Multiplicative Sparse Feature Decomposition for Efficient Multi-View Multi-Task Learning

Lu Sun^1 , **Canh Hao Nguyen**^{1} , **Hiroshi Mamitsuka**^{1,2}

^{1}Bioinformatics Center, Institute for Chemical Research, Kyoto University, Japan
^{2}Department of Computer Science, Aalto University, Finland
{sunlu, canhhao, mami}@kuicr.kyoto-u.ac.jp

Abstract

Multi-view multi-task learning refers to dealing with dual-heterogeneous data, where each sample has multi-view features, and multiple tasks are correlated via common views. Existing methods do not sufficiently address three key challenges: (a) saving task correlation efficiently, (b) building a sparse model and (c) learning view-wise weights. In this paper, we propose a new method to directly handle these challenges based on multiplicative sparse feature decomposition. For (a), the weight matrix is decomposed into two components via low-rank constraint matrix factorization, which saves task correlation by learning a reduced number of model parameters. For (b) and (c), the first component is further decomposed into two subcomponents, to select topic-specific features and learn view-wise importance, respectively. Theoretical analysis reveals its equivalence with a general form of joint regularization, and motivates us to develop a fast optimization algorithm in a linear complexity w.r.t. the data size. Extensive experiments on both simulated and real-world datasets validate its efficiency.

1 Introduction

Multi-View Multi-Task Learning (MVMTL) is an important learning problem with a variety of interesting real-world applications, such as text categorization [He and Lawrence, 2011], bioinformatics analysis [Zhang and Huan, 2012], semantic image annotation [Jin *et al.*, 2013] and web page classification [Lu *et al.*, 2017]. The objective of MVMTL is to improve the prediction of targets by relying on dual-heterogeneity [He and Lawrence, 2011]. Dual-heterogeneity comprises task-heterogeneity and feature-heterogeneity, indicating that multiple tasks possibly related with each other by the common views of samples, and each sample can have various representations from different views. A representative example is classification of protein functions, in which each protein is possibly associated with multiple functional classes, including metabolism, transcription and cellular organization, and has distinct features from multiple views, such as protein sequences and 3D structures.

The problems with single-heterogeneity have been extensively studied in Multi-Task Learning (MTL) and Multi-View Learning (MVL). The intuitive idea behind MTL/MVL is that, learning performance for one single task/view can be improved by leveraging samples from other related tasks/views. MVL captures feature-heterogeneity by saving view consistency in the way that predictive models from multiple views achieve mutual agreement on the unlabeled data [Blum and Mitchell, 1998; Hardoon *et al.*, 2004]. In contrast, MTL saves task-heterogeneity by modeling task correlation and learning multiple correlated tasks together, based on joint regularization [Liu *et al.*, 2009b; Gong *et al.*, 2012a] or feature decomposition [Jalali *et al.*, 2010; Han and Zhang, 2015].

Recent research has gradually shifted its emphasis from the problems with single-heterogeneity to the ones with dual-heterogeneity. Several MVMTL methods have recently been proposed, based on joint regularization [Zhang and Huan, 2012], factorization machines [Lu *et al.*, 2017], and multi-objective optimization [Zhou *et al.*, 2018]. Current MVMTL methods usually suffer from three problems: (a) saving task correlation in the way that models of each task-pair are similar, which is too strong on many real problems, and results in a relatively large number of model parameters to learn; (b) building a learning model on original features, however, irrelevant features would harm the generalization ability, and it is probable that a group of correlated tasks shares group-specific features; (c) modeling strict view consistency, that models from multiple views should be consistent, and assigning a same weight to different views. But it can be expected that multiple views provide supplementary information, and different views contribute distinct importance to the model.

To address all the three problems, in this paper, we propose a novel method via multiplicative **S**parse feature decom**P**osition for mu**L**ti-v**I**ew multi-**T**ask learning (**SPLIT**), which decomposes the weight matrix $\mathbf{\Theta}$ into three multiplicative components $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{H}$. Under an assumption that tasks can be reconstructed by a small number of latent topics, $\mathbf{\Theta}$ is first decomposed as $\mathbf{\Theta} = \mathbf{WH}$ by low-rank constraint matrix factorization, leading to a reduced number of effective parameters of $\mathbf{\Theta}$. To learn a topic-specific sparse model and a view-weighting scheme, $\mathbf{W}$ is further decomposed as $\mathbf{W} = \mathbf{A} \circ \mathbf{B}$ by Hadamard (element-wise) product, where $\mathbf{A}$ and $\mathbf{B}$ select topic-specific relevant features and assign different weights to distinct topic-view pairs, respectively. Fig. 1 illustrates the

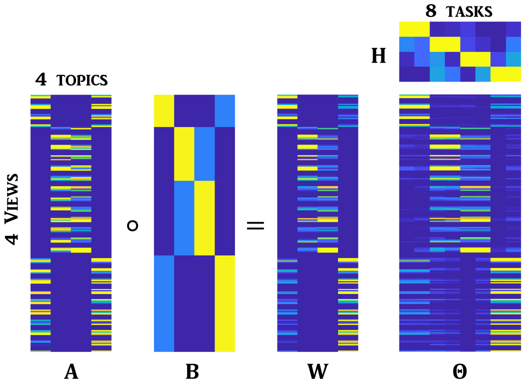

Figure 1: Multiplicative feature decomposition used for generating a simulated dataset with 4 views, 4 topics and 8 tasks. The weight matrix $\mathbf{\Theta}$ is decomposed by $\mathbf{\Theta} = (\mathbf{A} \circ \mathbf{B})\mathbf{H}$. Components $\mathbf{A}$ and $\mathbf{B}$ store topic-specific features and view-wise weights, respectively, while $\mathbf{W} = \mathbf{A} \circ \mathbf{B}$ and $\mathbf{H}$ together model task correlation. Darker blue (yellow) color indicates smaller (larger) values in magnitude.

multiplicative sparse feature decomposition. Our theoretical analysis shows two things: 1) the family of methods is equivalent to a joint regularization with a more general form of regularizers, and 2) feature-sparse component $\mathbf{A}$ is directly related to view-weighting component $\mathbf{B}$, motivating us to derive a fast optimization algorithm in a linear complexity w.r.t. the data size. Promising empirical results on various datasets demonstrate the efficiency of SPLIT for MVMTL. We highlight the contributions of this paper as follows:

- We focus on the key missing points of current MVMTL methods, i.e. feature selectivity and view weighting, and develop a new method that allows us to efficiently select relevant features, and assign view-wise importance. This was not realized by any method so far.
- Theoretical analysis reveals that the equivalence between a general family of the proposed framework and the jointly regularized approach, leading to two novel MVMTL formulations.
- An efficient optimization algorithm is developed to solve the optimization problem in a linear time w.r.t. the data size, which has not been realized by any method so far.

We begin by discussing related works in Section 2. Next, we introduce the proposed SPLIT in Section 3. We provide theoretical analysis of a family of SPLIT in Section 4, and develop an efficient algorithm in Section 5. In Section 6, we present experimental results on evaluation. Finally, we conclude this paper in Section 7.

2 Related works

For *Multi-task learning*, based on the assumption that task relationship is shared through task-common features, regularization with $\ell_{p,q}$-norm ($p > 1, q \geq 1$) is imposed on the weight matrix to encourage group sparsity [Liu *et al.*, 2009a; Gong *et al.*, 2012a]. To capture both task commonality and specificity, feature decomposition approaches are proposed via decomposing the weight matrix into multiple components by summation [Chen *et al.*, 2011; Gong *et al.*, 2012b] or multiplication [Lozano and Swirszcz, 2012; Wang *et al.*, 2016].

Multi-view learning aims to utilize features from different views to improve the performance of a baseline learner. Two types of methods, co-training [Blum and Mitchell, 1998; Muslea *et al.*, 2006; Sun and Jin, 2011] and co-regularization [Sindhwani *et al.*, 2005; Xie and Sun, 2015; Kan *et al.*, 2016], are proposed to make predictive models from multiple views consistent on the unlabeled data.

For *multi-view multi-task learning*, IteM^2 [He and Lawrence, 2011] constructs a bi-partite graph for each view, and projects any two tasks to a new reproducing kernel Hilbert space based on their common views. As an inductive learning method, CSL-MTMV [Jin *et al.*, 2013] aims to mine a low-dimensional subspace shared among related tasks with common views. Based on multilinear factorization machines, MFM was recently proposed in [Lu *et al.*, 2017] by learning both task-specific feature map and task-view shared multilinear structures. In [Li and Huan, 2018], asymmetric bilinear factor analyzers with rank constraints are applied to capture the interactions among tasks and views.

3 The proposed method

3.1 Preliminary

For a MVMTL problem with V views and T tasks, given the data $\mathbf{X}_t = [\mathbf{X}_t^1, ..., \mathbf{X}_t^V] \in \mathbb{R}^{n_t \times d}$ of t-th task, where $\mathbf{X}_t^v \in \mathbb{R}^{n_t \times d_v}$ denotes the v-th data with $d = \sum_v d_v$, we introduce a linear model to approximate the t-th target $\mathbf{y}_t \in \mathbb{R}^{n_t}$:

$$\mathbf{y}_t \approx \frac{1}{V}\sum_{v=1}^{V} \mathbf{X}_t^v \boldsymbol{\theta}_t^v = \frac{1}{V}\mathbf{X}_t \boldsymbol{\theta}_t, \quad \forall t. \tag{1}$$

In (1), $\boldsymbol{\theta}_t = [\boldsymbol{\theta}_t^1; ...; \boldsymbol{\theta}_t^V] \in \mathbb{R}^d$ is the parameter model of the t-th task, with $\boldsymbol{\theta}_t^v \in \mathbb{R}^{d_v}$ being its sub-vector of the v-th view. For T tasks, we have $\mathbf{\Theta} = [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_T] \in \mathbb{R}^{d \times T}$. For convenience, we omit the intercept in (1) by assuming that the input data and target have been centered in column-wise.

3.2 Methodology

In order to model task correlation efficiently, we assume that multiple tasks are correlated through a subspace constructed by a limited number of latent topics. For instance, in image annotation, the tags (tasks) “ocean” and “sky” can be treated as a single latent topic, since they typically emerge together and share similar color features. Thus, instead of directly learning $\mathbf{\Theta} \in \mathbb{R}^{d \times T}$, we impose a low-rank constraint on it, and restrict $\mathbf{\Theta}$ to learn only a reduced number of effective parameters, which is much less than $d \times T$. To this end, matrix factorization is applied to decompose $\mathbf{\Theta}$ by $\mathbf{\Theta} = \mathbf{WH}$, where $\mathbf{W} \in \mathbb{R}^{d \times K}$ and $\mathbf{H} \in \mathbb{R}^{K \times T}$ with K denoting the number of latent topics, $K \leq T$. Each column of $\mathbf{W}$ corresponds to one latent topic, and $\mathbf{H}$ is usually considered as a factor loading matrix, whose t-th column $\mathbf{h}_t$ stores the coefficients of K latent topics for the t-th task. In this sense, the model parameter $\boldsymbol{\theta}_t$ belonging to the t-th task enables to be reconstructed by $\boldsymbol{\theta}_t = \mathbf{W}\mathbf{h}_t$. This low-rank constraint matrix decomposition not only captures task correlations, but also controls overfitting by reducing the model size.

In terms of multi-view sparse learning, it is possible that a small feature subset from one view is crucial to detect a certain topic, even though that view is not discriminative for the topic. In other words, we expect that each topic is supported by a specific subset of features, while different views contributes distinct importance. A real example is image annotation, where features from edge view are critical to detect a latent topic comprising tags (tasks) "car", "bus" and "truck", and almost irrelevant for the topic consisting of tags (tasks) "ocean" and "sky". Thus, to learn a topic-specific sparse model and a view-weighting scheme, for the v-th view of the k-th column (topic) of $\mathbf{W}$, we decompose it by

$$\mathbf{w}_k^v = \beta_k^v \boldsymbol{\alpha}_k^v = \boldsymbol{\alpha}_k^v \circ \beta_k^v \mathbf{1}_{d_v} = \boldsymbol{\alpha}_k^v \circ \boldsymbol{\beta}_k^v, \tag{2}$$

where $\boldsymbol{\alpha}_k^v \in \mathbb{R}^{d_v}$ aims at selecting topic-specific features for the v-th view, $\beta_k^v \geq 0$ denotes the topic-specific weight of the v-th view, and $\mathbf{1}_{d_v}$ is a all-one vector in size of d_v. Therefore, the topic model $\mathbf{W}$ can be reformulated by

$$\mathbf{W} = \mathbf{A} \circ \left(\begin{bmatrix} \mathbf{1}_{d_1} & & \\ & \ddots & \\ & & \mathbf{1}_{d_V} \end{bmatrix} \mathcal{B} \right) = \mathbf{A} \circ \mathbf{B}, \tag{3}$$

where $\mathcal{B} \in \mathbb{R}^{V \times K}$ with β_k^v being the element in the v-th row of the k-th column, $\forall t, v$. In (3), $\mathbf{A} \in \mathbb{R}^{d \times K}$ and $\mathbf{B} \in \mathbb{R}^{d \times K}$ store the topic-specific sparse model and the view-weighting scheme, respectively, with the k-th column being $\boldsymbol{\alpha}_k = [\boldsymbol{\alpha}_k^1; ...; \boldsymbol{\alpha}_k^V]$ and $\boldsymbol{\beta}_k = [\boldsymbol{\beta}_k^1; ...; \boldsymbol{\beta}_k^V]$, respectively. We then use ℓ_1-norm and Frobenius norm to regularize $\mathbf{A}$ and $\mathcal{B}$, respectively, with $\|\mathbf{A}\|_{1,1} = \sum_{ij} |\alpha_j^i|$ and $\|\mathcal{B}\|_{\mathrm{F}}^2 = \sum_{ij} |\beta_j^i|^2$. In this way, topic-specific irrelevant features will be discarded by $\mathbf{A}$, while non-discriminative views will be assigned with small weights by $\mathcal{B}$.

Inspired by above motivations, we decompose the model parameter $\boldsymbol{\Theta}$ into a product of three component, i.e., $\boldsymbol{\Theta} = \mathbf{W}\mathbf{H} = (\mathbf{A} \circ \mathbf{B})\mathbf{H}$, and propose the multiplicative multi-task multi-view sparse feature learning model as follows:

$$\min_{\boldsymbol{\Theta}} \sum_{t=1}^{T} L(\mathbf{y}_t, \frac{1}{V}\mathbf{X}_t \boldsymbol{\theta}_t) + \lambda_1 \|\mathbf{A}\|_{1,1} + \lambda_2 \|\mathcal{B}\|_{\mathrm{F}}^2 + \eta \|\mathbf{H}\|_{\mathrm{F}}^2,$$
$$\text{s.t. } \boldsymbol{\Theta} = (\mathbf{A} \circ \mathbf{B})\mathbf{H}, \quad \mathbf{B} \geq 0. \tag{4}$$

For the v-th view of the t-th task ($\forall t, v$), we have

$$\boldsymbol{\theta}_t^v = \mathbf{W}^v \mathbf{h}_t = (\mathbf{A}^v \circ \mathbf{B}^v)\mathbf{h}_t = \sum_{k=1}^{K} h_t^k (\boldsymbol{\alpha}_k^v \circ \boldsymbol{\beta}_k^v), \tag{5}$$

where h_t^k is the coefficient of the k-th topic for the t-th task.

4 Theoretical analysis

In this section, we present theoretical analysis of a general form of (4), which helps us to develop an efficient algorithm for a family of SPLIT. This general form is formualted by

$$\min_{\substack{\boldsymbol{\Theta}=(\mathbf{A}\circ\mathcal{B})\mathbf{H}, \\ \mathbf{B}\geq 0}} \sum_{t=1}^{T} L(\mathbf{y}_t, \frac{1}{V}\mathbf{X}_t \boldsymbol{\theta}_t) + \lambda_1 \sum_{k=1}^{K} \sum_{v=1}^{V} \|\boldsymbol{\alpha}_k^v\|_p^p$$
$$+ \lambda_2 \sum_{k=1}^{K} \sum_{v=1}^{V} |\beta_k^v|^q + \eta \|\mathbf{H}\|_{\mathrm{F}}^2, \tag{6}$$

where $\|\cdot\|_p$ denotes ℓ_p-norm with $\|\boldsymbol{\alpha}\|_p = \sqrt[p]{\sum_j |\alpha_j|^p}$. Obviously, (6) becomes (4) with $p = 1$ and $q = 2$. The following theorem[1] shows that, under some conditions, (6) is equivalent to a jointly regularized model.

Table 1: A summary of conclusions of Theorem 1 with $p, q \in \{1, 2\}$. Here $R(\mathbf{W})$ is the regularization term w.r.t. $\mathbf{W}$ in (7).

(p, q)	(1, 1)	(1, 2)	(2, 1)	(2, 2)
γ	$2\sqrt{\lambda_1\lambda_2}$	$\gamma = 2\lambda_1^{\frac{2}{3}}\lambda_2^{\frac{1}{3}}$	$2\lambda_1^{\frac{1}{3}}\lambda_2^{\frac{2}{3}}$	$\gamma = 2\sqrt{\lambda_1\lambda_2}$
$R(\mathbf{W})$	$\sum_{k,v}\sqrt{\|\mathbf{w}_k^v\|_1}$	$\sum_{k,v}\sqrt[3]{\|\mathbf{w}_k^v\|_1^2}$	$\sum_{k,v}\sqrt[3]{\|\mathbf{w}_k^v\|_2^2}$	$\sum_{k,v}\|\mathbf{w}_k^v\|_2$
β_k^v	$\lambda_1\lambda_2^{-1}\|\boldsymbol{\alpha}_k^v\|_1$	$\sqrt{\lambda_1\lambda_2^{-1}\|\boldsymbol{\alpha}_k^v\|_1}$	$\lambda_1\lambda_2^{-1}\|\boldsymbol{\alpha}_k^v\|_2^2$	$\sqrt{\lambda_1\lambda_2^{-1}\|\boldsymbol{\alpha}_k^v\|_2^2}$

Theorem 1. *Let* $(\hat{\mathbf{W}}, \hat{\mathbf{H}})$ *be the optimal solution of the following optimization problem,*

$$\min_{\boldsymbol{\Theta}=\mathbf{W}\mathbf{H}} \sum_{t=1}^{T} L(\mathbf{y}_t, \frac{1}{V}\mathbf{X}_t \boldsymbol{\theta}_t) + \gamma \sum_{k=1}^{K} \sum_{v=1}^{V} \sqrt[2s]{\|\mathbf{w}_k^v\|_p^p} + \eta \|\mathbf{H}\|_F^2, \tag{7}$$

where $\mathbf{w}_k^v$ *is the* v*-th view sub-vector of the* k*-th column of* $\mathbf{W}$. *If* $\{\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{H}}\}$ *is the optimal solution of (6), we have* $\hat{\mathbf{W}} = \hat{\mathbf{A}} \circ \hat{\mathbf{B}}$, *given* $\gamma = 2\sqrt{\lambda_1^{2-\frac{p}{qs}} \lambda_2^{\frac{p}{qs}}}$ *and* $s = \frac{p+q}{2q}$. *In addition, the view-weighting component* $\hat{\mathbf{B}}$ *is related with the topic-specific sparse component* $\hat{\mathbf{A}}$ *by the following formula*

$$\beta_k^v = \sqrt[q]{\lambda_1 \lambda_2^{-1} \|\boldsymbol{\alpha}_k^v\|_p^p}, \quad \forall v, k. \tag{8}$$

Theorem 1 reveals the family in (6) is equivalent to a jointly regularized problem in (7) with a more general form of regulariers, while Eq. (8) implies the sparsity of the topic-specific component $\mathbf{A}$ is relative to the sparsity of the view-weighting component $\mathbf{B}$. A direct instantiation of Theorem 1 with $p = 1$ and $q = 2$ is exactly related with (4). Table 1 summarizes the conclusions derived from Theorem 1 with $p, q \in \{1, 2\}$.

5 Optimization algorithm

Despite of the equivalence of (4) and (7), here we aim to solve (4). It is because optimizing $\mathbf{A}$ and $\mathbf{B}$ separately is much easier than optimizing $\mathbf{W}$ directly, and $\mathbf{B}$ gives some insight into the relationship between views and topics. The objective function of (4) is convex w.r.t. $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{H}$, respectively, which motivates us to develop an alternating algorithm. For simplicity, least squared loss is considered here. The algorithm repeats following three steps until convergence.

(i) Update $\mathbf{A}$ with fixed $\mathbf{B}$ and $\mathbf{H}$:

$$\min_{\mathbf{A}} \sum_{t=1}^{T} \left\| \mathbf{y}_t - \frac{1}{V}\mathbf{X}_t (\mathbf{A} \circ \mathbf{B})\mathbf{h}_t \right\|^2 + \lambda_1 \|\mathbf{A}\|_{1,1}. \tag{9}$$

It is a lasso-like problem, which can be solved by many efficient gradient descent optimization approaches. Let

[1] Proofs are provided in the supplement: https://www.dropbox.com/s/vwl1jtt0qiso8j8/SPLIT_sup.pdf?dl=0.

$\nabla f(\mathbf{A})$ denote the derivative of the loss function in (9) w.r.t $\mathbf{A}$, whose k-th column equals to

$$\nabla f(\mathbf{A})|_k = \frac{2}{V}\boldsymbol{\beta}_k \circ \sum_{t=1}^{T} h_t^k \mathbf{X}_t^\top \left(\frac{1}{V}\mathbf{X}_t(\mathbf{A}\circ\mathbf{B})\mathbf{h}_t - \mathbf{y}_t \right). \tag{10}$$

Then, proximal gradient descent method can be applied to update $\mathbf{A}$ according to

$$\mathbf{A}^* \leftarrow soft(\mathbf{A} - \mu\nabla f(\mathbf{A}), \lambda_1), \tag{11}$$

where $soft(a,b) = sign(a)\max(|a| - b, 0)$ is the soft thresholding operator, and μ is the learning rate determined by line search.

(ii) Update $\mathbf{B}$ with fixed $\mathbf{A}$ and $\mathbf{H}$: According to Theorem 1 and Table 1, $\mathbf{B}$ is solved with a closed-form solution.

(iii) Update $\mathbf{H}$ with fixed $\mathbf{W} = \mathbf{A}\circ\mathbf{B}$:

$$\min_{\mathbf{H}} \sum_{t=1}^{T} \left\| \mathbf{y}_t - \frac{1}{V}\mathbf{X}_t\mathbf{W}\mathbf{h}_t \right\|^2 + \eta \left\|\mathbf{H}\right\|_{\mathrm{F}}^2. \tag{12}$$

Let $\tilde{\mathbf{X}}_t = \mathbf{X}_t\mathbf{W}$, above problem has a closed-form solution, whose t-th column is calculated by

$$\mathbf{h}_t = (\frac{1}{V}\tilde{\mathbf{X}}_t^\top\tilde{\mathbf{X}}_t + \eta V\mathbf{I}_K)^{-1}\tilde{\mathbf{X}}_t^\top\mathbf{y}_t. \tag{13}$$

Since it typically follows $K \leq T \ll d$, the closed-form solution (13) can be efficiently computed.

Proposition 1. *The proposed iterative optimization algorithm does not increase the objective function of (4) at each iteration, indicating that*

$$J(\mathbf{A}^{(i+1)}, \mathbf{B}^{(i+1)}, \mathbf{H}^{(i+1)}) \leq J(\mathbf{A}^{(i)}, \mathbf{B}^{(i)}, \mathbf{H}^{(i)}), \tag{14}$$

in the $(i+1)$-th iteration, with $J(\mathbf{A}, \mathbf{B}, \mathbf{H})$ denoting the objective function of (4) w.r.t. $\mathbf{A}$, $\mathbf{B}$ *and* $\mathbf{H}$.

Proposition 1 guarantees that the optimization algorithm does not increase the objective value of (4) in each iteration. In practice, accelerated proximal method [Nesterov, 2013] is applied to accelerate the algorithm. In terms of time complexity analysis, updating $\mathbf{A}$ comprises two major steps, gradient computation and soft thresholding, resulting in time complexities of $\mathcal{O}(d(N+TK))$ and $\mathcal{O}(dK)$, respectively, with $N = \sum_t n_t$ being the total number of samples from multiple tasks. Computation of $\mathbf{B}$ needs a time complexity of $\mathcal{O}(dK)$, and updating $\mathbf{H}$ with the closed-form solution has a time cost of $\mathbf{O}(K^2(N+TK))$. Therefore, the total time complexity of each iteration is $\mathcal{O}(d(N+TK) + K^2(N+TK))$, which is linear in number of samples, features and tasks.

6 Experiments

6.1 Datasets and comparing methods

Four real-world datasets are used for performance evaluation, and their statistics are summarized in Table 2. We compare SPLIT with five methods for performance evaluation. Ridge regression [Hoerl and Kennard, 1970] and Lasso [Tibshirani, 1996] are selected as baseline methods. As representative MTL methods with multiplicative feature learning, MLL [Lozano and Swirszcz, 2012] and MMTFL [Wang *et al.*, 2016] enable to model various patterns of task-specific/task-common features, and outperform several MTL methods on real-world applications. As a state-of-the-art MVMTL method, MFM [Lu *et al.*, 2017] is selected due to its superior performance compared with existing MVMTL methods. The proposed SPLIT[2] is implemented in two variants: SPLIT_1 and SPLIT_2, corresponding to $p = 1$ and $p = 2$, respectively, both with $q = 2$ in (6). The difference between them is that, for an irrelevant feature, SPLIT_1 might shrink corresponding element in $\mathbf{A}$ to zero, and SPLIT_2 might shrink it to a small non-zero value, instead.

Table 2: The statistics of used real-world datasets, where V and T denote the number of views and tasks, respectively, d_v is the number of features in the v-th view ($d = \sum_v d_v$), and n_{tp}/n_{tn} is the number of positive/negative samples in the t-th task ($n_t = n_{tp} + n_{tn}$).

Datasets	V	T	d_v	d	n_t		URL
					n_{tp}	n_{tn}	
Mirflickr	2	8	43∼150	193	668∼7,849	3,413∼10,594	URL1
Caltech101	6	4	40∼1,984	3,766	123∼798	1,588∼2,263	URL2
NUS-Object	5	7	64∼225	634	964∼2,370	8,000∼9,406	URL3
NUS-Scene	5	15	63∼224	629	1,039∼11,995	4,409∼15,365	URL3

URL1: https://press.liacs.nl/mirflickr/
URL2: http://www.vision.caltech.edu/Image_Datasets/Caltech101/
URL3: http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

6.2 Configuration

In evaluation, for each task we randomly select $a\%$, 20% and 20% of its total samples as training set, validation set and testing set, respectively, with $a \in \{10, 20, 30\}$. We repeat this procedure five times, and report the mean value and standard deviation of two metrics, Area Under ROC-Curve (AUC) and Accuracy. Grid search is conducted on the evaluation set, and the best parameter setting is used for prediction on the testing set. For grid search, values of regularization coefficients of comparing methods are selected from $\{10^a \mid |a| \in \{0,1,2,3,4\}\}$. The dimensionality of latent space in MFM is set as 20, as recommended in [Lu *et al.*, 2017]. The number K of latent topics of SPLIT is set according to $\frac{K}{T} \in \{0.3, 0.5, 0.7, 0.9\}$. For each iterative algorithm, we terminate it once the relative change of its objective is below 10^{-5}, and set the maximum number of iterations as 1000.

6.3 Experiments on simulated datasets

Simulated datasets

For simulated datasets, we set the number of tasks and views are set as $T = 8$ and $V = 4$, respectively, and select the dimensionality d_v of the v-th view from $\{25, 42, 58, 75\}$ with $d = \sum_v d_v = 200$. Each task has the same number ($n = 200$) of labeled samples. The weight matrix $\mathbf{\Theta}$ is decomposed into three parts in a multiplicative way, i.e., $\mathbf{\Theta} = (\mathbf{A}\circ\mathbf{B})\mathbf{H}$, with the latent dimensionality being $K = 4$. Elements of $\mathbf{A} \in \mathbb{R}^{d\times K}$ are randomly sampled according to normal distribution $\mathcal{N}(0, 16)$, while elements of $\mathcal{B} \in \mathbb{R}^{V\times T}$ and $\mathbf{H} \in \mathbb{R}^{K\times T}$ are randomly sampled based on uniform distributions $\mathcal{U}(0,1)$ and $\mathcal{U}(-1,1)$, respectively. To make $\mathbf{A}$ and

[2] We provide the MATLAB code of SPLIT at: https://www.dropbox.com/s/ej7joxq6nv2yoto/SPLIT.zip?dl=0

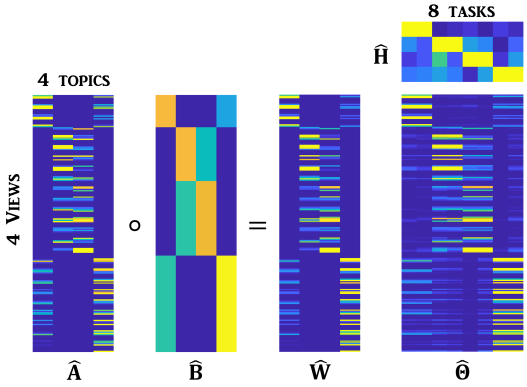

Figure 2: Illustration of multiplicative feature decomposition by SPLIT_1 on simulated data with designed model $\mathbf{\Theta}$ shown in Fig. 1. The model $\widehat{\mathbf{\Theta}} = (\widehat{\mathbf{A}} \circ \widehat{\mathbf{B}})\widehat{\mathbf{H}}$ is learned by SPLIT_1. Darker yellow (blue) color indicates larger (smaller) values in magnitude.

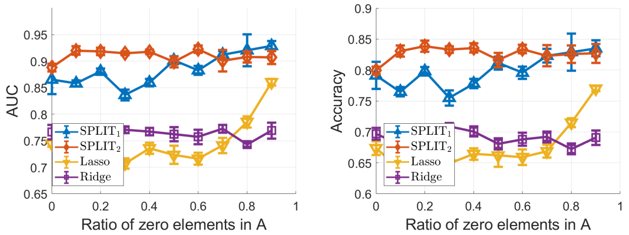

Figure 3: Comparison of SPLIT_1, SPLIT_2 and two baseline methods on ten simulated datasets, which are generated by varying the percentage of zero-elements in $\mathbf{A}$ from 0% to 90% by step 10%.

$\mathcal{B}$ sparse, $a\%$ and $b\%$ of total elements are assigned with value 0, respectively. Finally, for the t-th task, its target vector $\mathbf{y}_t$ is calculated by $\mathbf{y}_t = \frac{1}{V}\mathbf{X}_t\boldsymbol{\theta}_t + \boldsymbol{\delta}_t$, where $\mathbf{X}_t$ is randomly sampled from normal distribution $\mathcal{N}(0, 25)$, and $\boldsymbol{\delta}_t \sim \mathcal{N}(0, 1)$ denotes the stochastic noise.

Case study on feature decomposition and sparse learning

We illustrate multiplicative feature decomposition of SPLIT on one designed synthetic dataset in Fig 2[3], where $\widehat{\mathbf{\Theta}} = (\widehat{\mathbf{A}} \circ \widehat{\mathbf{B}})\widehat{\mathbf{H}}$ is learned by SPLIT with the setting $\lambda_1 = 10^1$, $\lambda_2 = 10^3$ and $\eta = 10^4$. The designed model $\mathbf{\Theta} = (\mathbf{A} \circ \mathbf{B})\mathbf{H}$ is shown in Fig. 1. As shown in Fig. 1 and Fig. 2, SPLIT_1 successfully detects the underlying models by selecting topic-specific features in $\mathbf{A}$, learning view-wise weights in $\mathbf{B}$, and saving task correlation in $\mathbf{W} = \mathbf{A} \circ \mathbf{B}$ and $\mathbf{H}$.

Next, to evaluate the performance of SPLIT on handling the datasets with sparse underlying models, we conduct an experiment on ten synthetic datasets by varying the percentage of zero-elements (irrelevant features) in $\mathbf{A}$ from 0% to 90% by step 10%. Fig. 3 shows the comparison result of SPLIT_1, SPLIT_2 and two baseline methods, Lasso and Ridge, in AUC and Accuracy. As the sparsity of $\mathbf{A}$ increases, SPLIT_1 and SPLIT_2 consistently outperform two baselines with a significant advantage. The sparse ratio 0.7 in $\mathbf{A}$ seems to be a turning point for sparse methods (SPLIT_1 and Lasso) and dense methods (SPLIT_2 and Ridge). Sparse methods perform worse than their dense counterparts when the ratio is below 0.7, but outperform them once the ratio exceeds 0.7. It shows that sparse learning is necessary for MVMTL, when the underlying model is indeed sparse.

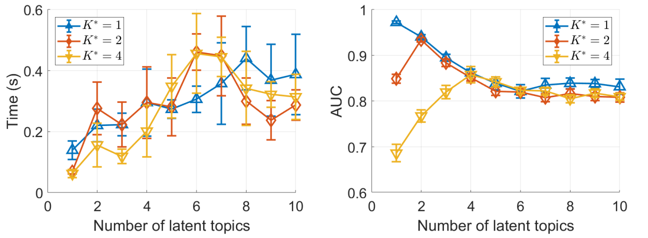

Figure 4: Performance of SPLIT_1 on three simulated datasets by varying its number K of latent topics from 1 to 8 by step 1. The datasets are generated by changing the number K^* of latent semantics in truth model $\mathbf{\Theta}^*$ according to $K^* \in \{1, 2, 4\}$.

Analysis on task correlation modeling

To model task correlation, SPLIT constructs multiple tasks based on a limited number of latent topics. We expect that it would avoid overfitting, and give computational efficiency. To evaluate this effect, we generate three simulated datasets by changing the number K^* of latent topics with $K^* \in \{1, 2, 4\}$, and apply SPLIT_1 on each dataset by varying its number K of latent topics from 1 to 8 by step 1. As shown in Fig. 4, as the value of K increases, SPLIT consumes more running time, because it needs to learn a larger number of effective parameters. In terms of AUC, as the value of K^* increases, SPLIT always achieves the best performance when $K = K^*$. Therefore, once tasks are modeled by a small number of latent topics in one dataset, SPLIT has a chance to improve the performance in less running time.

6.4 Experiments on real-world datasets

Evaluation of comparing methods

For evaluation of comparing methods, we conduct an experiment on four multi-view multi-task datasets, and report empirical results in AUC and Accuracy by Table 3[4]. In the experiment, we change the ratio n/N of training samples from 10% to 30% by step 10%. In Table 3, the best performance is highlighted in boldface. For all methods, we can see that as the ratio n/N increases, the performance in both AUC and Accuracy increases as well. Specifically, the two variants of SPLIT, SPLIT_1 and SPLIT_2, together perform the best in 20 cases out of total 24 cases. Such observation validates the effectiveness of SPLIT on handling MVMTL problems via multiplicative sparse feature decomposition. Its ability on selecting relevant features, weighting different views and saving task correlation, leads to a more powerful MVMTL learner. As multiplicative multi-task learning methods, MLL and MMTFL perform the second best, and compete with SPLIT_1 and SPLIT_2 on the Caltech101 and NUS-Object datasets. It is probably because

[3] A quantitative comparison is presented in the supplement.

[4] Statistical test on the results is provided in the supplement.

Table 3: Experimental results on four real-world datasets by selecting the percentage n/N of labeled data from $\{10\%, 20\%, 30\%\}$.

		AUC							**Accuracy**						
Dataset	n/N	Ridge	Lasso	MLL	MMTFL	MFM	SPLIT$_1$	SPLIT$_2$	Ridge	Lasso	MLL	MMTFL	MFM	SPLIT$_1$	SPLIT$_2$
Mirflickr	10%	.615±.004	.623±.000	.637±.002	.635±.002	.618±.003	.638±.004	**.639±.004**	.775±.002	.777±.001	.788±.001	.787±.001	.766±.004	.791±.001	**.792±.002**
	20%	.630±.003	.632±.003	.652±.003	.651±.003	.615±.002	**.656±.002**	.655±.002	.780±.001	.780±.000	.792±.001	.792±.001	.767±.004	.794±.002	**.795±.001**
	30%	.650±.002	.652±.002	.663±.002	.664±.002	.620±.006	**.667±.002**	.666±.002	.784±.001	.785±.001	.794±.001	.794±.001	.770±.003	**.798±.001**	.797±.001
Caltech101	10%	.986±.002	.990±.000	**.998±.000**	.997±.000	.978±.003	.991±.002	.990±.001	.929±.004	**.983±.001**	.978±.003	**.983±.001**	.969±.005	.981±.003	.982±.003
	20%	.990±.000	.990±.001	.997±.000	.997±.001	.987±.004	**.999±.000**	.992±.002	.981±.001	.986±.000	.981±.001	.983±.002	.974±.004	**.991±.001**	.989±.002
	30%	.990±.000	.990±.000	**.999±.000**	**.999±.000**	.990±.001	**.999±.000**	.992±.001	.986±.000	.987±.001	**.993±.001**	.990±.001	.984±.003	.991±.003	.990±.001
NUS-Object	10%	.843±.004	.833±.002	.841±.006	.841±.013	.836±.002	**.845±.004**	.839±.004	.863±.003	.859±.001	.876±.001	**.878±.005**	.857±.001	.869±.003	.866±.001
	20%	.851±.002	.863±.001	.869±.001	.862±.004	.848±.001	**.870±.001**	.866±.001	.876±.001	.881±.001	**.888±.002**	.885±.003	.866±.002	**.888±.002**	.885±.002
	30%	.860±.002	.867±.003	**.877±.002**	.874±.002	.856±.003	**.877±.002**	.874±.001	.880±.001	.882±.001	**.894±.001**	.892±.001	.871±.001	**.894±.001**	.892±.001
NUS-Scene	10%	.744±.006	.734±.012	.748±.003	.753±.003	.744±.005	.780±.001	**.782±.002**	.835±.001	.832±.012	.842±.001	.847±.000	.820±.003	**.864±.001**	**.864±.001**
	20%	.747±.001	.743±.001	.760±.002	.772±.002	.745±.001	.791±.001	**.792±.001**	.840±.001	.842±.001	.858±.001	.861±.001	.840±.006	**.865±.001**	**.865±.000**
	30%	.767±.001	.771±.001	.775±.004	.793±.001	.767±.005	**.800±.001**	**.800±.001**	.845±.001	.846±.001	.863±.000	.866±.001	.844±.008	.867±.001	**.868±.001**

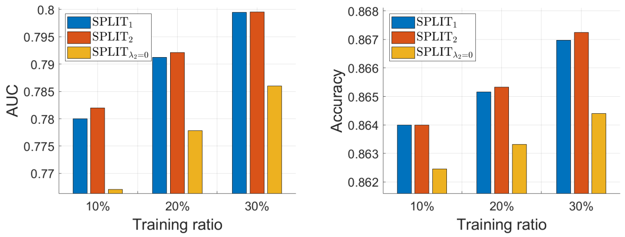

Figure 5: Analysis on the view-weighting effect of SPLIT on the NUS-Scene dataset. In this experiment, the training ratio is selected from $\{10\%, 20\%, 30\%\}$. SPLIT$_{\lambda_2=0}$ is a degenerated variant of SPLIT$_1$ by assigning a same weight to all views.

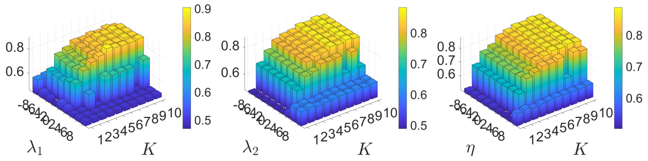

Figure 6: Sensitivity analysis of λ_1, λ_2, η and K on the NUS-Object dataset. Values (shown in the logarithmic scale) of λ_1, λ_2 and η are selected from $\{10^a \mid |a| \in \{0, 2, 4, 6, 8\}\}$, while the value of K is varied from 1 to 10 by step 1.

both of the methods enable to select task-specific features and encourage task correlations via common features. In terms of two baseline methods, Lasso outperforms Ridge regression in almost all the cases, indicating the importance on learning a sparse model by discarding useless features for MVMTL.

Analysis on the view-weighting effect

To demonstrate the effectiveness of view-weighting component $\mathbf{B}$ used in SPLIT, an experiment is performed by comparing SPLIT$_1$ and SPLIT$_2$ with a special variant SPLIT$_{\lambda_2=0}$, which removes the view-weighting component $\mathbf{B}$ and decomposes $\mathbf{\Theta}$ by $\mathbf{\Theta} = \mathbf{AH}$. The training ratio is selected from $\{10\%, 20\%, 30\%\}$, and comparison results are shown in Fig. 5. From Fig. 5, we can see that both SPLIT$_1$ and SPLIT$_2$ significantly outperform SPLIT$_{\lambda_2=0}$. It shows that it is important to learn view-wise weights for performance improvement on the NUS-Scene dataset.

Sensitivity analysis on hyperparameters

To understand the behavior of SPLIT, sensitivity analysis on hyperparameters is conducted on NUS-Object with the training ratio being 30%. SPLIT has three regularization parameters λ_1, λ_2 and η, controlling topic-specific sparsity, view-wise importance and regression weight, respectively, and one parameter K on the latent dimensionality, controlling the strongness of task correlation. Values of λ_1, λ_2 and η are selected from $\{10^a \mid |a| \in \{0, 2, 4, 6, 8\}\}$, while the value of K is varied from 1 to 10 by step 1. Three experiments are conducted to evaluate the pairwise correlation between the parameters. The first experiment on λ_1 and K is conducted by fixing $\lambda_2 = \eta = 1$, and similar setting is applied for other two experiments. Fig. 6 leads to three conclusions: (1) as the value of K increases, the performance first increases, and then becomes stable once $K \geq 5$; (2) compared to λ_2 and η, the performance is more sensitive to the value change of λ_1; (3) the best performance on NUS-Object is achieved by setting $\lambda_1 \leq 10^2$, and $\lambda_2, \eta \leq 10^4$.

7 Conclusion

In this paper, we propose a novel method, SPLIT, via multiplicative sparse feature decomposition, so as to address three challenges in MVMTL: saving task correlations efficiently, selecting relevant features and assigning view-wise weights. Our theoretical analysis provides an equivalence guarantee of SPLIT with a general form of joint regularization, according to which two formulations with specific settings are proposed and solved by an efficient optimization algorithm in a linear complexity w.r.t. the data size. Extensive experiments on a variety of datasets show that it is necessary for a successful MVMTL method to model task correlation, select relevant features and learn view-wise weights, and demonstrate that the proposed SPLIT enables to address the challenges in both simulated and real-world MVMTL applications.

Acknowledgements

L. S. has been supported in part by JST ACCEL (grant number JPMJAC1503). C. H. N. has been supported in part by MEXT Kakenhi 18K11434. H. M. has been supported in part by JST ACCEL (grant number JPMJAC1503), MEXT Kakenhi (grant numbers 16H02868 and 19H04169), FiDiPro by Tekes (currently Business Finland) and AIPSE program by Academy of Finland.

References

[Blum and Mitchell, 1998] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[Chen *et al.*, 2011] Jianhui Chen, Jiayu Zhou, and Jieping Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 42–50. ACM, 2011.

[Gong *et al.*, 2012a] Pinghua Gong, Jieping Ye, and Changshui Zhang. Multi-stage multi-task feature learning. In *Advances in neural information processing systems*, pages 1988–1996, 2012.

[Gong *et al.*, 2012b] Pinghua Gong, Jieping Ye, and Changshui Zhang. Robust multi-task feature learning. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 895–903. ACM, 2012.

[Han and Zhang, 2015] Lei Han and Yu Zhang. Learning multi-level task groups in multi-task learning. In *AAAI*, volume 15, pages 2638–2644, 2015.

[Hardoon *et al.*, 2004] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[He and Lawrence, 2011] Jingrui He and Rick Lawrence. A graph-based framework for multi-task multi-view learning. *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pages 25–32, 2011.

[Hoerl and Kennard, 1970] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[Jalali *et al.*, 2010] Ali Jalali, Sujay Sanghavi, Chao Ruan, and Pradeep K Ravikumar. A dirty model for multi-task learning. In *Advances in neural information processing systems*, pages 964–972, 2010.

[Jin *et al.*, 2013] Xin Jin, Fuzhen Zhuang, Shuhui Wang, Qing He, and Zhongzhi Shi. Shared Structure Learning for Multiple Tasks with Multiple Views. In *Machine Learning and Knowledge Discovery in Databases*, pages 353–368, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[Kan *et al.*, 2016] Meina Kan, Shiguang Shan, Haihong Zhang, Shihong Lao, and Xilin Chen. Multi-view discriminant analysis. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):188–194, 2016.

[Li and Huan, 2018] Xiaoli Li and Jun Huan. Interactions modeling in multi-task multi-view learning with consistent task diversity. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 853–861. ACM, 2018.

[Liu *et al.*, 2009a] Han Liu, Mark Palatucci, and Jian Zhang. Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 649–656. ACM, 2009.

[Liu *et al.*, 2009b] Jun Liu, Shuiwang Ji, and Jieping Ye. Multi-task feature learning via efficient l2, 1-norm minimization. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 339–348. AUAI Press, 2009.

[Lozano and Swirszcz, 2012] Aurelie C Lozano and Grzegorz Swirszcz. Multi-level lasso for sparse multi-task regression. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 595–602. Omnipress, 2012.

[Lu *et al.*, 2017] Chun-Ta Lu, Lifang He, Weixiang Shao, Bokai Cao, and Philip S. Yu. Multilinear Factorization Machines for Multi-Task Multi-View Learning. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining - WSDM '17*, pages 701–709, 2017.

[Muslea *et al.*, 2006] Ion Muslea, Steven Minton, and Craig A Knoblock. Active learning with multiple views. *Journal of Artificial Intelligence Research*, 27:203–233, 2006.

[Nesterov, 2013] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[Sindhwani *et al.*, 2005] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *Proceedings of ICML workshop on learning with multiple views*, volume 2005, pages 74–79. Citeseer, 2005.

[Sun and Jin, 2011] Shiliang Sun and Feng Jin. Robust co-training. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(07):1113–1126, 2011.

[Tibshirani, 1996] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[Wang *et al.*, 2016] Xin Wang, Jinbo Bi, Shipeng Yu, Jiangwen Sun, and Minghu Song. Multiplicative multitask feature learning. *Journal of Machine Learning Research*, 17(80):1–33, 2016.

[Xie and Sun, 2015] Xijiong Xie and Shiliang Sun. Multi-view twin support vector machines. *Intelligent Data Analysis*, 19(4):701–712, 2015.

[Zhang and Huan, 2012] Jintao Zhang and Jun Huan. Inductive multi-task learning with multiple view data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*, page 543, 2012.

[Zhou *et al.*, 2018] D. Zhou, J. Wang, B. Jiang, H. Guo, and Y. Li. Multi-task multi-view learning based on cooperative multi-objective optimization. *IEEE Access*, 6:19465–19477, 2018.