_{ik} is the set of undirected edges. N_{ik} consists of two types of nodes, i.e., the nodes that correspond to the examples in this task, and the nodes that correspond to the features in this view. There is an edge between an example node and a feature node if and only if the feature value for the example is positive, and the weight of the edge is just the feature value. Figure 1 shows an example of such bipartite graphs. Let W_{ik}, $(n_i + d_{ik}) \times (n_i + d_{ik})$, denote the affinity matrix for G_{ik}. It has the following structure.

$$W_{ik} = \begin{bmatrix} 0_{n_i \times n_i} & A_{ik} \\ A_{ik}^T & 0_{d_{ik} \times d_{ik}} \end{bmatrix}$$

[1]Exploring negative features is beyond the scope of the current paper and is our future work.

where A_{ik} is an $n_i \times d_{ik}$ matrix. If the t^{th} feature of the s^{th} example is positive, then $A_{ik}(s,t)$ (the element of A_{ik} in the s^{th} row and t^{th} column) is set to be this feature value. Furthermore, we normalize W_{ik} to obtain:

$$T_{ik} = D_{ik}^{-1/2} W_{ik} D_{ik}^{-1/2} \quad (1)$$

where D_{ik} is a diagonal matrix whose s^{th} element $D_{ik}(s)$ is equal to the sum of the s^{th} row of W_{ik}.

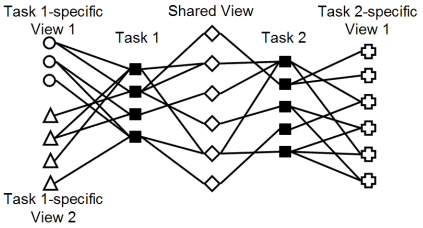


Figure 1. Illustration of M^2TV learning. Examples from Task 1 and Task 2 (black squares) have both a shared view (diamonds) and the task specific views (circles and triangles for the 2 views of Task 1, and pluses for the 1 view of task 2). The weight of an edge between an example node and a feature node is set to the feature value.

On bipartite graph G_{ik}, we hope to observe label consistency among the nodes. To be specific, a positive example (i.e., $g_i(\cdot) > 0$) should be connected with positive features (i.e., $f_{ik}(\cdot) > 0$) and vice versa. In a more principled way, we measure the consistency by

$$\begin{aligned} C_{ik} &= \sum_{s=1}^{n_i} \sum_{t=1}^{d_{ik}} A_{ik}(s,t) \left(\frac{g_i(s)}{\sqrt{D_{ik}(s)}} - \frac{f_{ik}(t)}{\sqrt{D_{ik}(n_i + t)}} \right)^2 \\ &= \|g_i\|^2 + \|f_{ik}\|^2 - 2 g_i^T L_{ik} f_{ik} \end{aligned}$$

where L_{ik} is an $n_i \times d_{ik}$ matrix, and its element in the s^{th} row and t^{th} column $L_{ik}(s,t) = T_{ik}(s, n_i + t)$.

In this way, for Task i, we have V_i such bipartite graphs, which correspond to $C_{i1}, \ldots, C_{iV_i}$. Therefore, the overall consistency of Task i is measured by

$$C_i = \sum_{k=1}^{V_i} a_{ik} C_{ik} + \mu_i \|g_i - y_i\|^2$$

where a_{ik}, μ_i are positive parameters, and y_i is an n_i-dimensional vector. The first m_i elements of y_i are set to be the class labels of the corresponding examples, and the remaining elements are set to be 0. In C_i, the first term implicitly measures the *consistency among different views* since the function $g_i(\cdot)$ is shared by all the bipartite graphs, and the second term measures the consistency with the label information.

On the other hand, if Task i and Task j are directly related, i.e., $S_{ij} \neq \phi$, we hope to observe *similarity on the common views* of the two tasks. To be specific, $\forall k \in S_{ij}$, $\|f_{ik} - f_{jk}\|^2$ should be small. In this way, given a certain task, the information of other related tasks can be leveraged to improve its performance.

Combining the overall consistency of each task and the similarity on the common views of different tasks, we have the following objective function for $GraM^2$.

$$Q(f,g) = \sum_{i=1}^{T} C_i + b\sum_{i=1}^{T}\sum_{j=1}^{T}\sum_{k\in S_{ij}} \|f_{ik} - f_{jk}\|^2 \quad (2)$$

where b is a non-negative parameter. When $b = 0$, different tasks are decoupled.

4.2. An RKHS for $GraM^2$

In this subsection, we construct an RKHS for $GraM^2$, whose inner product depends on the common views of related tasks. For the ease of explanation, here we assume that $T = 2$, $V = 3$, $V_1 = V_2 = 2$, and $S_{12} = \{1\}$, although the analysis can be carried out to more complex cases. Related to the two tasks, we have two hypothesis spaces, $\mathcal{H}_1$ and $\mathcal{H}_2$. $\mathcal{H}_1$ ($\mathcal{H}_2$) is defined on $N_1 = \cup_{k=1}^{2} N_{1k}$ ($N_2 = \cup_{k=1}^{2} N_{2k}$), i.e., the examples in Task 1 (Task 2) as well as the features in the two views of Task 1 (Task 2). To be specific, $\forall h_1 \in \mathcal{H}_1$, its function value on an example in Task 1 is equal to $g_1(\cdot)$; its function value on a feature in the first view of Task 1 is equal to $f_{11}(\cdot)$; and its function value on a feature in the second view of Task 1 is equal to $f_{12}(\cdot)$. $\forall h_2 \in \mathcal{H}_2$, its function value is defined in a similar way. Furthermore, we impose the following norms on $\mathcal{H}_1$: $\|h_1\|_{\mathcal{H}_1}^2 = \sum_{k=1}^{2} a_{1k}C_{1k} = h_1^T M_1 h_1$, and on $\mathcal{H}_2$: $\|h_2\|_{\mathcal{H}_2}^2 = \sum_{k=1}^{2} a_{2k}C_{2k} = h_2^T M_2 h_2$, where h_1 and h_2 can be seen as both a function and a column vector whose elements are equal to their function values on N_1 and N_2 respectively, and

$$M_1 = \begin{bmatrix} (a_{11}+a_{12})I_{n_1\times n_1} & -a_{11}L_{11} & -a_{12}L_{12} \\ -a_{11}L_{11}^T & a_{11}I_{d_{11}\times d_{11}} & 0_{d_{11}\times d_{12}} \\ -a_{12}L_{12}^T & 0_{d_{12}\times d_{11}} & a_{12}I_{d_{12}\times d_{12}} \end{bmatrix}$$

$$M_2 = \begin{bmatrix} (a_{21}+a_{22})I_{n_2\times n_2} & -a_{21}L_{21} & -a_{22}L_{22} \\ -a_{21}L_{21}^T & a_{21}I_{d_{21}\times d_{21}} & 0_{d_{21}\times d_{22}} \\ -a_{22}L_{22}^T & 0_{d_{22}\times d_{21}} & a_{22}I_{d_{22}\times d_{22}} \end{bmatrix}$$

M_1 and M_2 have the following property.

Proposition 4.1. *M_1 and M_2 are positive semi-definite.*

Proof. Omitted for brevity. □

So $\mathcal{H}_1$ with norm $\|\cdot\|_{\mathcal{H}_1}$ and $\mathcal{H}_2$ with norm $\|\cdot\|_{\mathcal{H}_2}$ are RKHSs whose reproducing kernels $k_{\mathcal{H}_1} : N_1 \times N_1 \to \mathcal{R}$ and $k_{\mathcal{H}_2} : N_2 \times N_2 \to \mathcal{R}$ are given by elements of $M_1^\dagger$ and $M_2^\dagger$, the Moore-Penrose pseudo-inverse of M_1 and M_2 respectively.

Next, consider the following space

$$\mathcal{H} = \{[h_1^T, h_2^T]^T : h_1 \in \mathcal{H}_1, h_2 \in \mathcal{H}_2\}$$

with the inner product defined by

$$\langle [h_1^T, h_2^T]^T, [(h_1')^T, (h_2')^T]^T \rangle_{\mathcal{H}} =$$
$$\langle h_1, h_1' \rangle_{\mathcal{H}_1} + \langle h_2, h_2' \rangle_{\mathcal{H}_2} + b(f_{11} - f_{21})^T (f_{11}' - f_{21}')$$

Notice that this inner product depends on the common view of the two tasks (f_{i1} and f_{i1}').

With respect to $\mathcal{H}$, we have the following theorem showing that it is actually an RKHS.

Theorem 4.2. *Let $N_0 = N_1 \cup N_2$. $\mathcal{H}$ is an RKHS with the following reproducing kernel $k_{\mathcal{H}} : N_0 \times N_0 \to \mathcal{R}$ defined as follows.*
If $z \in N_1$ and $z' \in N_1$,

$$k_{\mathcal{H}}(z,z') = k_{\mathcal{H}_1}(z,z') - b\vec{k}_{\mathcal{H}_1}(z,F)(I+bS)^{-1}\vec{k}_{\mathcal{H}_1}(F,z') \quad (3)$$

If $z \in N_1$ and $z' \in N_2$,

$$k_{\mathcal{H}}(z,z') = b\vec{k}_{\mathcal{H}_1}(z,F)(I+bS)^{-1}\vec{k}_{\mathcal{H}_2}(F,z') \quad (4)$$

If $z \in N_2$ and $z' \in N_1$,

$$k_{\mathcal{H}}(z,z') = b\vec{k}_{\mathcal{H}_2}(z,F)(I+bS)^{-1}\vec{k}_{\mathcal{H}_1}(F,z') \quad (5)$$

If $z \in N_2$ and $z' \in N_2$,

$$k_{\mathcal{H}}(z,z') = k_{\mathcal{H}_2}(z,z') - b\vec{k}_{\mathcal{H}_2}(z,F)(I+bS)^{-1}\vec{k}_{\mathcal{H}_2}(F,z') \quad (6)$$

where F denotes the set of features in the common view of Task 1 and Task 2. $\vec{k}_{\mathcal{H}_1}(F,z)$ and $\vec{k}_{\mathcal{H}_2}(F,z)$ are column vectors whose elements are set to be $k_{\mathcal{H}_1}(z_0,z)$ and $k_{\mathcal{H}_2}(z_0,z)$, $z_0 \in F$, respectively. $\vec{k}_{\mathcal{H}_1}(z,F) = \vec{k}_{\mathcal{H}_1}^T(F,z)$. $\vec{k}_{\mathcal{H}_2}(z,F) = \vec{k}_{\mathcal{H}_2}^T(F,z)$. $S = K_{\mathcal{H}_1}(F,F) + K_{\mathcal{H}_2}(F,F)$, where $K_{\mathcal{H}_1}(F,F)$ and $K_{\mathcal{H}_2}(F,F)$ are Gram matrices of $k_{\mathcal{H}_1}$ and $k_{\mathcal{H}_2}$ over the set of common features respectively.

Proof. See Appendix A. □

Based on this theorem, $GraM^2$ framework can be reduced to standard supervised learning as follows.

$$h^* = [h_1^*, h_2^*]^T = \arg\min_{h\in\mathcal{H}} \|h\|_{\mathcal{H}}^2 + \sum_{i=1}^{2} \mu_i \sum_{l=1}^{m_l} (h_i(x_{il}) - y_{il})^2$$

The optimal functions g_i^* and f_{ik}^* can be obtained from h_i^* respectively.

5. $IteM^2$: The Proposed Algorithm

In Section 4, we introduced the objective function for $GraM^2$, i.e., Equation (2). In this section, we propose an effective algorithm ($IteM^2$) for solving this optimization problem, followed by an analysis of its optimality, convergence and time complexity.

5.1. The Proposed $IteM^2$ Algorithm

The proposed $IteM^2$ algorithm is described in Algorithm 1. It works as follows. In Step (1), we initialize using Algorithm 2. To be specific, we calculate the normalized affinity matrices using Equation (1), and initialize the function g_i for Task i to contain the label

information. Then from Step (2) to (25), we repeatedly update both the function g_i and the function f_{ik} by n_{iter} times. In particularly, between Steps (9) and (17), for the k^{th} view, we collectively update the functions f_{ik} for the tasks with this view. To be specific, we calculate the matrix A_3 as follows.

$$A_3 = A_2 A_1^{-1} \tag{7}$$

where A_1 denotes an $|I_k| \times |I_k|$ matrix with diagonal element $A_1(i,i)$ set to $a_{I_k(i)k} + 2b(|I_k|-1)$, ($I_k(i)$ is the i^{th} element of I_k), and the other elements set to $-2b$; A_2 denotes an $d_k \times |I_k|$ matrix whose i^{th} column is set to $a_{I_k(i)k} L^T_{I_k(i)k} g_{I_k(i)}$. Finally, in Step (26), we obtain the predicted class labels using Algorithm 3, which normalizes the function g_i according to the proportion of both classes in the labeled set of each task.

5.2. Analysis of $IteM^2$

The following theorem guarantees the optimality and convergence of the iteration process between Step (2) and (25) of $IteM^2$ algorithm.

Algorithm 1 $IteM^2$ Algorithm

Input: W_{ik}, a_{ik}, μ_i, S_{ij}, y_{il}, $i,j = 1,\ldots,T$, $k = 1,\ldots,V_i$, $l = 1,\ldots,m_i$, b, n_{iter}
Output: Predicted class labels $\hat{y}_{il}$, $i = 1,\ldots,T$, $l = m_i+1,\ldots,n_i$

1: Initialize using Algorithm 2
2: **for** $t = 1$ to n_{iter} **do**
3: **for** $i = 1$ to T **do**
4: $S_i = \cup_{j=1}^{T} S_{ij}$
5: **for** $k \notin S_i$ **do**
6: $f_{ik} = L_{ik}^T g_i$
7: **end for**
8: **end for**
9: **for** $k = 1$ to V **do**
10: Let I_k denote the set of indices such that $\forall s \in I_k$, $\exists s' \in I_k$, $k \in S_{ss'}$ and $\forall s' \notin I_k$, $k \notin S_{ss'}$
11: **if** I_k is not empty **then**
12: Calculate matrix A_3 using Equation (7)
13: **for** $i = 1$ to $|I_k|$ **do**
14: Set $f_{I_k(i)k}$ to be the i^{th} column of A_3
15: **end for**
16: **end if**
17: **end for**
18: **for** $i = 1$ to T **do**
19: Set $a_i = \sum_{k=1}^{V_i} a_{ik}$ and $g_i = \frac{\mu_i}{\mu_i + a_i} y_i$
20: **for** $k = 1$ to V_i **do**
21: $g_i = g_i + \frac{a_{ik}}{\mu_i + a_i} L_{ik} f_{ik}$
22: **end for**
23: **end for**
24: **end for**
25: Assign class labels using Algorithm 3

Algorithm 2 Initialization of $IteM^2$ Algorithm

Input: W_{ik}, y_{il}, $i = 1,\ldots,T$, $k = 1,\ldots,V_i$, $l = 1,\ldots,m_i$
Output: T_{ik}, L_{ik} and the initial value for g_i, $i = 1,\ldots,T$, $k = 1,\ldots,V_i$

1: **for** $i = 1$ to T **do**
2: **for** $k = 1$ to V_i **do**
3: Calculate T_{ik} and L_{ik} based on Equation (1)
4: **end for**
5: Initialize g_i such that $g_i(l) = y_{il}$, $l = 1,\ldots,m_i$, $g_i(l) = 0$, $l = m_i+1,\ldots,n_i$
6: **end for**

Algorithm 3 Label Assignment of $IteM^2$ Algorithm

Input: y_{il}, g_i, $i = 1,\ldots,T$, $l = 1,\ldots,m_i$
Output: Predicted class labels $\hat{y}_{il}$, $i = 1,\ldots,T$, $l = m_i+1,\ldots,n_i$

1: **for** $i = 1$ to T **do**
2: Set p_i to be the proportion of positive examples in the labeled set of Task i
3: Sort g_i in descending order. Set $\hat{y}^{il} = 1$ for the top p_i portion of the ranked list, and set $\hat{y}^{il} = -1$ for the remaining
4: **end for**

Theorem 5.1. *(Optimality and Convergence) If n_{iter} is sufficiently large, $V_i = v$, $\mu_i = \mu$, and $a_{ik} = a$, $i = 1,\ldots,T$, $k = 1,\ldots,v$, Step (2) to (25) of $IteM^2$ converge to the optimal solution of Equation (2).*

Proof. Taking the first derivative of $Q(f,g)$ with respect to g_i, we have $\frac{\partial Q(f,g)}{\partial g_i} = \sum_{k=1}^{V_i}(2a_{ik}g_i - 2a_{ik}L_{ik}f_{ik}) + 2\mu_i(g_i - y_i)$. Setting it to 0, we have

$$g_i = \frac{\mu_i}{\mu_i + \sum_{k=1}^{V_i} a_{ik}} y_i + \sum_{k=1}^{V_i} \frac{a_{ik}}{\mu_i + \sum_{k=1}^{V_i} a_{ik}} L_{ik} f_{ik} \tag{8}$$

$\forall k$, $1 \le k \le V_i$, if Task i does not share the k^{th} view with any other task, $\frac{\partial Q(f,g)}{\partial f_{ik}} = 2a_{ik}f_{ik} - 2a_{ik}L_{ik}^T g_i$. Setting it to 0, we have

$$f_{ik} = L_{ik}^T g_i \tag{9}$$

On the other hand, if Task i shares the k^{th} view with some other tasks, let I_{ik} denote the set of indices such that, $\forall s \in I_{ik}$, $k \in S_{is}$. Note that $I_k = \{i, I_{ik}\}$. In this case, we have $\frac{\partial Q(f,g)}{\partial f_{ik}} = 2a_{ik}f_{ik} - 2a_{ik}L_{ik}^T g_i + 4b\sum_{s \in I_{ik}}(f_{ik} - f_{sk})$. For all the other indices in I_k, we have a similar equation. Therefore, by setting all these equations to 0, the obtained f_{ik} functions for tasks with indices in I_k correspond to the columns of

$$A_3 = A_2 A_1^{-1} \tag{10}$$

Without loss of generality, assume that $I_1 = \{1,\ldots,r\}$, $r > 1$. When the conditions in the theorem are satisfied,

$$A_1^{-1} = \frac{1}{a^2+2rab}\begin{bmatrix} a+2b & 2b & \cdots & 2b \\ 2b & a+2b & \cdots & \vdots \\ \vdots & \vdots & \ddots & 2b \\ 2b & \ldots & 2b & a+2b \end{bmatrix}$$

Replacing f_{i1} in Equation (8) with the one obtained in Equation (10), we have, for $i = 1, \ldots, r$,

$$g_i = \frac{\mu}{\mu+av} y_i + \sum_{k=2}^{v} \frac{a}{\mu+av} L_{ik} f_{ik}$$
$$+ \frac{a}{\mu+av}\left(\sum_{j=1}^{r} \frac{2b}{a^2+2rab} a L_{i1} L_{j1}^T g_j + \frac{a}{a^2+2rab} a L_{i1} L_{i1}^T g_i\right)$$

In matrix form, the third term in the above equations (for all the g_is) can be collectively written as $\frac{a}{\mu+av} L_1 g$, where

$$L_1 = \begin{bmatrix} \frac{a^2+2ab}{a^2+2rab} & \frac{2ab}{a^2+2rab} & \cdots & \frac{2ab}{a^2+2rab} \\ \frac{2ab}{a^2+2rab} & \frac{a^2+2ab}{a^2+2rab} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \frac{2ab}{a^2+2rab} \\ \frac{2ab}{a^2+2rab} & \cdots & \frac{2ab}{a^2+2rab} & \frac{a^2+2ab}{a^2+2rab} \end{bmatrix} *$$
$$\left(\begin{bmatrix} L_{11} \\ \vdots \\ L_{r1} \end{bmatrix} \begin{bmatrix} L_{11}^T & \cdots & L_{r1}^T \end{bmatrix}\right)$$

Here $*$ denotes Khatri-Rao product, and $g = [g_1^T, \ldots, g_r^T]^T$. It is easy to see that the two parts of L_1 are both positive semi-definite; the eigenvalues of the first matrix are between 0 and 1; and the diagonal blocks of the second matrix also have eigenvalues between 0 and 1. Therefore, according to (Horn & Mathias, 1992), L_1 is positive semi-definite, and its eigenvalues are between 0 and 1. Similarly, we can show that L_k is positive semi-definite, and its eigenvalues are between 0 and 1, $k = 1, \ldots, v$. Therefore, by iteratively solving Equations (8), (9) and (10), Step (2) to (25) converge to the optimal solution of Equation (2). □

Regarding the time complexity of $IteM^2$, we have the following lemma, which indicates that the proposed $IteM^2$ algorithm is scalable to the size of the data set as well as the dimensionality of the feature space.

Lemma 5.2. *(Time Complexity) The time complexity of $IteM^2$ is $O(n_{iter}\left(\sum_{i=1}^{T} n_i \sum_{k=1}^{V_i} d_{ik} + d_S T^2 + VT^3\right) + \sum_{i=1}^{T} n_i \log n_i)$, where d_S is the sum of dimensionality of shared views whose $|I_k| > 0$.*

Proof. Omitted for brevity. □

6. Experimental Results

In this section, we present some experimental results showing the effectiveness of the proposed $IteM^2$ algorithm. To the best of our knowledge, there is no existing work for problems where multiple related tasks have both shared views and task-specific views. Therefore, we compare with the following algorithms:

1. SVM-2K (Farquhar et al., 2005): an optimization-based algorithm for problems with multiple views.
2. SMTL (Liu et al., 2007): a semi-supervised multi-task learning framework, which uses unlabeled data based on Markov random walk.
3. CASO (Chen et al., 2009): a multi-task learning algorithm, which improves the ASO algorithm (Ando & Zhang, 2005) with a novel regularizer.

In our experiments, we apply SVM-2K on the multiple views of each task respectively; we apply SMTL and CASO on the common views of all the tasks. To provide a fair comparison, we adjust the output of these competitors in the same way as Algorithm 3. We repeat all the experiments 40 times, and report both the average classification error and the standard deviation.

For the proposed $IteM^2$ algorithm, we simply set $a_{ik} = 1$, $i = 1, \ldots, T$, $k = 1, \ldots, V_i$ since there is no evidence showing the superiority of one view or another. Following the convention in (Zhou et al., 2003), we set $\mu_i = 0.01$, $i = 1, \ldots, T$. The number of iteration steps n_{iter} is set to 100, and b is set to 1 based on the parameter study in the next subsection. For SVM-2K, we set the parameters by cross-validation; for SMTL and CASO, we set the parameters according to (Liu et al., 2007) and (Chen et al., 2009) respectively.

6.1. Two Tasks with Non-identical Views

We first perform experiments on 20 newsgroups data set (Asuncion & Newman, 2007). On this data set, we created 3 problems. Each problem has 2 tasks, which share a common view consisting of the common vocabulary. The task specific vocabulary corresponds to the unique view of each task. Therefore, $T = 2$, $V = 3$, $V_1 = V_2 = 2$, $S_{12} = \{1\}$. For details, please refer to Table 1, where the number following 'P' denotes the problem index, the number following 'T' denotes the task index, and the number in the parenthesis is the number of examples.

Table 1. Task description for 20 newsgroups data set.

LABEL	+1	-1
P1T1	COMP.GRAPHICS (581)	REC.AUTOS (592)
P1T2	COMP.OS.MS-WINDOWS.MISC (572)	REC.MOTORCYCLES (596)
P2T1	COMP.SYS.IBM.PC.HARDWARE (587)	SCI.MED (594)
P2T2	COMP.SYS.MAC.HARDWARE (575)	SCI.SPACE (593)
P3T1	REC.AUTOS (592)	TALK.POLITICS.MIDEAST (564)
P3T2	REC.MOTORCYCLES (596)	TALK.POLITICS.GUNS (545)

Figure 2 shows the results of $IteM^2$ when we vary the value of b from 100 to 0. When $b = 0$, the performance is the worst, especially when the number of labeled examples from each task is small. This is because the label information from other tasks is not utilized. On the other hand, the performance of $IteM^2$ is quite robust over a wide range of values for b. Therefore, in subsequent experiments, we fix $b = 1$.

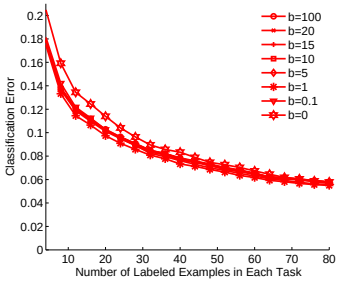
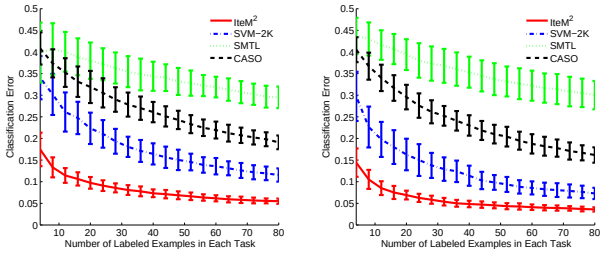


Figure 2. Parameter study: 20 newsgroups data set P1T1.

Figure 3 to Figure 5 show the comparison results on the 3 problems respectively. In each figure, the left subfigure shows the classification error for Task 1 vs. the number of labeled examples in each task, and the right subfigure shows the classification error for Task 2 vs. the number of labeled examples in each task. From these figures, we can see that the performance of $IteM^2$ is always the best on both tasks, since SVM-2K does not utilize the label information from other tasks, whereas SMTL and CASO do not consider the consistency among different views of a single task. Furthermore, notice that the difference between SVM-2K and $IteM^2$ is significant when the number of labeled examples is small. This observation, which is common in all the experiments, is consistent with our intuition because labeled examples from other tasks are particularly useful when we do not have many labeled examples in a single task.

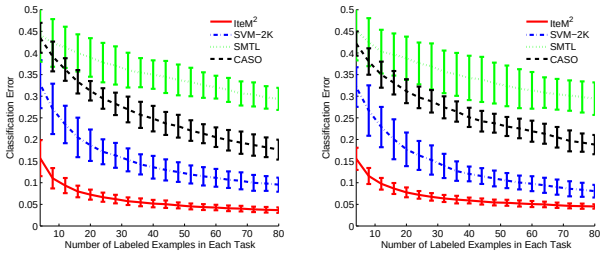
6.2. Multiple Tasks with Identical Views

Next, we test the performance of $IteM^2$ on WebKB data set, which was used to study the co-training algorithm in (Blum & Mitchell, 1998). This data set consists of 1051 web pages collected from the computer science departments of 4 universities. The goal is to classify each web page as course or non-course. On this data set, we have 4 tasks, each of which consists of the web pages from one university. For each task, we have 3 views, which correspond to the words in the web page, the words in the anchor text of hyperlinks pointing to that page, and the words in the title of the web page. Notice that all 3 views are shared by the 4 tasks. Therefore, $T = 4$, $V = 3$, $V_i = 3$, and $S_{ij} = \{1, 2, 3\}$, $i, j = 1, \ldots, 4$, $i \neq j$. Notice that for such problems (multiple tasks with identical views), the input to $IteM^2$, SMTL and CASO are the same since all the tasks have identical views. Figure 6 shows the average classification error of all the tasks vs. the number of labeled examples in each task. We can see that the performance of $IteM^2$ is significantly better than SMTL and CASO, which indicates the importance of leveraging the consistency of multiple views.



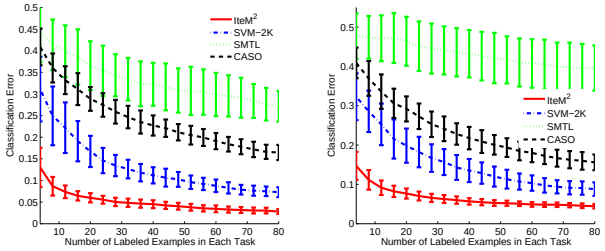
(a) Task 1 (b) Task 2

Figure 3. Classification error: 20 newsgroups data set P1.



(a) Task 1 (b) Task 2

Figure 4. Classification error: 20 newsgroups data set P2.



(a) Task 1 (b) Task 2

Figure 5. Classification error: 20 newsgroups data set P3.

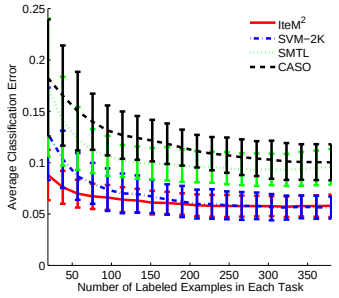


Figure 6. Average classification error: WebKB data set.

6.3. Multiple Tasks and Non-identical Views

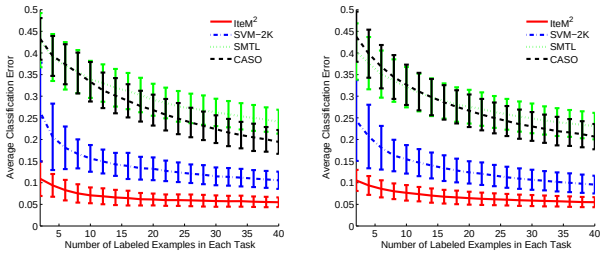
Finally, we study the more general case where we have multiple tasks with non-identical views. To this end,

we use the email spam data set from ECML 2006 discovery challenge[2]. Here we have 2 problems. In Problem A, we have emails from 3 different users (2500 emails per user); whereas in Problem B, we have emails from 15 different users (400 emails per user). The goal is to classify spam vs. ham. For both problems, we create different tasks for different users. Similar as in Subsection 6.1, the common view of all the tasks correspond to the common vocabulary, and the unique view of each task correspond to the task-specific vocabulary. Therefore, for Problem A, $T = 3$, $V = 4$, $V_1 = V_2 = V_3 = 2$, and $S_{12} = S_{13} = S_{23} = \{1\}$; for Problem B, $T = 15$, $V = 16$, $V_i = 2$, and $S_{ij} = \{1\}$, $i, j = 1, \ldots, 15$, $i \neq j$.

Figure 7 shows the average classification error of all the tasks vs. the number of labeled examples in each task for Problem A and B. We can see that again the performance of $IteM^2$ is the best in this general setting. On the other hand, SVM-2K performs better than multi-task learning algorithms. This may be due to the fact that SMTL and CASO only use the common vocabulary, which limits their discrimination power.

7. Conclusion

In this paper, we introduce M^2TV learning for problems with dual-heterogeneity, i.e., multiple related tasks have both shared views and task-specific views. We propose a graph-based framework ($GraM^2$) for M^2TV learning, which has a natural connection to RKHS. Furthermore, we propose an effective algorithm ($IteM^2$) to solve our $GraM^2$ framework, which is guaranteed to converge to the optimal solution in a scalable way. Experimental results on several real data sets demonstrate the effectiveness of our method.



(a) Problem A (b) Problem B

Figure 7. Average classification error: ECML data set.

References

Ando, Rie Kubota and Zhang, Tong. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

Argyriou, Andreas, Evgeniou, Theodoros, and Pontil, Massimiliano. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[2]http://www.ecmlpkdd2006.org/challenge.html

Asuncion, A. and Newman, D.J. UCI machine learning repository, 2007. URL http://archive.ics.uci.edu/ml/.

Blum, Avrim and Mitchell, Tom M. Combining labeled and unlabeled sata with co-training. In *COLT*, 1998.

Cavallanti, Giovanni, Cesa-Bianchi, Nicol'o, and Gentile, Claudio. Linear algorithms for online multitask and multiview classification. *Journal of Machine Learning Research*, 11:2901–2934, 2010.

Chen, Jianhui, Tang, Lei, Liu, Jun, and Ye, Jieping. A convex formulation for learning shared structures from multiple tasks. In *ICML*, pp. 18, 2009.

Chen, Ning, Zhu, Jun, and Xing, Eric P. Predictive subspace learning for multi-view data: a large margin approach. In *NIPS*, 2010.

Evgeniou, T., Micchelli, C.A., and Pontil, M. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

Farquhar, J.D.R., Hardoon, D.R., Meng, H., Shawe-Taylor, J., and Szedmák, S. Two view learning: Svm-2k, theory and practice. In *NIPS*, 2005.

Horn, Roger A. and Mathias, Roy. Block-matrix generalizations of schur's basic theorems on hadamard products. *Linear Algebra and its Applications*, 1992.

Jacob, Laurent, Bach, Francis, and Vert, Jean-Philippe. Clustered multi-task learning: A convex formulation. In *NIPS*, pp. 745–752, 2008.

Liu, Qiuhua, Liao, Xuejun, and Carin, Lawrence. Semi-supervised multitask learning. In *NIPS*, 2007.

Nigam, Kamal and Ghani, Rayid. Analyzing the effectiveness and applicability of co-training. In *CIKM*, pp. 86–93, 2000.

Parameswaran, Shibin and Weinberger, Kilian Q. Large margin multi-task metric learning. In *NIPS*, 2010.

Sindhwani, Vikas and Rosenberg, David S. An rkhs for multi-view learning and manifold co-regularization. In *ICML*, pp. 976–983, 2008.

Zhao, Peilin and Hoi, Steven C.H. Otl: A framework of online transfer learning. In *ICML*, pp. 1231–1238, 2010.

Zhou, Dengyong, Weston, Jason, Gretton, Arthur, Bousquet, Olivier, and Schölkopf, Bernhard. Ranking on data manifolds. In *NIPS*, 2003.

A. Proof Sketch of Theorem 4.2

Similar to (Sindhwani & Rosenberg, 2008), we can show that $\mathcal{H}$ is a Hilbert space. To further prove that it is an RKHS, we only consider the case where $z \in N_1$. The case where $z \in N_2$ can be proven similarly. First, based on Equations (3) and (4), we can show that $k_{\mathcal{H}}(z, \cdot) \in \mathcal{H}$. Next we prove the reproducing property of $k_{\mathcal{H}}(z, \cdot)$. $\forall h' = [(h'_1)^T, (h'_2)^T]^T \in \mathcal{H}$, we calculate $\langle h', k_{\mathcal{H}}(z, \cdot)\rangle_{\mathcal{H}}$ as follows.

$$\langle h', k_{\mathcal{H}}(z, \cdot)\rangle_{\mathcal{H}} = \langle [(h'_1)^T, (h'_2)^T]^T, [h_1^T, h_2^T]^T\rangle_{\mathcal{H}}$$

$$= \langle h_1, h'_1\rangle_{\mathcal{H}_1} + \langle h_2, h'_2\rangle_{\mathcal{H}_2} + b(h_1(F) - h_2(F))^T(h'_1(F) - h'_2(F))$$

$$= h'_1(z) - b\vec{k}_{\mathcal{H}_1}(z, F)(I_{d_{11}\times d_{11}} + bS)^{-1}h'_1(F)$$

$$+ b\vec{k}_{\mathcal{H}_1}(z, F)(I_{d_{11}\times d_{11}} + bS)^{-1}h'_2(F)$$

$$+ b(h_1(F) - h_2(F))^T h'_1(F) - b(h_1(F) - h_2(F))^T h'_2(F)$$

$$= h'_1(z) = h'(z)$$

where the second last equation is based on the following.

$$h_1^T(F) - h_2^T(F) = \vec{k}_{\mathcal{H}_1}(z, F) - b\vec{k}_{\mathcal{H}_1}(z, F)(I_{d_{11}\times d_{11}} + bS)^{-1}S$$

$$= \vec{k}_{\mathcal{H}_1}(z, F)(I_{d_{11}\times d_{11}} + bS)^{-1}$$