Learning with Dual Heterogeneity: A Nonparametric Bayes Model

Hongxia Yang
IBM Watson Research Center
1101 Kitchawan Rd
Yorktown Heights, NY, 10598
yangho@us.ibm.com

Jingrui He
School of Computing, Informatics, Decision Systems Engineering
Arizona State University
699 S Mill Ave, Tempe, AZ 85281
jingrui.he@gmail.com

ABSTRACT

Traditional data mining techniques are designed to model a single type of heterogeneity, such as multi-task learning for modeling task heterogeneity, multi-view learning for modeling view heterogeneity, etc. Recently, a variety of real applications emerged, which exhibit dual heterogeneity, namely both task heterogeneity and view heterogeneity. Examples include insider threat detection across multiple organizations, web image classification in different domains, etc. Existing methods for addressing such problems typically assume that multiple tasks are equally related and multiple views are equally consistent, which limits their application in complex settings with varying task relatedness and view consistency. In this paper, we advance state-of-the-art techniques by adaptively modeling task relatedness and view consistency via a nonparametric Bayes model: we model task relatedness using normal penalty with sparse covariances, and view consistency using matrix Dirichlet process. Based on this model, we propose the *NOBLE* algorithm using an efficient Gibbs sampler. Experimental results on multiple real data sets demonstrate the effectiveness of the proposed algorithm.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: [nonparametric statistics]; I.5.1 [**Pattern Recognition**]: Models—*statistical*

Keywords

Nonparametric Bayes modeling; multi-task multi-view; Gibbs sampler.

1. INTRODUCTION

Nowadays, we are facing large amount of data in a variety of areas, such as social media, manufacturing, traffic analytics, etc. A common challenge in these areas is how to handle multiple types of data heterogeneity. For example, in social media, we may have micro-blogs coming from heterogeneous sources, such as Facebook and Twitter, and each micro-blog may be characterized by heterogeneous features, such as key words, hashtags, number of re-tweets, number of Facebook likes, etc; in manufacturing, we may have products from heterogeneous manufacturing lines, and each product may be characterized by heterogeneous environmental variables, such as temperature, pressure, etc; in traffic analytics, we can collect traffic information from heterogeneous geographic locations (e.g., different states), and for each location, we may have heterogeneous traffic indicators, such as volume, GPS positions, etc.

Recent years have seen growing interest in addressing problems with multiple types of data heterogeneity [22, 19, 14, 43, 20, 21, 42]. In particular, for problems with dual heterogeneity, i.e., both task and view heterogeneity, researchers have proposed multi-task multi-view learning, or M^2TV learning, to jointly learn in multiple related tasks with overlapping, partially overlapping or completely different feature spaces [22, 43, 21, 42]. Compared with traditional multi-task learning [12, 46, 11, 38, 30], where the feature space is *homogeneous* across different tasks, i.e., a single view, M^2TV learning is able to handle *heterogeneous* feature spaces; compared with traditional multi-view learning [13, 24, 16, 28, 8], where the examples come from a *homogeneous* task, i.e., a single task, M^2TV learning is able to leverage *heterogeneous* (related) tasks to improve the learning performance in each task.

A key question in M^2TV learning is how to model the relatedness among multiple tasks/views. Existing methods for M^2TV learning [22, 43, 21, 42] usually assume that all the tasks are equally related, and all the views are equally consistent. Therefore, they mainly focus on exploring various types of task relatedness and view consistency. In this paper, we go one step further, and study: (1) if all the tasks are equally related and all the views are equally consistent; (2) to what extent the multiple tasks are related and the multiple views are consistent. This is motivated by the fact that in many real applications, it is often not known a priori the degree of relatedness among multiple tasks and consistency among multiple views. In the adversarial cases where some tasks are *negatively* related to the others and some views are contaminated by noise, simply applying the existing methods for M^2TV learning may even hurt the performance. Although in traditional multi-task learning, there has already been some work accommodating various task

KDD'14, August 24–27, 2014, New York, NY, USA.

http://dx.doi.org/10.1145/2623330.2623727.

relatedness [45, 46, 12, 11], to the best of our knowledge, our work is the first to study this problem in the context of M^2TV learning.

To this end, motivated by the successful application of Bayesian hierarchical modeling in multi-task learning and multi-view learning [19, 3, 5], we propose a nonparametric Bayes method for M^2TV learning. In this method, task relatedness is modeled via a normal penalty that decomposes the full covariance matrix into the Kronecker product, and view consistency is modeled via a matrix Dirichlet process. Furthermore, we design the *NOBLE* algorithm, which stands for ***NO***nparametric ***B***ayes ***LE***arning with dual heterogeneity. It is based on an efficient Gibbs sampler scalable to relatively high dimensions. The main contributions of this paper can be summarized as follows.

1. For the first time, in the context of M^2TV learning, we study problems where multiple tasks may exhibit different degree of relatedness, and multiple views may exhibit different degree of consistency;
2. We propose a nonparametric Bayes method for M^2TV learning which adaptively learns various task relatedness and view consistency;
3. We design the *NOBLE* algorithm based on an efficient Gibbs sampler scalable to relatively high dimensions;
4. We compare the performance of our proposed *NOBLE* algorithm with state-of-the-art techniques on various real data sets.

The rest of the paper is organized as follows. In Section 2, we briefly review the related work. The nonparametric Bayes method for M^2TV learning is proposed in Section 3, followed by the algorithm description of *NOBLE* in Section 4. Section 5 compares *NOBLE* with state-of-the-art methods on real data sets. Finally, we conclude in Section 6.

2. RELATED WORK

In this section, we briefly review the related work in heterogeneous learning and Dirichlet process mixture models.

2.1 Heterogeneous Learning

The goal of heterogeneous learning is to leverage multiple types of heterogeneities (e.g., task heterogeneity, view heterogeneity, instance heterogeneity, label heterogeneity, etc) to improve the performance of predictive modeling. For example, in [22, 23, 33, 43, 19, 21, 42], the authors jointly modeled the task and view heterogeneities; in [41] the authors jointly modeled the view and instance heterogeneities; in [26], the authors jointly modeled the instance and label heterogeneities.

For problems with both task and view heterogeneity, the authors of [22] focused on multiple tasks with completely different feature spaces, and proposed to construct a single prediction model in the shared induced space; the authors of [23] proposed to learn shared predictive structures on common views from multiple related tasks, and used the consistency among different views to improve the performance; the authors of [43] used co-regularization in each task to obtain a linear mapping, and used additional regularization functions across different tasks to impose task relatedness; the authors of [19] proposed a latent probit model to jointly learn the domain transforms, and a probit classifier shared in the common domain; the authors of [42] proposed a large margin framework to address transfer learning problems[1] with the same set of views in the source and target domains; the authors of [21] proposed a graph-based framework to model the relationship among multiple tasks/views, and designed an iterative algorithm $IteM^2$ to find the classification function. The major difference between our work and the existing work is the following. Existing methods assume that all the tasks are equally related and all the views are equally consistent. Therefore, they mainly focus on exploring various kinds of task relatedness and view consistency. In our work, we go one step further, and study: (1) if all the tasks are equally related and all the views are equally consistent; and (2) to what extent the multiple tasks are related and the multiple views are consistent.

The problem of varying task relatedness has been studied in traditional multi-task learning. For example, in [37], the authors proposed to use bipartite graphs to represent multi-task learning, and made use of Gaussian process to model varying task relatedness; in [12], the authors proposed a robust multi-task learning (RMTL) algorithm that learns multiple tasks simultaneously as well identifies the irrelevant tasks; in [46], the authors showed the equivalent relationship between alternating structure optimization and clustered multi-task learning; etc. However, the above methods and analysis only apply in the multi-task setting, and it is not straightforward to extend them to M^2TV learning.

In particular, Bayesian modeling has been widely used in multi-task learning and multi-view learning over the last decade. Research work dedicated to Bayesian hierarchical modeling has demonstrated effectiveness and improvement in performance [19, 3, 5]. The proposed methods have been successfully applied to different areas, such as information retrieval [7] and computer vision [30]. Typical approaches to transfer information among multiple tasks/views include: sharing hidden nodes in neural networks, placing a common prior in hierarchical models, sharing a common structure on the predictor space, and structured regularization in kernel methods, among others [19, 38, 9, 40, 39].

2.2 Dirichlet Process Mixture Models

In this paper, we propose to use Dirichlet process (DP) prior to encourage view clustering in the context of M^2TV learning. Before presenting our model, we briefly review DP mixture models. In a Bayesian mixture model, we assume that the true density of the response $\boldsymbol{Y}$ can be written as a mixture of parametric densities, conditioned on a hidden parameter θ. For example, in a Gaussian mixture, θ corresponds to the mean μ and variance σ^2. The marginal probability of an observation is given by a continuous mixture, $f(y) = \int_{\mathcal{T}} f(y|\theta)P(d\theta)$, where $\mathcal{T}$ is the set of all possible parameters and the prior P is a measure on that space. DP models uncertainty about the prior density P [17, 2]. If P is drawn from a Dirichlet process then it can be analytically integrated out of the conditional distribution of θ_T given $\theta_{1:(T-1)}$, where θ_T denotes the T^{th} parameter for observation y_T. Specifically, the random variable θ_T has a Polya

[1]Transfer learning is very similar to multi-task learning except that in transfer learning, we only care about the learning performance in the target domain.

urn distribution [6]:

$$\theta_T|\theta_{1:(T-1)} \sim \frac{1}{\alpha+T-1}\sum_{t=1}^{T-1}\delta_{\theta_t} + \frac{\alpha}{\alpha+T-1}G_0.$$

The above equation reveals the clustering property of the joint distribution of $\theta_{1:T}$, where there is a positive probability that each θ_t will take on the value of another $\theta_{t'}$, leading some of the parameters to share values. This equation also makes clear the roles of α and G_0. The unique values of $\theta_{1:(T-1)}$ are drawn independently from G_0; the parameter α controls how likely θ_T is to be a newly drawn value from G_0 rather than to take one of values from $\theta_{1:(T-1)}$. G_0 controls the distribution of a new component.

In a DP mixture, θ is a latent parameter to an observed data point y [2]: $P \sim \mathrm{DP}(\alpha G_0)$, $\theta_t \sim P$, $y_t|\theta_t \sim f(\cdot|\theta_t)$. Examining the posterior distribution of $\theta_{1:T}$ given $y_{1:T}$ brings out its interpretation as an "infinite clustering" model. Because of the clustering property, observations are grouped by their shared parameters. Unlike finite clustering models, however, the number of groups is random and unknown. Moreover, a new data point can be assigned to a new cluster that was not previously seen in the data.

However, the DP prior does not allow local clustering of tasks/views with respect to a subset of the feature vector without making independence assumptions. Considering sample s ($s = 1, \ldots, n_t$) from task (or view) t ($t = 1, \ldots, T$), suppose that the response variable is y_{ts} and related feature vector is $\boldsymbol{x}_{ts}$ with dimension n_p by 1. A common strategy for such problem is to use a hierarchical model of the form $y_{ts} \sim p(\boldsymbol{x}_{ts}, \boldsymbol{f}_t, \boldsymbol{\phi})$, where $p(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{\phi})$ is the conditional distribution of y given feature vector $\boldsymbol{x}$, parameters $\boldsymbol{f}$ and $\boldsymbol{\phi}$. $\boldsymbol{\phi}$ are global parameters and $\boldsymbol{f} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_T)$ is a vector of task-specific (or view-specific) coefficients. We could specify independent DP priors for the coefficients [35, 14]: $f_{tp} \overset{\text{iid}}{\sim} G_t, G_t \sim DP(\alpha_j, G_{0j})$ for $p = 1, \ldots, n_p$. This approach allows differential clustering of the coefficients for different feature components, however, independence is assumed across the feature components. This is unappealing, because $f_{tp} = f_{t'p}$ provides information that tasks t and t' are similar, which should intuitively increase the probability that $f_{tp} = f_{t'p'}$, for $p \neq p'$. Motivated by this desire to borrow information across related feature components and tasks simultaneously, [35] propose a matrix stick-breaking process (MSBP) by assuming

$$f_{tp} \overset{\text{ind}}{\sim} G_{tp}, \quad \mathcal{G} \sim \mathcal{P},$$

where $\mathcal{G} = \{G_{tp}, p = 1, \ldots, n_p, t = 1, \ldots, T\}$ is a matrix of random probability measures, and $\mathcal{P}$ is a probability measure on $(\Omega, \mathcal{G})$, with Ω the space of $T \times n_p$ matrices with the (t,p)th element a probability measure on $(\mathcal{X}_t, \mathcal{B}_t)$. Here, $\mathcal{G}$ is a σ-algebra of subsets of Ω and $\mathcal{B}_t$ is a Borel σ-algebra of subsets of $\mathcal{X}_t$, $f_{tp} \in \mathcal{X}_t$. The proposed MSBP allows separate clustering and borrowing of information for the different feature components through

$$G_{tp} = \sum_{h=1}^{H}\{V_{tph}\prod_{l<h}V_{tpl}\}\delta_{\Theta_{ph}}, \quad \Theta_{ph} \overset{\text{ind}}{\sim} G_{0p},$$

$$V_{tph} = U_{th}W_{ph}, \quad U_{th} \overset{\text{iid}}{\sim} \mathrm{Beta}(1,\alpha), \quad W_{ph} \overset{\text{iid}}{\sim} \mathrm{Beta}(1,\beta).$$

To provide an intuitive explanation for the above formulation, we first consider the sticks W_{ph}. If W_{ph} is large for a particular index h^*, then the corresponding parameters Θ_{ph^*} is likely to be shared among multiple tasks. We also note that this sharing among tasks is encouraged by large U_{th^*}. Since U_{th^*} may be large for multiple different tasks t, this implies that if parameter sharing occurs for one predictor among the multiple tasks, then it is also likely that there will be sharing for other predictors. We can therefore generalize the following key properties of MSBP: (i) if a given parameter for predictor p, Θ_{ph^*}, is shared among some of the tasks, it is more likely to be shared among other tasks; (ii) if sharing occurs between multiple predictors for a subset of tasks, then it is more encouraged that sharing will occur between other predictors within these tasks.

3. NONPARAMETRIC BAYES LEARNING WITH DUAL HETEROGENEITY

3.1 Notation

Suppose that we have T tasks and V views in total. For the v^{th} view, there are d_v features. For the t^{th} task ($t = 1, \ldots, T$), there are n_t examples and each example can be represented as $\boldsymbol{x}_{ts} = [(\boldsymbol{x}_{ts1})', \ldots, (\boldsymbol{x}_{tsV})']'$ with label $\hat{y}_{ts}$ ($s = 1, \ldots, n_t$), where $()'$ denotes vector transpose. $\boldsymbol{x}_{tsv} \in \mathbb{R}^{d_v}$ denotes the features from the v^{th} view ($v = 1, \ldots, V$) of the sth example in the t^{th} task, and $\hat{y}_{ts}$ is either discrete for classification problems, or real-valued for regression problems. Notice that if a certain view is missing, the associated features will all be 0. Therefore, our problem setting is essentially the same as in [21] where some views are shared by multiple tasks, and some views are task specific. Without loss of generality, suppose that we know the output $\hat{y}_{t1}, \ldots, \hat{y}_{tm_t}$ of the first m_t examples, where m_t is usually much smaller than n_t. Our goal is to leverage both the label information from all the related tasks, as well as the consistency among different views of a single task to predict the output of the remaining $n_t - m_t$ examples.

3.2 Model Formulation

In our proposed model, we first decompose each task into multiple single-view models. Each of them generates a predictor based on the features in the single view, which can be used to make predictions on future unseen examples. Here we relax the common assumption in multi-view learning [8, 29, 34] that different views are conditionally independent given the class label. To be specific, for the t^{th} task ($t = 1, \ldots, T$), we use a mixture linear regression model for the estimated output $\hat{y}_{ts}$ ($s = 1, \ldots, n_t$) by averaging the prediction results from all single-view models as follows:

$$\hat{y}_{ts} = \sum_{v=1}^{V}(\boldsymbol{x}_{tsv})'\boldsymbol{f}_{tv} + \epsilon_{ts},$$

where $\boldsymbol{f}_{tv} \in \mathbb{R}^{d_v}$ is the coefficient vector, and $\epsilon_{tsv} \in \mathbb{R}$ is the observational error. Based on the above model, we estimate the task relatedness and the view consistency as follows.

1. **Task Relatedness**: Here we use a Gaussian process defined on ϵ_{ts} to model the task relatedness. To be specific, we assume that $\boldsymbol{\epsilon}_s = \{\epsilon_{ts}\}_{t=1,\ldots,T} \sim \mathrm{N}(0, K)$, where $K \in \mathbb{R}^{T\times T}$ is the kernel matrix of the Gaussian process, and it is the key to determining the various task relatedness. Different from [37], where only a single information source is used to obtain the kernel function,

in this paper, we fully leverage the multi-view property to estimate K in a more reliable way. To be specific, in order to estimate K, we define a task graph as follows: the graph consists of T nodes with each node representing a single task; let $B \in \mathbb{R}^{T\times T}$ denote the adjacency matrix of the graph, whose element in the t^{th} row and t'^{th} column is $B_{tt'} = \frac{1}{n_t n_{t'}} \sum_{s=1}^{n_t} \sum_{s'=1}^{n_{t'}} < \boldsymbol{x}_{ts}, \boldsymbol{x}_{t's'} >$, where $t, t' = 1, \ldots, T$, and $< \cdot, \cdot >$ denotes vector inner product. For this graph, we can compute the Laplacian $\Delta = D - B$, where $D \in \mathbb{R}^{T\times T}$ is a diagonal matrix with each diagonal element equal to the row sum of B. Using Δ, we obtain K as follows:

$$K = \big[\beta(\Delta + \frac{1}{\sigma^2}\mathbf{I})\big]^{-1},$$

where both β and σ^2 are positive parameters. In particular, β controls the overall sharpness of the distribution: large values of β mean that the distribution is more peaked around its mean. For more flexibility, we let $\beta \sim \text{Ga}(a, b)$, which stands for Gamma distribution with shape parameter a and scale parameter b. It will be adapted to the data through adjusting the distribution related parameters a and b. σ^2 controls the amount of regularization. For this parameter, we could use the following prior $\sigma^2 \sim \text{IG}(c, d)$, which stands for Inverse-Gamma distribution with shape parameter c and scale parameter d.

We would like to point out several important aspects of the proposed Gaussian process. First, the kernel matrix K, whose elements indicate the similarity among various tasks, depends on the inverse of the regularized graph Laplacian Δ. Therefore, the relatedness between two tasks is global in the sense that it depends on all the tasks. Second, if we also have unlabeled data in addition to the labeled training data, all the unlabeled data can be used to define the adjacency matrix B (since it does not require label information), thus making it more robust to noise. Finally, the adjacency matrix B depends on the features from all the views through $\boldsymbol{x}_{ts}$. It tends to be more reliable if certain views have been contaminated by noise.

2. **View Consistency**: To estimate the various view consistency, we jointly model the coefficient vectors $\boldsymbol{f}_{tv}$ $(v = 1, \ldots, V)$ through:

$$\begin{pmatrix} \boldsymbol{f}_{t1} \\ \vdots \\ \boldsymbol{f}_{tV} \end{pmatrix} \sim \text{N}\left(\mathbf{0}\ , \begin{bmatrix} \Psi_{11} & \Psi_{12} & \cdots & \Psi_{1V} \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_{V1} & \Psi_{V2} & \cdots & \Psi_{VV} \end{bmatrix} \right)$$

where $\Psi_{vv'} \in \mathbb{R}_+^{d_v \times d_{v'}}$ denotes the covariance matrix between the v^{th} and the v'^{th} views. $\Psi_{vv'} = \Psi_{v'v}$.

Furthermore, a Dirichlet Process (DP) prior can be used here to encourage view cluster. However, without the conditional independence assumption, the DP prior does not allow local clustering of views with respect to a subset of the feature vectors. To address this problem, we extend the matrix DP prior [15] to define the covariance matrix $\Psi_{vv'}$, which encourages cross-view sharing of data. To be specific, we borrow information by incorporating dependency in the prior distributions for the matrices $\{\Psi_{vv'}\}$.

We start by assuming for $v \geq v' \geq 1$,

$$\Psi_{vv'} \overset{\text{ind}}{\sim} F_{vv'}, \quad \mathcal{F} \sim \mathcal{P},$$

Here $\mathcal{F} = \{F_{vv'}, V \geq v \geq v' \geq 1\}$ is a matrix of random probability measures. Let Ω be the space of symmetric $V \times V$ matrices and $\mathcal{F}$ will be a σ-algebra of subsets of Ω. $\mathcal{P}$ is a probability measure on $(\Omega, \mathcal{F})$.

Next, our focus is on the specification of $\mathcal{P}$. Assuming each element in $\mathcal{F}$ has a stick-breaking representation, i.e.,

$$F_{vv'} = \sum_{h=1}^{\infty} \big\{W_{vv',h} \prod_{l<h}(1 - W_{vv',l})\big\}\delta_{\Theta_h}, \Theta_h \overset{\text{ind}}{\sim} G, \quad (1)$$

where $\boldsymbol{W}_{vv'} = \{W_{vv',h}, h = 1, \ldots, \infty\}$, for $V \geq v \geq v' \geq 1$, is an array of random stick-breaking weights. $\Theta_h \in \mathbb{R}_+^{d_v \times d_v}$ stands for the latent covariance matrix[2] that is drawn from the base measure G, which usually takes the Inverse-Wishart (IW) distribution. Notice that similar to the usual Dirichlet Process, $\Psi_{vv'}$ equals to Θ_h with probability proportional to $W_{vv',h} \prod_{l<h}(1-W_{vv',l})$.

Dependency within dimensions of $\mathcal{F}$ will be incorporated through dependent stick-breaking weights and the common parametric prior G. For the stick-breaking weights, we decompose them as follows

$$W_{vv',h} = \gamma_{vh}\gamma_{v'h}, \gamma_{vh} \sim \text{Beta}(1, \alpha), \alpha \overset{\text{ind}}{\sim} \text{Ga}(1, \alpha_0),$$

where both γ_{vh} and $\gamma_{v'h}$ are random variables with the same Beta distribution, $\alpha > 0$ is a parameter in the Beta distribution, and $\alpha_0 > 0$ is the scale parameter in the Gamma distribution. In this way, we guarantee the symmetric property: $W_{vv',h} = W_{v'v,h}$. Furthermore, according to [15], the definition of γ_{vh} ensures that

$$\sum_{h=1}^{\infty} \big\{W_{vv',h} \prod_{l<h}(1 - W_{vv',l}) = 1\big\}$$

Therefore, Equation (1) is a valid probability measure.

We use the following example to show the intuition of the above formulation. Let $V = 4$, and $V_1, \ldots, V_4$ stand for the four different views. Then the probability that two covariance matrices $\Psi_{V_1V_2}$ and $\Psi_{V_1V_3}$ are same can be computed as follows.

$$\Pr(\Psi_{V_1V_2} = \Psi_{V_1V_3}) = \frac{1}{(\alpha+1)(\alpha+2) - 1}$$

Furthermore, the conditional probability of these two matrices being the same given that $\Psi_{V_4V_2} = \Psi_{V_4V_3}$ can be computed as follows.

$$\lim_{\alpha\to 0} \Pr(\Psi_{V_1V_2} = \Psi_{V_1V_3} | \Psi_{V_4V_2} = \Psi_{V_4V_3}) = \frac{1}{\alpha+1}$$

From the above equations, we can see that the probability of $\Psi_{V_1V_2}$ and $\Psi_{V_1V_3}$ being the same ranges between 0 and 1, depending on the value of the parameter α. Both converge to 1 in the limit as $\alpha \to 0$ and to 0 as $\alpha \to \infty$. We can verify that $\Pr(\Theta_{V_1V_2} = \Theta_{V_1V_3}) \leq \Pr(\Theta_{V_1V_2} = \Theta_{V_1V_3} | \Theta_{V_4V_2} = \Theta_{V_4V_3})$. It means given that view 2, 4 and

[2]For the sake of explanation, we assume that d_v is a constant for $v = 1, \ldots, V$; otherwise we fill in 0 values to make the dimensionality of each view equal.

view 3, 4 are equally correlated in terms of the covariance matrices, then there will be an increased probability that view 1, 2 and view 1, 3 are equally correlated.

Finally, for the base measure G of the view covariance matrix, we consider the following degenerate distribution:

$$G = \pi \mathrm{I}_0 + (1-\pi)G_0, G_0 \sim \mathrm{IW}(\nu, \Psi_0)$$

where $0 \leq \pi \leq 1$, ν is the degrees of freedom of the Inverse-Wishart distribution, $\Psi_0 \in \mathbb{R}_+^{d_v \times d_v}$ is the scale matrix. When $\Psi_{vv'}$ falls into the I_0 cluster, the corresponding covariance matrix will be a zero matrix, and the nonsignificant $\boldsymbol{f}_{tv}$ will be set to 0.

Figure 1 shows the graphical representation of the proposed model. To generalize, for each example s in the task t (y_{st}), task relatedness is characterized through K where β controls the overall sharpness of the distribution and σ^2 controls the amount of regularization. On the other hand, there is a view-specific feature $\boldsymbol{x}_{tsv}$ for the s example in the t^{th} task and we use $\boldsymbol{f}_{tv}$ to characterize v^{th} view effect in the t^{th} task. We extend the DP to characterize the covariance matrix $\Psi_{vv'}$ in order to cluster the coefficients $\boldsymbol{f}_{tv}$. $\mathbf{W}$ and α characterize the stick weights; G characterizes the base measure for DP, which is a degenerate distribution with probability π to be the null matrix and with probability $1-\pi$ to be an Inverse-Wishart distribution G_0 with degrees of freedom ν and scale matrix Ψ_0.

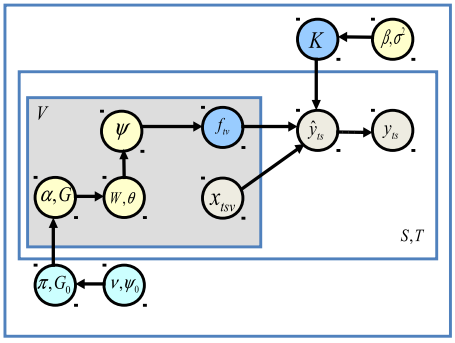

Figure 1: Graphical representation for the proposed model.

4. THE PROPOSED ALGORITHM

In this section, we present the *NOBLE* algorithm, which stands for ***NO***nparametric ***B***ayes ***LE***arning with dual heterogeneity. It is based on an efficient Gibbs algorithm that is scalable to relatively high dimensions. For simplicity, we assume that $n_t = S$ in the following. In particular, each iteration of the Gibbs sampler draws samples through the following sequence. The joint likelihood of the samples is as follows:

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{f}, K) = (2\pi)^{-\frac{TS}{2}} |\mathrm{I}_S \otimes K|^{-\frac{1}{2}} \cdot \exp\left\{ -\frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{f})'(\mathrm{I}_S \otimes K)^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{f}) \right\} \quad (2)$$

where

$$\boldsymbol{y} = \begin{pmatrix} y_{11} \\ \vdots \\ y_{1S} \\ \vdots \\ y_{21} \\ \vdots \\ y_{2S} \\ \vdots \\ y_{TS} \end{pmatrix}, \quad \boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_{11} & & & \\ \vdots & & & \\ \boldsymbol{x}_{1S} & & & \\ & \boldsymbol{x}_{21} & & \\ & \vdots & & \\ & \boldsymbol{x}_{2S} & & \\ & & \boldsymbol{x}_{T1} & \\ & & \vdots & \\ & & \boldsymbol{x}_{TS} & \end{pmatrix},$$

with $\boldsymbol{x}_{ts} = \begin{pmatrix} \boldsymbol{x}_{ts1} \\ \vdots \\ \boldsymbol{x}_{tsV} \end{pmatrix}$, and $\boldsymbol{f} = \begin{pmatrix} \boldsymbol{f}_1 \\ \vdots \\ \boldsymbol{f}_T \end{pmatrix}, \boldsymbol{f}_t = \begin{pmatrix} \boldsymbol{f}_{t1} \\ \vdots \\ \boldsymbol{f}_{tV} \end{pmatrix}$.

The posterior distribution of $\boldsymbol{f}_{tv}$ is proportional to combining the joint likelihood and the prior in Equation (1). Therefore we can update $\boldsymbol{f}_{tv}$ jointly from the following conjugate mulvariate normal distribution:

$$p(\boldsymbol{f}|\cdots) \sim \mathrm{MN}\Big(\big(\boldsymbol{X}(K^{-1} \otimes \mathrm{I}_S)\boldsymbol{X} + \mathrm{I}_T \otimes \Psi\big)^{-1}\boldsymbol{X}'(K^{-1} \otimes \mathrm{I}_S\hat{\boldsymbol{y}}), \big(\boldsymbol{X}(K^{-1} \otimes \mathrm{I}_S)\boldsymbol{X} + \mathrm{I}_T \otimes \Psi\big)^{-1} \Big), \quad (3)$$

Similarly, by combining the joint likelihood as in Equation (2) and the prior $\mathrm{Ga}(a, b)$, β is updated directly through the following Gamma distribution $p(\beta|\cdots)$:

$$\mathrm{Ga}\Big(a + \frac{1}{2}TV, b + \frac{1}{2}(\hat{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{f})'((\Delta + \frac{1}{\sigma^2}\mathrm{I}_T) \otimes \mathrm{I}_S)(\hat{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{f})\Big). \quad (4)$$

In each iteration, given the prior $\mathrm{IG}(c, d)$, σ^2 is drawn through:

$$p(\sigma^2|\cdots) \sim \mathrm{IG}\Big(c + \frac{1}{2}TV, d + \frac{1}{2}\beta(\hat{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{f})'(\hat{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{f})\Big). \quad (5)$$

Since the DP prior implies that D is almost surely discrete, the prior will automatically group the m coefficient-specific hyperparameters $\Psi_{vv'}$ into L clusters Ψ_l^*, where $L \leq \frac{1}{2}V(V-1)$. One of these clusters will most likely correspond to $\Psi_l^* = I_{d_v \times d_v}$, and the other clusters will not be 0. We denote $J_{vv'} = l$ if the (v, v')th covariance matrix is clustered in the lth latent cluster. Our proposed prior can be seen more clearly through the equivalent stick breaking form $J_{vv'} \sum_{l=1}^{\infty} W_l \delta_l$ with

$$\Psi_l^* \sim \begin{cases} \pi\delta_0, & \text{for } l = 1 \\ (1-\pi)\mathrm{IW}(\nu, \Psi_0), & \text{for } l > 1. \end{cases}$$

Extending the exact block Gibbs sampler of [36], the joint prior distributions of $J_{vv'}$ and a latent variable $\zeta_{vv'}$ can be written as

$$f(J_{vv'}, \zeta_{vv'}|\boldsymbol{W}) = \sum_{l: W_l > \zeta_{vv'}} \delta_l(\cdot) = \sum_{l=1}^{\infty} 1(\zeta_{vv'} < W_l)\delta_l(\cdot).$$

We implement the following exact block Gibbs sampler steps:

(1). Sample $\zeta_{vv'} \sim \mathrm{uniform}(0, W_{J_{vv'}})$, for $v \geq v' \geq 1$ with $W_l = \gamma_l \prod_{h<l}(1-\gamma_h)$.

Sample the stick-breaking random variables γ_l from $\gamma_l \sim \text{beta}\Big(1 + m_l, \alpha + \sum_{s=l+1}^{L} m_s\Big)$, for $l = 1, \ldots, L$ with L the minimum value satisfying $W_1 + \ldots + W_L > 1 - \min\{\zeta_{vv'}\}$. m_l is the number of components clustered into the lth cluster.

Sample Ψ_l^* for $l = 1, \ldots, L$ by
(1) For $l = 1$,$\Psi_1^* = 0$.
(2) For $2 \le l \le L$, since $(\boldsymbol{f}_{tv}|\boldsymbol{f}_{t(-v)}) \propto \exp\big\{-\frac{1}{2}\boldsymbol{f}_{tv}{}'\Psi_{vv'}\boldsymbol{f}_{tv} - \boldsymbol{f}_{tv}\sum_{v\sim v'}\Psi_{vv'}\boldsymbol{f}_{tv'}\big\}$, Ψ_l^* can be drawn directly from:

$$\Psi_l^* \sim \text{IW}\big(m_l + \nu + 1, \Psi_0 + \textstyle\sum_t \sum_{J_{vv'}=l} \boldsymbol{f}_{tv}\boldsymbol{f}_{tv'}\big).$$

(2). Sample $J_{vv'}$ for $v \ge v' \ge 1$ from the multinomial conditional with

$$\Pr(J_{vv'} = l|\cdot) \propto 1(\zeta_{vv'} < \pi_l)\prod_t \exp\{-\boldsymbol{f}_{tv}\Psi_l^*\boldsymbol{f}_{tv'}\}.$$

After updating γ_h, with the relationship $\gamma_{vh} \sim \text{beta}(1, \alpha), \alpha \overset{\text{ind}}{\sim} \text{Ga}(1, \alpha_0)$, we sample α through

$$p(\alpha|\cdots) \sim \mathcal{E}\bigg(\alpha_0 - \sum_{v,h}\log(1 - \gamma_{vh})\bigg). \tag{6}$$

where $\mathcal{E}(x;\lambda) = \lambda\exp(-\lambda x)$ is the exponential density.

Based on the above discussion, the proposed *NOBLE* algorithm is summarized in Algorithm 1.

Algorithm 1 *NOBLE* Algorithm

Require: y_{ts}, $\boldsymbol{x}_{tsv}$, IN_{tsv}, K $t = 1, \ldots, T$, $s = 1, \ldots, S$, $v = 1, \ldots, V$
Ensure: the initial value for $\boldsymbol{f}_{tv}$, β, σ^2, $\Psi_{vv'}$ and α
for $i = 1$ to Total number of iterations **do**
  for $t = 1$ to T **do**
    for $v, v' = 1$ to V **do**
      Update $\boldsymbol{f}$ through the multivariate normal distribution in Equation (3);
      Update β through the Gamma distribution as in Equation (4);
      Draw σ^2 directly from Inverse Gamma distribution in Equation (5);
      Update DP related parameters using exact block Gibbs sampler as described in the above Step 4;
      Update α using truncated exponential distribution as in equation (6).
    end for
  end for
end for

5. EXPERIMENTS

In this section, we present some experimental results showing the effectiveness of the proposed *NOBLE* algorithm and compare against the following algorithms[3]:

1. regMVMT [44]: an inductive multi-view learning algorithm for multiple related tasks through a co-regularized framework.
2. SMTL [27]: a Bayesian semi-supervised learning framework for problems with multiple tasks using unlabeled data based on Markov random walk.
3. CASO [10]: a multi-task learning algorithm improving the ASO algorithm [1] through a novel regularizer.

For all 4 algorithms, we repeat the experiments 10 times and report the average classification error[4]. For regMVMT, the parameters are optimized using cross-validation. For SMTL and CASO, the parameters are set according to [27] and [10] respectively. For the proposed *NOBLE* algorithm, we simply set non-informative hyperparameters as $\alpha_0 = 1$, $\pi = 1/2$, $\nu = 2n_p + 1$ and $\Psi_0 = \mathbf{I}_{n_p}$ without prior knowledge about the correlation among the tasks and the relative importance of each view in the predictive model of each task. We also performed convergence diagnostics, such as trace plots and Geweke's convergence diagnostic for randomly selected parameters. No signs of adverse mixing have been found. All results are based on 3,000 Gibbs sampling iterations after a burn-in period of 2,000.

In our experiments, to generate multiple views from the original feature space, we adopt a similar strategy as in [25], and apply different linear/nonlinear dimensionality reduction methods, including ICA with different functions (pow[3] or order 3 polynomial kernel, Tanh, Gaussian, skew) [18], PCA based (PCA, Prob PCA [31], and kernel PCA), MDS, diffusion maps, Laplacian, and Laplacian Eigenmaps [32], resulting in 11 views total.

20 newsgroups data set. We first consider the 20 newsgroups data set [4]. This data set consists of articles from 20 different newsgroups forming a hierarchical structure. Here we focus on the "comp" and "rec" categories (similar experimental results are observed for the other categories and thus omitted for brevity), and create 4 tasks from them. To be specific, for each task, we pick one subcategory from "comp" and "rec" respectively and randomly sample 100 articles from each subcategory to form 2 classes, each described by 53975 features.

To test the capability of our proposed algorithm to recover data sets with different sparsity, we experiment on data sets with various numbers of labeled examples: varying from randomly selecting 20 to 180 observed samples and use the remaining as test set.

Figure 2 shows the comparison results of the 4 algorithms with varying training set size. Each subfigure shows the average classification error for a single task. From these figures, we can see that the performance of *NOBLE* dominates the other methods, and the margin becomes more significant as the number of labeled examples increases. This is because *NOBLE* is able to learn from data: (1) if all the tasks/views are related, and (2) how much they are related to each other.

Figure 3 shows that by using the DP prior, we are able to partition the 11 views into 2 groups roughly: one consists of 7 views generated using ICA and PCA based dimensionality reduction methods, and the other consists of 4 views generated using MDS, diffusion maps, Laplacian, and Laplacian Eigenmaps. In each iteration of the algorithm, there is a positive probability that $\Psi_{V_iV_j} = \Psi_{V_iV_{j'}}$ for every $j \ne j'$. Two views $j \ne j'$ are said to be clustered in terms of sharing covariance matrices if and only if $\Psi_{V_iV_j} = \Psi_{V_iV_{j'}}$ and Figure 3 is the average over the iterations. The clustering

[3]We did not compare with $IteM^2$ [21] since in our experiments, the features are not guaranteed to be non-negative. As shown in [43], the performance of $IteM^2$ is not satisfactory in this case.

[4]For the sake of clarity, we did not display the error bars.

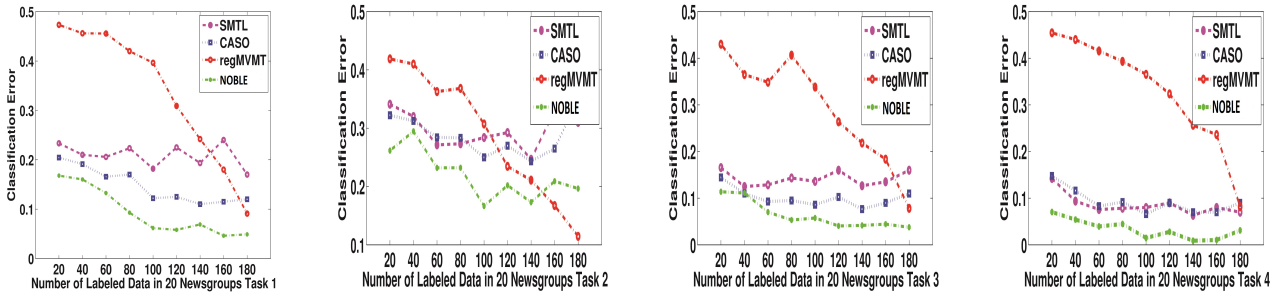

Figure 2: Comparison results on the 20 Newsgroups data.

of the views encoded by the ties among the covariance matrices will simply be referred to as the "clustering of the views", although it should be understood that it is the data themselves that are clustered. The fact that our model induces ties among the views is the means by which it borrows strength across objects for estimation.

WebKB data set. Next we test the performance of *NOBLE* on WebKB data set, where the goal is to classify whether a web page is course related or not [8]. We also create 4 tasks from this data set, each including 200 web pages collected from the same university.

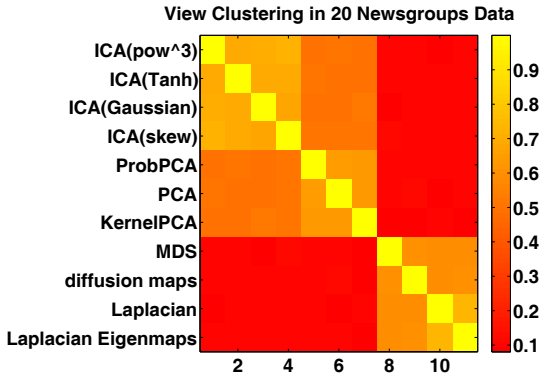

Figure 3: *NOBLE* clustering probability for 11 views of the 20 newsgroups data.

Figure 4 shows the comparison results with varying training set size. Similarly as before, we can see that the performance of *NOBLE* is better than the other 3 competitors in each of the 4 tasks.

Email spam data set. Finally, we compare on the email spam data set from ECML 2006 discovery challenge.[5] The goal is to classify if each email is spam or ham. In problem A, There are 3 users with 2,500 emails each, which are considered as 3 related tasks.

Comparison results are shown in Figure 5. On this data set, we also see improved performance of *NOBLE* over the competitors except for Task 1: when the training set size is small, *NOBLE* and CASO are pretty close to each other; when the training set size is large, the performance of *NOBLE* is consistently improved whereas the performance of CASO fluctuates. We notice that throughout the extensive experiments, regMVMT cannot perform well when the training sample size is small.

We also test the computation time per iteration in *NOBLE* as we vary the training set size, which is shown in Figure 6. From this figure, we can see that *NOBLE* scales linearly with respect to the total number of labeled examples, thus it is scalable to relatively large data sets.

[5] `http://www.ecmlpkdd2006.org/challenge.html`.

6. CONCLUSION

In this paper, we propose a nonparametric Bayes model for addressing problems with dual-heterogeneity, i.e., task heterogeneity (multiple related tasks) and view heterogeneity (multiple views). Compared with state-of-the-art techniques which assume that the tasks are equally related and the views are equally consistent, we aim at answering the following two questions: (1) Are all the tasks equally related and all the views equally consistent? (2) To what extent are the tasks related to each other, and the views consistent with each other? To this end, we make use of the normal penalty with sparse inverse covariances and the matrix DP prior to adaptively learn the task relatedness and the view consistency. Furthermore, we propose the *NOBLE* algorithm based on an efficient Gibbs sampler, which constructs predictors for all the tasks leveraging both the multi-task and multi-view nature. Experimental results on several real data sets show that *NOBLE* outperforms existing methods in M^2TV learning.

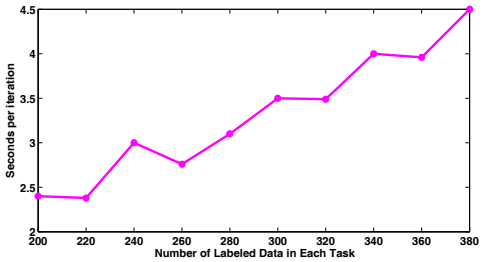

Figure 6: Computation time per iteration of *NOBLE*.

References

[1] R. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

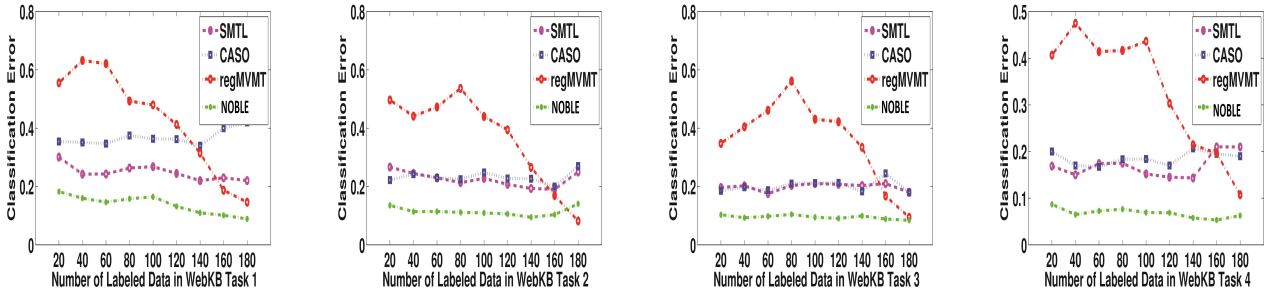

Figure 4: Comparison results on the WebKB data.

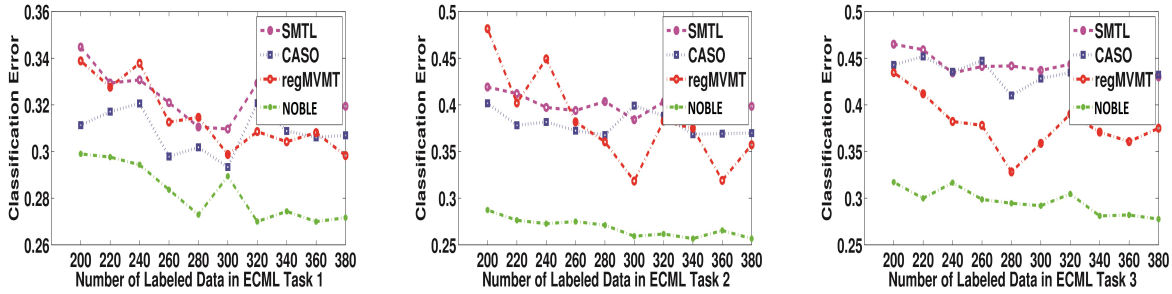

Figure 5: Comparison results on the email spam data.

[2] C. Antoniak. Mixtures of dirichlet processes with applications to bayeisan nonparametric problems. *The Annals of Statistics*, 2:1152–1174, 1974.

[3] C. Archambeau, S. Guo, and O. Zoeter. Sparse bayesian multi-task learning. *NIPS*, 2011.

[4] A. Asuncion and D. Newman. UCI machine learning repository, 2007.

[5] B. Bakker and T. Hesks. Task clustering and gating for bayesian multitask learning. *JMLR*, 4:83–99, 2003.

[6] D. Blackwell and J. MacQueen. Ferguson distributions via polya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.

[7] D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, 2003.

[8] A. Blum and T. M. Mitchell. Combining labeled and unlabeled sata with co-training. In *COLT*, 1998.

[9] D. Burr and H. Doss. A bayesian semiparametric model for random-effects meta-analysis. *Journal of the American Statistical Association*, 100(469):242–251, 2005.

[10] J. Chen, T. Lei, J. Liu, and J. Ye. A convex formulation for learning shared structures from multiple tasks. *ICML*, 2009.

[11] J. Chen, J. Liu, and J. Ye. Learning incoherent sparse and low-rank patterns from multiple tasks. In *KDD*, pages 1179–1188, 2010.

[12] J. Chen, J. Zhou, and J. Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *KDD*, pages 42–50, 2011.

[13] C. Christoudias, R. Urtasun, and T. Darrell. Multi-view learning in the presence of view disagreement. In *UAI*, pages 88–96, 2008.

[14] L. Ding, A. Yilmaz, and R. Yan. Interactive image segmentation using dirichlet process multiple-view learning. *IEEE Transactions on Image Processing*, 21(4):2119–2129, 2012.

[15] D. Dunson, Y. Xue, and L. Carin. The matrix stick-breaking process: flexible bayes meta analysis. *Journal of the American Statistical Association*, 103:317–327, 2008.

[16] J. Farquhar, D. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmak. Two view learning: Svm-2k, theory and practice. *NIPS*, 2005.

[17] T. Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1:209–230, 1973.

[18] H. Gavert, J. Hurri, J. Sarela, and A. Hyvarinen. The fastica package for matlab. `http://research.ics.aalto.fi/ica/fastica/`, 2005.

[19] S. Han, X. Liao, and L. Carin. Cross-domain multitask learning with latent probit models. *NIPS*, 2012.

[20] M. Harel and S. Mannor. Learning from multiple outlooks. In *ICML*, pages 401–408, 2011.

[21] J. He and R. Lawrence. A graphbased framework for multi-task multi-view learning. In *ICML*, pages 25–32, 2011.

[22] J. He, Y. Liu, and Q. Yang. Linking heterogeneous input spaces with pivots for multi-task learning. In *SDM*, 2014.

[23] X. Jin, F. Zhuang, S. Wang, Q. He, and Z. Shi. Shared structure learning for multiple tasks with multiple views. In *ECML/PKDD (2)*, pages 353–368, 2013.

[24] S. M. Kakade and D. P. Foster. Multi-view regression via canonical correlation analysis. In *COLT*, pages 82–96, 2007.

[25] A. Kumar, P. Rai, and H. D. III. Co-regularized multi-view spectral clustering. In *NIPS*, pages 1413–1421, 2011.

[26] Y.-X. Li, S. Ji, S. Kumar, J. Ye, and Z.-H. Zhou. Drosophila gene expression pattern annotation through multi-instance multi-label learning. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 9(1):98–112, 2012.

[27] Q. Liu, X. Liao, and L. Carin. Semi-supervised multi-task leraning. *NIPS*, 2007.

[28] I. Muslea, S. Minton, and C. A. Knoblock. Active + semi-supervised learning = robust multi-view learning. In *ICML*, pages 435–442, 2002.

[29] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *CIKM*, pages 86–93, 2000.

[30] J. O'Sullivan and S. Thrun. Discovering structure in multiple learning tasks: The tc algorithm. *ICML*, pages 489–497, 1996.

[31] M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

[32] L. van der Maaten, E. Postma, and H. van den Herik. Dimensionality reduction: A comparative review. *Tilburg University Technical Report, TiCC-TR 2009-005*, 2009.

[33] H. Wang, F. Nie, H. Huang, S. L. Risacher, A. J. Saykin, and L. Shen. Identifying disease sensitive and quantitative trait-relevant biomarkers from multidimensional heterogeneous imaging genetics data via sparse multimodal multitask learning. *Bioinformatics*, 28(12):127–136, 2012.

[34] W. Wang and Z.-H. Zhou. A new analysis of co-training. In *ICML*, pages 1135–1142, 2010.

[35] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007.

[36] C. Yau, O. Papaspiliopoulos, G. Roberts, and C. Holmes. Nonparametric hidden markov models with application to the analysis of copy-number-variation in mammalian genomes. *Journal of Royal Statistical Society: Series B*, 73(1):37–57, 2010.

[37] K. Yu and W. Chu. Gaussian process models for link analysis and transfer learning. In *NIPS*, 2007.

[38] K. Yu, A. Schwaighofer, and V. Tresp. Learning gaussian processes from multiple tasks. *ICML*, 2005.

[39] K. Yu, A. Schwaighofer, V. Tresp, W. Ma, and H. Zhang. Collaborative ensemble learning: Combining collaborative and content-based information filtering via hierarchical bayes. *UAI*, 2003.

[40] K. Yu, V. Tresp, and S. Yu. A nonparametric hierarchical bayesian framework for information filtering. *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004.

[41] D. Zhang, J. He, and R. D. Lawrence. Mi2ls: multi-instance learning from multiple informationsources. In *KDD*, pages 149–157, 2013.

[42] D. Zhang, J. He, Y. Liu, L. Si, and R. D. Lawrence. Multi-view transfer learning with a large margin approach. In *KDD*, pages 1208–1216, 2011.

[43] J. Zhang and J. Huan. Inductive multi-task learning with multiple view data. In *KDD*, pages 543–551, 2012.

[44] J. Zhang and J. Huan. Inductive multi-task learning with multiple view data. In *KDD*, pages 543–551, 2012.

[45] Y. Zhang and D.-Y. Yeung. A convex formulation for learning task relationships in multi-task learning. *CoRR*, abs/1203.3536, 2012.

[46] J. Zhou, J. Chen, and J. Ye. Clustered multi-task learning via alternating structure optimization. In *NIPS*, pages 702–710, 2011.