Variable Selection and Task Grouping for Multi-Task Learning

Jun-Yong Jeong
POSTECH
Pohang, South Korea
june0227@postech.ac.kr

Chi-Hyuck Jun
POSTECH
Pohang, South Korea
chjun@postech.ac.kr

ABSTRACT

We consider multi-task learning, which simultaneously learns related prediction tasks, to improve generalization performance. We factorize a coefficient matrix as the product of two matrices based on a low-rank assumption. These matrices have sparsities to simultaneously perform variable selection and learn and overlapping group structure among the tasks. The resulting bi-convex objective function is minimized by alternating optimization, where sub-problems are solved using alternating direction method of multipliers and accelerated proximal gradient descent. Moreover, we provide the performance bound of the proposed method. The effectiveness of the proposed method is validated for both synthetic and real-world datasets.

CCS CONCEPTS

• **Computing methodologies → Multi-task learning**; • **Information systems → Data mining**;

KEYWORDS

Multi-task learning; Low-rank; Sparse representation; k-support norm

ACM Reference Format:
Jun-Yong Jeong and Chi-Hyuck Jun. 2018. Variable Selection and Task Grouping for Multi-Task Learning. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3219992

1 INTRODUCTION

Multi-task learning (MTL) refers to simultaneously learning multiple related prediction tasks rather than learning each task independently [8, 39]. Simultaneous learning enables us to share common information among related tasks, and works as an inductive bias to improve generalization performance. MTL is based on the premise the fact that humans can learn a new task easily when they already have knowledge from similar tasks.

Re-vised The major challenges in MTL are how to share common information among related tasks and how to prevent unrelated tasks from being sharing. Previous studies achieved this by performing variable selection [16, 38], assuming a structure among tasks [11, 15, 23, 41], or imposing a low-rank constraint [1, 2, 20].

The variable selection approach selects a subset of variables for related tasks [26, 38]. Traditional studies are based on a strict assumption that selected variables are shared among all tasks [26, 33]. Recent studies have suggested a more flexible approach that involves selecting variables by decomposing a coefficient into a shared part and an individual part [13, 16] or factorizing a coefficient using a variable specific part and a task-variable part [38]. Although the variable selection approach provides better interpretability than the other approaches, it has limited ability to share common information among related tasks.

The structure approach assumes that related tasks formulate a certain structure. A group structure is most frequently used, in which tasks in a same group have similar coefficients. Initial studies simultaneous learn a disjoint group structure and maximize similarities among the coefficients of tasks within each group [15, 40]. Later studies improve them to learn an overlapping group structure [27, 41]. The special case of overlapping group structure is a tree structure. At first, the tree structure should be known priori [19], later it can be learned during optimization [11]. A graph structure is also exploited to represent asymmetric task relatedness [23].

The low-rank approach assumes that coefficient vectors lie within a low-dimensional latent space [1, 2] and is a representation learning that transforms input variables into low-dimensional features and learns coefficient vectors in the feature space [28]. The low-rank approach has also been widely studied in multi-output regression, where entire tasks have real-valued outputs and share the same training set [9]. It can be achieved by imposing a trace-constraint [2], encouraging sparsity on the singular values of a coefficient matrix [12, 29, 35, 36], or factorizing a coefficient matrix as the product of a variable-latent matrix and a latent-task matrix [1, 2, 17, 20, 28]. Several studies have shown that the low-rank approach is equivalent to the group structure approach [29, 40]. Thus, recent studies on the low-rank approach have focused on improving the ability of models to learn group structures among tasks [4, 18, 20]. The low-rank approach provides a flexible way to share common information among related tasks and reduces the effective number of parameters.

It attempts to combine the variable selection approach and the learning of group structures among tasks, especially those based on the low-rank approach. This combination learns sparse representations to provide better interpretability and shares common information among related tasks in a group to improve generalization performance. Previous studies have either partially achieved this goal or have limitations. For example, Chen and Huang [9] factorized a coefficient matrix and imposed sparsity between the rows of a variable-latent matrix to perform variable selection. They

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219992

solved multi-output regression and did not explicitly learn a group structure among tasks. Kumar and Daumé III [20] also factorized a coefficient matrix and imposed sparsity within the column vectors of a latent-task matrix to learn overlapping group structures among tasks, but they did not perform variable selection. Richard et al. [35, 36] penalized both a trace norm and an ℓ_1 norm to simultaneously perform variable selection and impose a low-rank structure. However, a trace norm penalty requires the use of extensive assumptions to ensure a low-rank structure [30] and singular value decomposition for each iteration of the optimization. Han and Zhang [11] learned overlapping group structures among tasks by decomposing a coefficient matrix into component matrices, but they could not remove irrelevant variables. Wang et al. [38] factorized a coefficient matrix as the product of full-rank matrices to perform variable selection, but did not explicitly learn a group structure among tasks.

This paper proposes the variable selection and task grouping-MTL (**VSTG-MTL**) approach, which simultaneously performs variable selection and learns an overlapping group structure among tasks based on the low-rank approach. Our main ideas are to express a coefficient matrix as the product of a variable-latent matrix and a latent-task matrix and impose sparsities on these matrices. The sparsities between and within the rows of a variable-latent matrix help the model to select relevant variables and have flexibility. We also encourage sparsity within the columns of a latent-task matrix to learn an overlapping group structure among tasks, and note that learning the latent-task matrix is equivalent to learning task coefficient vectors in a low-dimensional feature space where features can be highly correlated. This correlation is considered in the model by applying a k-support norm [29]. The resulting bi-convex problem is minimized by alternating optimization, where sub-problems are solved by applying the alternating direction method of multipliers (ADMM) and accelerated proximal gradient descent. We provide an upper bound on the excess risk of the proposed method to guarantee its performance. Experiments conducted on four synthetic datasets and five real-world datasets show that the proposed VSTG-MTL approach outperforms several benchmark MTL methods and that the k-support norm is effective on handling the possible correlation.

We summarize our contributions as follows

- To the best our knowledge, this is the first work that simultaneously performs variable selection and learns an overlapping group structure among tasks using the low-rank approach.
- We focus on the possible correlation from a representation learning and apply a k support norm to improve generalization performance.
- We present an upper bound on the excess risk of the proposed method.

2 PRELIMINARY

In this section, we explain multi-task learning, low-rank structures, and k-support norms.

2.1 Low-rank Structure for Multi-task Learning

Suppose that we are given D variables and T supervised learning tasks, where the j-th task has an input matrix $\mathbf{X}_j = \left[\left(\mathbf{x}_j^1\right)^T, \ldots, \left(\mathbf{x}_j^{N_j}\right)^T \right]^T \in \mathbb{R}^{N_j \times D}$ with $\mathbf{x}_j^n \in \mathbb{R}^D$ and an output vector $\mathbf{y}_j = \left[y_j^1, \ldots, y_j^{N_j} \right]^T \in \mathbb{R}^{N_j}$. Next, we focus on a linear relation between input and output

$$y_j^n = f(\mathbf{w}_j^T \mathbf{x}_j^n), \tag{1}$$

where f is an identity function for a regression problem $y_j^n \in \mathbb{R}$ or a logit function for a binary classification problem $y_j^n \in \{-1, 1\}$ and $\mathbf{w}_j \in \mathbb{R}^D$ represents a coefficient vector for the j-th task. Then, we can describe the matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_T] \in \mathbb{R}^{D \times T}$ as a coefficient matrix. We then impose a low-rank structure on the coefficient matrix $\mathbf{W}$ to share common information among related tasks [1, 20]. The low-rank structure assumes that the coefficient vectors $\mathbf{w}_j$, $j = 1, \ldots, T$ lie within a low-dimensional latent space and are expressed by linear combinations of latent bases. The coefficient matrix $\mathbf{W}$ can be factorized as the product of two low rank matrices $\mathbf{UV}$, where $\mathbf{U} \in \mathbb{R}^{D \times M}$ is the variable-latent matrix, $\mathbf{V} \in \mathbb{R}^{M \times T}$ is the latent-task matrix, and $M << \min\{D, T\}$ is the number of latent bases. Then, we can express the coefficient of the i-th variable for the j-th task w_{ij} and the coefficient vector for the j-th task $\mathbf{w}_j$ as follows

$$w_{ij} = \mathbf{u}^i \mathbf{v}_j \tag{2}$$

$$\mathbf{w}_j = \mathbf{U}\mathbf{v}_j = \sum_{m=1}^{M} v_{mj} \mathbf{u}_m, \tag{3}$$

where $\mathbf{u}^i \in \mathbb{R}^{1 \times M}$ and $\mathbf{u}_m \in \mathbb{R}^D$ are the i-th row vector and m-th column vector of the variable-latent matrix $\mathbf{U}$, respectively, and $\mathbf{v}_j \in \mathbb{R}^M$ is the j-th column vector of the latent-task matrix $\mathbf{V}$. The above equations reveal the roles of the two matrices. The i-th row vector $\mathbf{u}_i$ and the m-th column vector $\mathbf{u}^m$ can be regarded as being of equal importance of that of the i-th variable and m-th latent basis. Then, the j-th column vector $\mathbf{v}_j$ can be regarded as the weighting vector for the j-th task.

Furthermore, this low-rank structure can be considered as a representation learning [9, 28]. We can rewrite Eq. (1) as

$$y_j^n = f\left(\mathbf{w}_j^T \mathbf{x}_j^n\right) = f\left(\mathbf{v}_j^T (\mathbf{U}^T \mathbf{x}_j^n)\right). \tag{4}$$

The transpose of the variable-latent matrix $\mathbf{U}^T$ and the j-th weighting vector $\mathbf{v}_j$ represent a linear map from a variable space to a feature space, where $\mathbf{x} \in \mathbb{R}^D$ is mapped to $[\mathbf{u}_1^T \mathbf{x}, \ldots, \mathbf{u}_M^T \mathbf{x}]^T \in \mathbb{R}^M$ and the coefficient vector of the j-th task on the feature space, respectively. We note that unless the latent bases $\mathbf{u}_m$, $m = 1, \ldots, M$ are orthogonal, the features $\mathbf{u}_m^T \mathbf{x}$, $m = 1, \ldots, M$ can be highly correlated.

2.2 The k-support Norm

We commonly use an ℓ_1 norm as a convex approximation to an ℓ_0 norm in regularized regression. When features are correlated and form several groups, the ℓ_1 norm penalty tends to select a few features from the groups, where we can improve the generalization performance by selecting all correlated features [3]. In this case, a possible alternative to the ℓ_1 norm is a k-support norm $\| \cdot \|_k^{sp}$, i.e., the tightest convex relaxation of sparsity within a Euclidean ball [3]. The k-support norm is defined for each $\mathbf{w} \in \mathbb{R}^M$ as follows:

$$\|\mathbf{w}\|_k^{sp} := \min \left\{ \sum_{g \in \mathcal{G}_k} \|\mathbf{s}_g\| : \text{supp}(\mathbf{s}_g) \subseteq g, \sum_{g \in \mathcal{G}_k} \mathbf{s}_g = \mathbf{w} \right\},$$

where $\mathcal{G}_k$ denotes all subsets of $1, \dots, M$ of cardinality of at most k. Moreover, $\|\mathbf{w}\|_1^{sp} = \|\mathbf{w}\|_1$ and $\|\mathbf{w}\|_M^{sp} = \|\mathbf{w}\|_2$. Thus, the k-support norm is a trade-off between an ℓ_1 norm and an ℓ_2 norm. This property can be enhanced by inspecting the following proposition:

Proposition 2.1. (*Proposition 2.1* [3]) *For every* $\mathbf{w} \in \mathbb{R}^M$,

$$\|\mathbf{w}\|_k^{sp} = \left(\sum_{l=1}^{k-p-1} \left(|w|_l^{\downarrow} \right)^2 + \frac{1}{p+1} \left(\sum_{l=k-p}^{M} |w|_l^{\downarrow} \right)^2 \right)^{\frac{1}{2}},$$

where $w_l^{\downarrow}$ *is the l-th largest element of the absolute values of* $\mathbf{w}$, *letting* $|w|_0^{\downarrow}$ *denote* $+\infty$, *and* p *is the unique integer in* $\{0, \dots, k-1\}$ *satisfying*

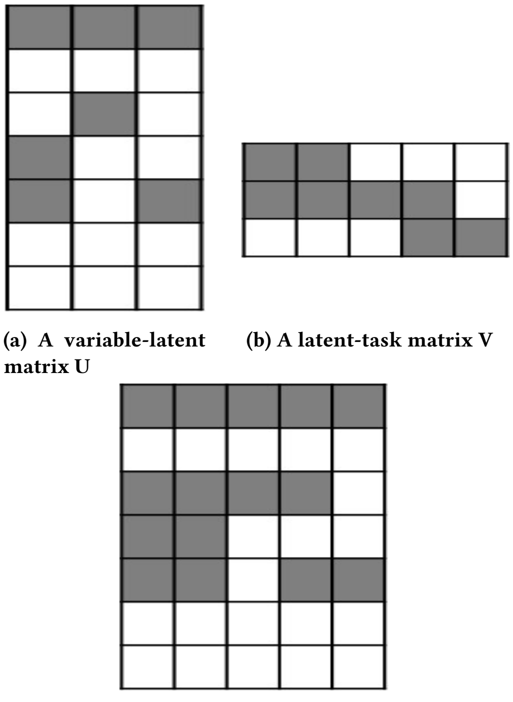
$$|w|_{k-p-1}^{\downarrow} > \frac{1}{p+1} \sum_{l=k-p}^{M} |w|_l^{\downarrow} >= |w|_{k-p}^{\downarrow}.$$

The above proposition shows that the k-support norm imposes both the uniform shrinkage of an ℓ_2 norm on the largest components and the spare shrinkage of an ℓ_1 norm on the smallest components. Thus, in a similar way to Elastic net [42], the k-support norm penalty encourages the selection of a few groups of correlated features and imposes the uniform shrinkage of the ℓ_2 norm on the selected groups.

3 FORMULATION

We aim to simultaneously learn an overlapping group structure among tasks and select relevant variables. To achieve these goals, we employ the low-rank assumption shown in Section 3.1 and impose sparsities on a variable-latent matrix $\mathbf{U}$ and a latent-task matrix $\mathbf{V}$. Fig. 1 shows an example of VSTG-ML, where the gray and white entries express non-zero and zero values, respectively. Each row of the variable-latent matrix $\mathbf{U}$ and the coefficient matrix $\mathbf{W}$ represent a variable. Similarly, each column of the latent-task matrix $\mathbf{V}$ and the coefficient matrix $\mathbf{W}$ represent a task; each column of the variable-latent matrix $\mathbf{U}$ and row of the latent-task matrix $\mathbf{V}$ represent a latent basis or feature. The variable-latent matrix $\mathbf{U}$ in Fig. 1(a) shows the sparsities between and within its rows, while the latent-task matrix $\mathbf{V}$ in Fig. 1(b) shows the sparsity within its columns. The coefficient matrix $\mathbf{W}$ in Fig. 1(c) expresses the product of these matrices.

The sparsity between the variable importance vectors $\mathbf{u}^i$, $i = 1, \dots, D$ induces a model that can be used to select relevant variables [9]. If the i-th variable importance vector $\mathbf{u}^i$ is set to $\mathbf{0}$, then the corresponding variable is removed from the model in accordance with Eq. (2). For example, in Fig. 1(a), the 2nd, 6th and 7th variables are excluded from the model, whereas the 1st, 3rd, 4th, and 5th variables are selected. Simultaneously, the sparsity within the variable importance vector $\mathbf{u}^i$ improves the flexibility of the model. The latent basis vector $\mathbf{u}^m$ does not necessarily depend on all selected variables. Instead, it can have non-zero values from a subset of the selected variables.


(a) A variable-latent matrix U

(b) A latent-task matrix V

(c) A coefficient matrix W

Figure 1: Example of VSTG-MTL. The gray and white entries express non-zero and zero values, respectively. The feature-latent matrix U shows the sparsities between and within its rows representing variables, and the latent-task matrix V shows the sparsity within its columns representing tasks. The columns of the variable-latent matrix U and the rows of the latent-task matrix V represent latent bases or features

The sparsities within the weighting vectors $\mathbf{v}_j, j = 1, \dots, T$ learn an overlapping group structure among tasks [20]. The group structure among tasks are decided by the sparsity patterns on the weighting vector $\mathbf{v}_j$. Tasks with same sparsity patterns on the weighting vector $\mathbf{v}_j$ belong to the same group, whereas those with the orthogonal ones belong to disjoint groups. Two groups are regard as being overlapped if their sparsity patterns are not orthogonal, i.e., they partially share the latent bases. For example, in Fig. 1(b), the 1st and 2nd tasks belong to the same group and share the 2nd latent basis with the 3rd and 4th tasks. However, they do not share any latent basis with the 5th task. As mentioned in Sec 2.1, learning the j-th weighting vectors $\mathbf{v}_j$ is equivalent to learning the coefficient vector of the j-th task in a feature space induced by the transpose of the variable-latent matrix $\mathbf{U}^T$. The features $\mathbf{u}_m^T\mathbf{x}$, $m = 1, \dots, M$ can be highly correlated unless the latent bases are orthogonal. Thus, instead of the ℓ_1 norm, the k-support norm is appropriate to encouraging the sparsity within the weighting vector $\mathbf{v}_j$. The k-support norm induces the less sparse weighting vector $\mathbf{v}_j$ than that from the ℓ_1 norm and similarly enhances the overlaps in the task groups.

We formulate the following problem

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V}} & \sum_{j=1}^{T} \frac{1}{N_j} L\left(\mathbf{y}_j, \mathbf{X}_j \mathbf{U} \mathbf{v}_j\right) \\ \text{s.t } & \|\mathbf{U}\|_1 \le \alpha_1, \quad \|\mathbf{U}\|_{1,\infty} \le \alpha_2, \\ & \sum_{j=1}^{T} \left(\|\mathbf{v}_j\|_k^{sp}\right)^2 \le \beta \end{aligned} \tag{5}$$

where $L(\cdot,\cdot)$ is the empirical loss function, which becomes a squared loss $\frac{1}{2}\|\mathbf{y}_j - \mathbf{X}_j\mathbf{U}\mathbf{v}_j\|_2^2$ for a regression problem and a logistic loss $\sum_{j=1}^{N_j} \log\left(1 + \exp\left(-y_j^n \mathbf{v}_j^T \mathbf{U}^T \mathbf{x}_j^n\right)\right)$ for a binary classification problem; $\|\mathbf{U}\|_1 = \sum_{i=1}^{D}\sum_{m=1}^{M} |u_{im}|$ is the ℓ_1 norm; $\|\mathbf{U}\|_{1,\infty} = \sum_{i=1}^{D} \|\mathbf{u}^i\|_\infty$ is the $\ell_{1,\infty}$ norm; $\|\mathbf{v}_j\|_k^{sp}$ is the k-support norm; and α_1, α_2, and β are the constraint parameters. The $\ell_{1,\infty}$ norm and the ℓ_1 norm constraints encourage the sparsities between and within the variable importance vectors $\mathbf{u}^i$, $i = 1, \ldots, D$. The squared k-support norm constraint encourages the sparsity within the weighting vectors $\mathbf{v}_j$, $j = 1, \ldots, T$ while considering possible correlations among the features.

4 OPTIMIZATION

The optimization problem (5) is bi-convex for the variable-latent matrix $\mathbf{U}$ and latent-task matrix $\mathbf{V}$; for a given $\mathbf{U}$, it is convex for $\mathbf{V}$ and vice versa. We transform the above constraint problem to the following regularized objective function

$$\sum_{j=1}^{T} \frac{1}{N_j} L\left(\mathbf{y}_j, \mathbf{X}_j \mathbf{U} \mathbf{v}_j\right) + \gamma_1 \|\mathbf{U}\|_1 + \gamma_2 \|\mathbf{U}\|_{1,\infty} + \mu \sum_{j=1}^{T} \left(\|\mathbf{v}_j\|_k^{sp}\right)^2, \tag{6}$$

where γ_1, γ_2, and μ are the regularization parameters. Then, we apply alternating optimization to obtain the partial minimum of the objective function (6).

Initial estimates of the matrices $\mathbf{U}$ and $\mathbf{V}$ are crucial in generalization performance considering that the optimization function (6) is non-convex. To compute reasonable initial estimates, for each task, we learn a ridge regression or logistic regression coefficient:

$$\mathbf{w}_j^{init} := \underset{\mathbf{w}}{\text{argmin}} \frac{1}{N_j} L\left(\mathbf{y}_j, \mathbf{X}_j \mathbf{w}\right) + \left(\sqrt{\gamma_1^2 + \gamma_2^2 + \mu^2}\right) \|\mathbf{w}\|_2^2. \tag{7}$$

We also define an initial coefficient matrix that stacks the ridge coefficient as a column vector:

$$\mathbf{W}^{init} = [\mathbf{w}_1^{init}, \ldots, \mathbf{w}_T^{init}]. \tag{8}$$

Then, we compute the top-M left-singular vectors $\mathbf{P} \in \mathbb{R}^{D\times M}$, the top-$M$ right singular vectors $\mathbf{Q} \in \mathbb{R}^{T\times M}$, and the top-$M$ singular value matrix $\Sigma \in \mathbb{R}^{M\times M}$ of the initial coefficient matrix $\mathbf{W}^{init}$. The initial estimates $\mathbf{U}^{init}$ and $\mathbf{V}^{init}$ are given by $\mathbf{P}\Sigma^{\frac{1}{2}}$ and $\Sigma^{\frac{1}{2}}\mathbf{Q}^T$, respectively.

4.1 Updating U

For a fixed latent-task matrix $\mathbf{V}$, the objective function for the variable-latent matrix $\mathbf{U}$ becomes as follows:

$$\sum_{j=1}^{T} \frac{1}{N_j} L\left(\mathbf{y}_j, \mathbf{X}_j \mathbf{U} \mathbf{v}_j\right) + \gamma_1 \|\mathbf{U}\|_1 + \gamma_2 \|\mathbf{U}\|_{1,\infty}. \tag{9}$$

It is solved by applying ADMM [7]. First, we introduce auxiliary variables $\mathbf{Z}_h \in \mathbb{R}^{D\times M}$, $h = 1, 2, 3$ and reformulate the above problem as follows:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{Z}_1,\mathbf{Z}_2,\mathbf{Z}_3} & \sum_{j=1}^{T} \frac{1}{N_j} L\left(\mathbf{y}_j, \mathbf{X}_j \mathbf{Z}_1 \mathbf{v}_j\right) + \gamma_1 \|\mathbf{Z}_2\|_1 + \gamma_2 \|\mathbf{Z}_3\|_{1,\infty} \\ & \text{s.t } \mathbf{A}\mathbf{U} + \mathbf{B}\mathbf{Z} = \mathbf{0}, \end{aligned} \tag{10}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_D \\ \mathbf{I}_D \\ \mathbf{I}_D \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -\mathbf{I}_D & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_D & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I}_D \end{bmatrix}, \quad \text{and} \quad \mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \\ \mathbf{Z}_3 \end{bmatrix}.$$

Let Λ_h be a scaled Lagrangian multiplier for the h-th auxiliary variables $\mathbf{Z}_h$ and $\Lambda = \left[\Lambda_1^T, \Lambda_2^T, \Lambda_3^T\right]^T$. Then, the variable-latent matrix $\mathbf{U}$ is updated as follows:

$$\begin{aligned} \mathbf{U}^{t+1} & := \underset{\mathbf{U}}{\text{argmin}} \frac{\rho}{2} \|\mathbf{A}\mathbf{U} + \mathbf{B}\mathbf{Z}^t + \Lambda^t\|_F^2 \\ & = \underset{\mathbf{U}}{\text{argmin}} \|\mathbf{U} - \mathbf{Z}_1^t + \Lambda_1^t\|_F^2 + \|\mathbf{U} - \mathbf{Z}_2^t + \Lambda_2^t\|_F^2 \\ & + \|\mathbf{U} - \mathbf{Z}_3^t + \Lambda_3^t\|_F^2 \\ & = \frac{1}{3} \sum_{h=1}^{3} \left(\mathbf{Z}_h^t - \Lambda_h^t\right), \end{aligned} \tag{11}$$

where t denotes the iteration and $\rho > 0$ is the ADMM parameter. The auxiliary variables $\mathbf{Z}_h$, $h = 1, 2, 3$ are updated by solving the following problem

$$\begin{aligned} \mathbf{Z}^{t+1} := \underset{\mathbf{Z}}{\text{argmin}} & \sum_{j=1}^{T} \frac{1}{N_j} L\left(\mathbf{y}_j, \mathbf{X}_j \mathbf{Z}_1 \mathbf{v}_j\right) + \gamma_1 \|\mathbf{Z}_2\|_1 + \gamma_2 \|\mathbf{Z}_3\|_{1,\infty} \\ & + \frac{\rho}{2} \|\mathbf{A}\mathbf{U}^{t+1} + \mathbf{B}\mathbf{Z} + \Lambda^t\|_F^2. \end{aligned} \tag{12}$$

In detail, the first auxiliary variable $\mathbf{Z}_1$ is updated as follows

$$\mathbf{Z}_1^{t+1} := \underset{\mathbf{Z}_1}{\text{argmin}} \sum_{j=1}^{T} \frac{1}{N_j} L\left(\mathbf{y}_j, \mathbf{X}_j \mathbf{Z}_1 \mathbf{v}_j\right) + \frac{\rho}{2} \|\mathbf{U}^{t+1} - \mathbf{Z}_1 + \Lambda_1^t\|_F^2. \tag{13}$$

For regression problems with a squared loss, we can compute the close-form updating equation by equating the gradient of the optimization problem (13) to zero as follows:

$$\begin{aligned} \sum_{j=1}^{T} \frac{1}{N_j} \mathbf{X}_j^T \mathbf{X}_j \mathbf{Z}_1 \mathbf{v}_j \mathbf{v}_j^T + \rho \mathbf{Z}_1 & = \sum_{j=1}^{T} \frac{1}{N_j} \mathbf{X}_j^T \mathbf{y}_j \mathbf{v}_j^T + \rho \left(\mathbf{U}^{t+1} + \Lambda^{\mathrm{t}}\right) \\ \left[\sum_{j=1}^{T} \frac{1}{N_j} \mathbf{v}_j \mathbf{v}_j^T \otimes \mathbf{X}_j^T \mathbf{X}_j^T + \rho \mathbf{I}\right] & \text{vec}(\mathbf{Z}_1) \\ = \sum_{j=1}^{T} \frac{1}{N_j} \text{vec}\left(\mathbf{X}_j^T \mathbf{y}_j \mathbf{v}_j^T\right) & + \rho \text{vec}\left(\mathbf{U}^{t+1} + \Lambda_1^t\right), \end{aligned}$$

where vec$(\cdot)$ is the vectorization operator and $\otimes$ is the Kronecker product. The above linear system of equations is solved by using the Cholesky or LU decomposition.
For binary classification problems with a logistic loss, it is solved by using L-BFGS [32], where the gradient is given as follows:

$$\nabla_{\mathbf{Z}_1} = \sum_{j=1}^{T}\sum_{n=1}^{N_j} \frac{-y_j^n \mathbf{x}_j^n \mathbf{v}_j^T}{1+\exp\left(y_j^n \mathbf{v}_j^T \mathbf{Z}_1^T \mathbf{x}_j^n\right)} + \rho\left(\mathbf{Z}_1 - \mathbf{U}^{t+1} - \mathbf{\Lambda}_1^t\right).$$

The other auxiliary variables $\mathbf{Z}_2$ and $\mathbf{Z}_3$ are updated as follows:

$$\begin{aligned}\mathbf{Z}_2^{t+1} &:= \underset{\mathbf{Z}_2}{\operatorname{argmin}}\, \gamma_1 \|\mathbf{Z}_2\|_1 + \frac{\rho}{2}\|\mathbf{U}^{t+1} - \mathbf{Z}_2 + \mathbf{\Lambda}_2^t\|_F^2 \\ &= \operatorname{prox}_{\frac{\gamma_1}{\rho}\|\cdot\|_1}\left(\mathbf{U}^{t+1} + \mathbf{\Lambda}_2^t\right),\end{aligned} \tag{14}$$

$$\begin{aligned}\mathbf{Z}_3^{t+1} &:= \underset{\mathbf{Z}_3}{\operatorname{argmin}}\, \gamma_2 \|\mathbf{Z}_3\|_{1,\infty} + \frac{\rho}{2}\|\mathbf{U}^{t+1} - \mathbf{Z}_3 + \mathbf{\Lambda}_3^t\|_F^2 \\ &= \operatorname{prox}_{\frac{\gamma_2}{\rho}\|\cdot\|_{1,\infty}}\left(\mathbf{U}^{t+1} + \mathbf{\Lambda}_3^t\right),\end{aligned} \tag{15}$$

where $\operatorname{prox}_{\lambda\|\cdot\|_1}(\cdot)$ and $\operatorname{prox}_{\lambda\|\cdot\|_{1,\infty}}(\cdot)$ are the proximal operators of an ℓ_1 norm and an $\ell_{1,\infty}$ norm, respectively, which are shown in [34].
The Lagrangian multipliers $\mathbf{\Lambda}_h$, $h = 1, 2, 3$ are updated as follows:

$$\mathbf{\Lambda}_h^{t+1} := \mathbf{\Lambda}_h^{t+1} + \mathbf{U}^{t+1} - \mathbf{Z}_h^{t+1}. \tag{16}$$

Then, the primal and dual residuals $\mathbf{r}^{t+1}$ and $\mathbf{s}^{t+1}$ are given by

$$\mathbf{r}^{t+1} = \mathbf{A}\mathbf{U}^{t+1} + \mathbf{B}\mathbf{Z}^{t+1} \tag{17}$$

$$\mathbf{s}^{t+1} = \rho\mathbf{A}^T\mathbf{B}\left(\mathbf{Z}^{t+1} - \mathbf{Z}^t\right) \tag{18}$$

We set the ADMM parameter ρ to 2 and consider that the ADMM to update $\mathbf{U}$ converges if $\|\mathbf{r}^{t+1}\| \to 0$ and $\|\mathbf{s}^{t+1}\| \to 0$. Note that the updating equation for the variable-latent matrix $\mathbf{U}$ in Eq. (11) does not guarantee sparsity. Thus, after convergence, the final variable-latent matrix $\mathbf{U}$ is given by the second auxiliary variable $\mathbf{Z}_2$, which guarantees sparsity due to the proximal operator of the ℓ_1 norm.

The convergence rate of the ADMM is $O(1/\epsilon)$ iterations for the desired accuracy ϵ [31]. The time complexity per iteration is determined by that of updating $\mathbf{Z}_1$. For a regression problem, if we precompute and store $\left[\sum_{j=1}^T \frac{1}{N_j}\mathbf{v}_j\mathbf{v}_j^T \otimes \mathbf{X}_j^T\mathbf{X}_j^T + \rho\mathbf{I}\right]$ and $\sum_{j=1}^T \frac{1}{N_j}$ vec $\left(\mathbf{X}_j^T\mathbf{y}_j\mathbf{v}_j^T\right)$, it takes $O(D^3M^3)$ cost to solve the linear equation. For a binary classification problem, L-BFGS has superlinear local convergence rate [32].

4.2 Updating V

For a fixed variable-latent matrix $\mathbf{U}$, the problem for the latent-task matrix $\mathbf{V}$ is separable into its column vector $\mathbf{v}_j$ as follows:

$$\mathbf{v}_j = \underset{\mathbf{v}}{\operatorname{argmin}}\, \frac{1}{N_j} L(\mathbf{y}_j, \mathbf{X}_j\mathbf{U}\mathbf{v}) + \mu\left(\|\mathbf{v}\|_k^{sp}\right)^2 \tag{19}$$

The j-th weighting vector $\mathbf{v}_j$ is updated by solving the squared k-support norm regularized regression or logistic regression, where an input matrix becomes $\mathbf{X}_j\mathbf{U}$. The above problem is solved by using an accelerated proximal gradient descent [5], where a step-size is pre-computed from the inverse of the Lipschitz continuous constant of the gradient of the loss function $L(\cdot,\cdot)$.

Algorithm1 VSTG-MTL

```
input
$\mathbf{X}_j$ and $\mathbf{y}_j$: training data for task $j = 1, \dots, T$
$M$: number of latent bases
$\gamma_1, \gamma_2, \mu$: regularization parameters
$k$: parameter for the $k$-support norm
$\rho$: parameter for ADMM
output
$\mathbf{U}$: variable-latent matrix
$\mathbf{V}$: latent-task matrix
$\mathbf{W}$=$\mathbf{U}\mathbf{V}$: Coefficient matrix
procedure
1. Estimate an initial coefficient matrix $\mathbf{W}^{init}$
by using Eqs. (7) and (8).
2. Compute the top-$M$ left singular vectors $\mathbf{P}$, the top-$M$ right
singular vectors $\mathbf{Q}$, and the top-$M$ singular value matrix $\Sigma$
$\mathbf{W}^{init} = \mathbf{P}\Sigma\mathbf{Q}^T$.
3. Estimate initial estimates for $\mathbf{U}^0$ and $\mathbf{V}^0$ as follows:
$\mathbf{U}^0 = \mathbf{P}\Sigma^{\frac{1}{2}}$ and $\mathbf{V}^0 = \Sigma^{\frac{1}{2}}\mathbf{Q}$
4. Repeat step 5 to 13.
5.   Repeat step 6 to 8.
6.     Update the variable-latent matrix $\mathbf{U}$ by using Eq. (11).
7.     Update the auxiliary variables $\mathbf{Z}_h$, $h = 1, 2, 3$
by solving Eqs. (13), (14), and (15).
8.     Update scaled Lagrangian multipliers $\mathbf{\Lambda}_h$, $h = 1, 2, 3$
by using Eq. (16).
9.   until the Frobeneus norms of $\mathbf{r}$ and $\mathbf{s}$ in Eqs. (17) and (18)
converge.
10.   Set the variable-latent matrix $\mathbf{U}$ to be equal to
the second auxiliary variable $\mathbf{Z}_2$.
11.   for $j = 1, \dots, T$ do
12.     Update the weighting vector $\mathbf{v}_j$ by solving Eq. (19).
13.   end for
14. until the objective function in Eq. (6) converges.
```

The convergence rate of the accelerated proximal gradient descent is $O(1/\sqrt{\epsilon})$ iterations for the desired accuracy ϵ [5]. The time complexity per iteration is dominated by computing a proximal operator of the squared k-support norm, which is given by $O((M+k)\log M)$ [21].

4.3 Algorithm

Algorithm 1 summarizes the procedure to optimize the objective function (6). Practically, it may be wasteful to wait until the ADMM converges with high accuracy [7]. Instead, we apply early stopping to the ADMM to update $\mathbf{U}$ with achieving modest accuracy. The maximum iterations for the ADMM should be a few tens. However, the accelerated gradient descent algorithm to update $\mathbf{v}_j$ converges with high accuracy. It may compensates the early stopping when updating $\mathbf{U}$.

5 THEORETICAL ANALYSIS

In this section, we provide an upper bound on excess error of the proposed method based on the previous work from Maurer et al. [28].

Suppose $\mu_1, \ldots, \mu_T$ be probability measures on $\mathbb{R}^D \times \mathbb{R}$. Then, the input matrix $\mathbf{X}_j$ and the output vector $\mathbf{y}_j$ are drawn from the probability measure μ_j with $N_j = N$. We express $\bar{\mathbf{X}} = [\mathbf{X}_1, \ldots, \mathbf{X}_T]$.

The optimization problem (5) is reformulated as follows:

$$\min_{\mathbf{U} \in \mathcal{H}, \mathbf{v}_j \in \mathcal{F}} \frac{1}{NT} \sum_{j=1}^{T} L'\left(\mathbf{y}_j, \mathbf{X}_j \mathbf{U} \mathbf{v}_j\right), \tag{20}$$

where $\mathcal{H} = \left\{\mathbf{x} \in \mathbb{R}^D \rightarrow \left(\mathbf{u}_1^T \mathbf{x}, \ldots, \mathbf{u}_M^T \mathbf{x}\right) \in \mathbb{R}^M : \mathbf{u}_1, \ldots, \mathbf{u}_M \in \mathbb{R}^D, \sum_{m=1}^{M} \|\mathbf{u}_m\|_1 \leq \alpha_1, \sum_{i=1}^{D} \|\mathbf{u}^i\|_{1,\infty} \leq \alpha_2\right\}$, $\mathcal{F} = \left\{\mathbf{z} \in \mathbb{R}^M \rightarrow \mathbf{v}^T \mathbf{z} \in \mathbb{R} : \mathbf{v} \in \mathbb{R}^M, \left(\|\mathbf{v}\|_{sp}^{k}\right)^2 \leq \beta^2\right\}$, and L' is the scaled loss function in [0, 1]. We are interested in the expected error given by

$$\varepsilon\left(\mathbf{U}, \mathbf{v}_1, \ldots, \mathbf{v}_T\right) := \frac{1}{T} \sum_{j=1}^{T} \mathbb{E}_{\left(\mathbf{X}_j, \mathbf{y}_j\right) \sim \mu_j} L'\left(\mathbf{y}_j, \mathbf{X}_j \mathbf{U} \mathbf{v}_j\right).$$

Let $\hat{\mathbf{U}}$ and $\hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_T$ be the optimal solution of the optimization problem (20), then we have the following theorem

THEOREM 5.1. **(Upper bound on excess error).** *If $\alpha_1^2 \leq M$, with probability at least 1- δ in $\bar{\mathbf{X}}$ the excess error is bounded by*

$$\varepsilon\left(\hat{\mathbf{U}}, \hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_T\right) - \min_{\mathbf{U} \in \mathcal{H}, \mathbf{v}_j \in \mathcal{F}} \varepsilon\left(\mathbf{U}, \mathbf{v}_1, \ldots, \mathbf{v}_T\right)$$

$$\leq c_1 \beta M \sqrt{\frac{\|\hat{C}\left(\bar{\mathbf{X}}\right)\|_1}{NT}} + c_2 \beta \sqrt{\frac{M \|\hat{C}\left(\bar{\mathbf{X}}\right)\|_\infty}{N}} + \sqrt{\frac{8 \ln(2/\delta)}{NT}},$$

where $\|\hat{C}\left(\bar{\mathbf{X}}\right)\|_1 = \frac{1}{T} \sum_{j=1}^{T} tr\left(\hat{\Sigma}(\mathbf{X}_j)\right)$, $\|\hat{C}\left(\bar{\mathbf{X}}\right)\|_\infty = \frac{1}{T} \sum_{j=1}^{T} \lambda_{max}\left(\hat{\Sigma}(\mathbf{X}_j)\right)$, $\hat{\Sigma}(\mathbf{X}_j)$ is the empirical covariance of input data for the j-th task, $\lambda_{max}(\cdot)$ is the largest eigenvalue, and c_1 and c_2 are universal constants.

PROOF. We can show that $\sum_{m=1}^{M} \|\hat{\mathbf{u}}_m\|_2^2 \leq \sum_{m=1}^{M} \|\hat{\mathbf{u}}_m\|_1^2 \leq \alpha_1^2 \leq M$ and $\|\hat{\mathbf{v}}_j\|_2^2 \leq \left(\|\hat{\mathbf{v}}_j\|_k^{sp}\right)^2 \leq \beta^2$ considering $\hat{\mathbf{U}} \in \mathcal{H}$ and $\hat{\mathbf{v}}_j \in \mathcal{F}$. Then, the optimization problem (20) satisfies the conditions on Lemma 3 and Theorem 4 in [28] and the result follows. □

The above theorem shows the roles of the hyper-parameters. The constraint parameters α_1 and β should be low enough to satisfy $\alpha_1^2 \leq M$, and produce a tighter bound. Thus, the corresponding regularization parameters γ_1 and μ should be large enough to fulfill the above condition and tighten the bound.

6 EXPERIMENT

In this section, we present experiments conducted to evaluate the effectiveness of our proposed method. We compare our proposed methods with the following benchmark methods:

- **LASSO method**: This single-task learning method learns a sparse prediction model for each task independently.
- **L1+trace norm [36]**: This MTL method simultaneously achieves a low-rank structure and variable selection by penalizing both the nuclear norm and the ℓ_1 norm of the coefficient matrix.
- **Multiplicative multi-task feature learning (MMTFL) [38]**: This MTL method factorizes a coefficient matrix as the product of full rank matrices to select the relevant input variables. In this paper, we set $p = 1$ and $k = 2$.
- **Clustered multi-task learning (CMTL)** [40]: This MTL methods learns disjoint groups among tasks.
- **Group overlap multi-task learning (GO-MTL) [20]**: This MTL method factorizes a coefficient matrix as the product of low-rank matrices and learn an overlapping group structure among tasks by imposing sparsity on the weighting vectors.

The hyper-parameters of all methods are selected by minimizing the error from an inner 10-fold cross validation step or a validation set. To reduce the computational complexity of the proposed method, we set the third regularization parameter μ to be equal to the first regularization parameter γ_1. The regularization parameters of all methods are selected from the search grid $\{2^{-10}, \ldots, 2^3\}$. The number of latent bases M and for GO-MTL and VSTG-MTL and the one of clusters for CMTL is selected from the search grid $\{1, 3, 5, 7, 9, 11, 13\}$. For the synthetic datasets, the value of k is set to 1 (**VSTG-MTL** k**=1**), which is equivalent to the squared ℓ_1 norm, or 3 (**VSTG-MTL** k**=3**) to identify the effectiveness of the k-support norm for correlated features. In the real datasets, it is selected from $\{1, 3, 5, 7\}$ (**VSTG-MTL** k**=opt**). The Matlab implementation of the proposed method is available at the following URL: https://github.com/JunYongJeong/VSTG-MTL.

The evaluation measurements approach used are the root mean squared error (**RMSE**) for a regression problem and the error rate (**ER**) for a classification problem. For synthetic datasets, we also compute the relative estimation error (**REE**) $\|\mathbf{W}^* - \hat{\mathbf{W}}\|_F / \|\mathbf{W}^*\|_F$, where $\mathbf{W}^*$ is the true coefficient matrix and $\hat{\mathbf{W}}$ is the estimated one. We repeat the experiments 10 times and compute the mean and standard deviation of the evaluation measurement. We also perform a Wilcoxon signed rank test with $\alpha = 0.05$, which is a non-parametric paired t-test, to find the best model statistically. The statistically best models are highlighted in bold in Tables 1 and 2.

6.1 Synthetic Datasets

We generate the following four synthetic datasets. We use 25-dimensional variables ($D = 25$) and 20 tasks ($T = 20$). For the j-th task, we generate 50 training observations and 100 test observations from $\mathbf{x}_j^n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$ and $y_j^n = \mathbf{w}_j^T \mathbf{x}_j^n + \mathcal{N}(0, 1)$. A true coefficient matrix $\mathbf{W}^* = [\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*]$ has a low-rank structure $M := rank(\mathbf{W}) = 5$ and is estimated by $\mathbf{UV}$, where $\mathbf{U} \in \mathbb{R}^{D \times M}$ and $\mathbf{V} \in \mathbb{R}^{M \times T}$. Each synthetic dataset differs on the structure of the two matrices $\mathbf{U}$ and $\mathbf{V}$.

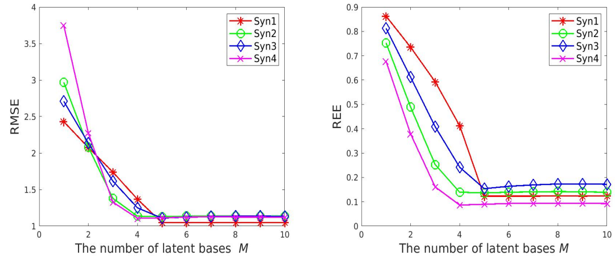
- **Syn1. Orthogonal features and disjoint task groups** : For $m = 1, \ldots, M$, the latent basis $\mathbf{u}_m$ only has non-zero values from the $(4m-3)$-th to the $4m$-th components. The non-zero values are generated through a normal distribution with mean 1.0 and std 0.25. Similarly, the weighting vectors $\mathbf{v}_{4m-3}, \ldots, \mathbf{v}_{4m}$ only have nonzero values on the m-th component. The nonzero values are generated through a uniform distribution from 1 to 1.5. Thus, the last five variables are irrelevant. The latent bases $\mathbf{u}_m$, $m = 1, \ldots, M$, as well as the corresponding features, are orthogonal to each other. Each latent basis $\mathbf{u}_m$ forms a disjoint group, where each group consists of four variables and tasks.

Table 1: Results for the synthetic datasets showing the average RMSE and REE with 10 repetitions. The statistically best models are highlighted in bold.

Synthetic	Measure	LASSO	L1+trace	MMTFL	CMTL	GO-MTL	VSTG-MTL $k = 1$	VSTG-MTL $k = 3$
Syn1	RMSE	1.4625	1.1585	1.1384	1.3170	1.0935	**1.0550**	1.0795
		± 0.1349	± 0.1585	± 0.0257	± 0.0298	± 0.0185	± **0.0228**	± 0.0184
	REE	0.4155	0.2249	0.2089	0.3277	0.1737	0.1226	0.1536
		± 0.0595	± 0.0200	± 0.0169	± 0.0186	± 0.0165	± 0.0149	± 0.0128
Syn2	RMSE	1.6811	1.2639	1.2377	1.3720	1.1509	1.1067	1.1090
		± 0.1146	± 0.0418	± 0.0401	± 0.0497	± 0.0267	± 0.0282	± 0.0258
	REE	0.3703	0.2040	0.1921	0.2479	0.1488	0.1231	0.1230
		± 0.0441	± 0.0169	± 0.0152	± 0.0170	± 0.0122	± 0.0118	± 0.0095
Syn3	RMSE	1.5303	1.2244	1.1797	1.3470	1.1129	1.1013	**1.0068**
		± 0.0483	± 0.0320	± 0.0287	± 0.0334	± 0.0250	± 0.0244	± **0.0201**
	REE	0.3801	0.2262	0.2001	0.2881	0.1565	0.1473	**0.1412**
		± 0.0328	± 0.0211	± 0.0168	± 0.0200	± 0.0148	± 0.0133	± **0.0110**
Syn4	RMSE	1.7380	1.2673	1.2271	1.4418	1.1278	1.0863	**1.0618**
		± 0.1032	± 0.0312	± 0.0309	± 0.0402	± 0.0235	± 0.0225	± **0.0211**
	REE	0.2729	0.1419	0.1302	0.1911	0.0945	0.0768	**0.0741**
		± 0.0365	± 0.0125	± 0.0111	0.0103	± 0.0087	± 0.0117	± **0.0093**

- **Syn2. Orthogonal features and overlapping task groups** : The variable-latent matrix **U** is generated by the same procedure as that shown in Syn1. For $m = 1, \ldots, M - 1$, the weighting vectors $\mathbf{v}_{4m-3}, \ldots, \mathbf{v}_{4m}$ only have nonzero values on the m-th and $(m + 1)$-th components. The last four weighting vectors $\mathbf{v}_{4M-3}, \ldots, \mathbf{v}_{4M}$ only have the nonzero values on the $(M - 1)$-th and M-th components. The nonzero values are generated using the same uniform distribution as that used in Syn1. Then, the last five variables are irrelevant and the features are still orthogonal. The tasks have M overlapping groups, where each group consists of four variables and five tasks.
- **Syn3. Correlated features and disjoint task groups** : For $m = 1, \ldots, M$, the latent basis $\mathbf{u}_m$ only has nonzero values from the $(3m - 2)$-th to the $(3m + 3)$-th components. The nonzero values are generated using the same normal distribution as that used in Syn1. The latent-task matrix **V** is generated using the same procedure as that used in Syn1. The last seven variables are irrelevant and the latent bases are not orthogonal, resulting in correlation among features. The tasks have M disjoint groups, where each group consists of six variables and four tasks.
- **Syn4. Correlated features and overlapping task groups** : The variable-latent matrix **U** is generated using the same procedure as that used in Syn3. The latent-task matrix **V** is generated using the same procedure as that used in Syn2. The last seven input variables are irrelevant. Thus, the features are correlated and the tasks have M overlapping groups, where each group consists of six variables and five tasks.

Table 1 summarizes the experimental results for the four synthetic datasets in terms of RMSE and REE. For all the synthetic datasets, the MTL methods outperform the single-task learning method LASSO and VSTG-MTL exhibits the best performance. Moreover, we can identify the effect of the k-support norm on the correlated features. On Syn1 and Syn2, where the latent bases $\mathbf{u}_m$, $m = 1, \ldots, M$ are orthogonal , **VSTG-MTL** k**=1** outperforms **VSTG-MTL** k=**3**. This results indicates that the squared-ℓ_1 norm penalty performs better than the squared k support norm penalty with $k = 3$ when the features are orthogonal. In contrast, on Syn3 and Syn4, where the latent bases $\mathbf{u}_r$, $m = 1, \ldots, M$ are not orthogonal, **VSTG-MTL** k=**3** outperforms **VSTG-MTL** k=**1**. These results confirm to our premise that the k-support norm penalty can improve generalization performance more than the ℓ_1 norm penalty when correlation exists.



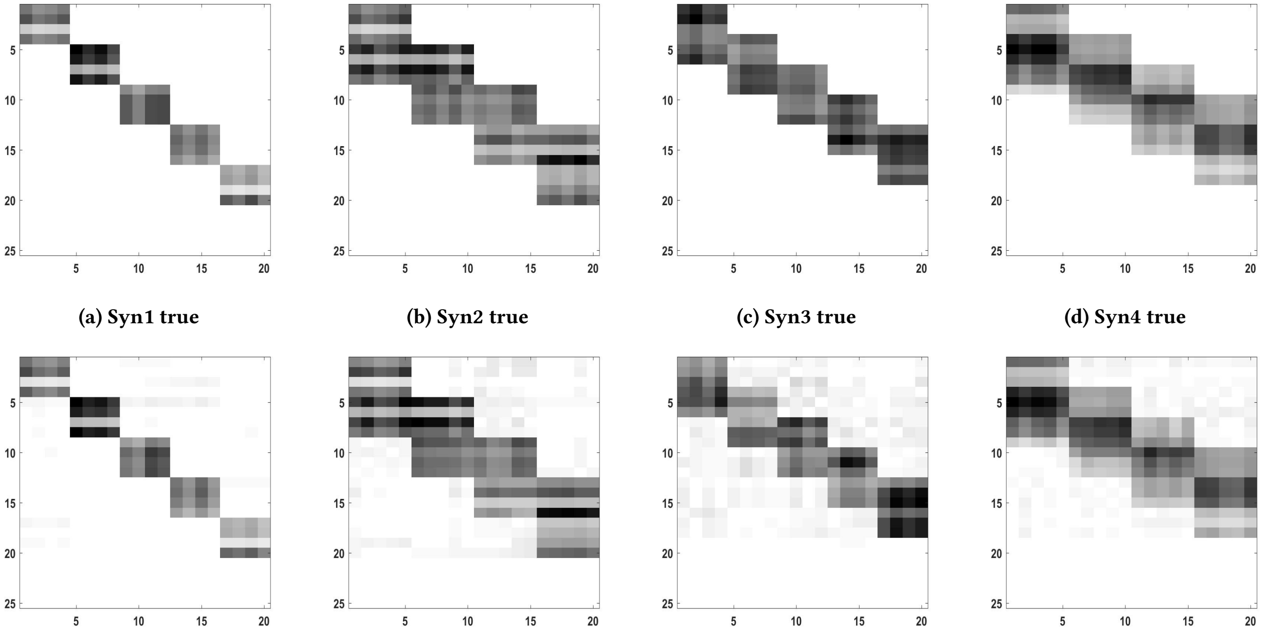
(a) The number of latent bases M vs RMSE

(b) The number of latent bases M vs REE

Figure 2: Effect of the number of latent bases M from the synthetic datasets.

Fig. 2 represents the effect of the number of latent bases M on RMSE, and REE, where k is set to 1 and the other hyper-parameters are selected by cross-validation. Both RMSE and REE is the lowest when the specified M is around the true one, which is five for the synthetic datasets, and converges after that. The sparsity inducing penalty terms make redundant elements in **U** and **V** to zero.

The true coefficient matrix and estimated matrix using the proposed method are shown in Fig. 3, where the dark and white color entries indicate large and zero values, respectively. VSTG-MTL



(a) Syn1 true

(b) Syn2 true

(c) Syn3 true

(d) Syn4 true

(e) Syn1 VSMTL k=1

(f) Syn2 VSMTL k=1

(g) Syn3 VSMTL k=3

(h) Syn4 VSMTL k=3

Figure 3: True and estimated coefficient matrices by VSTG-MTL. The dark and white color entries indicate the large and zero values, respectively.

can recover a group structure among tasks and exclude irrelevant variables.

6.2 Real Datasets

We also evaluate the performance of VSTG-MTL on the following five real datasets. After splitting the dataset into a training set and a test set, we transform the continuous input variables from the training set into $[-1, 1]$ by dividing the maximums of their absolute values. Then, we divide the continuous input variables in the test set by using the same values as those in the training set.

- **School exam dataset**[1] [10]: This multi-task regression dataset is obtained from the Inner London Education Authority. It consists of an examination of 15362 students from 139 secondary schools in London during a three year period: 1985-1987. We have 139 tasks and 15362 observations, where each task and observation correspond to a prediction of the exam scores of a school and a student, respectively. Each observation is represented by 3 continuous and 23 binary variables including school and student-specific attributes. We follow the split procedure shown in [2], resulting in a training set of 75% observations and a test set of 25% observations.
- **Parkinson's disease dataset** [2] [37]: This multi-task regression dataset is obtained from biomedical voice measurements taken from 42 people with early-stage Parkinson's disease. We have 42 tasks and 5875 observations, where each task and observation correspond to a prediction of the symptom score (motor UPDRS) for a patient and a record of a patient, respectively. Each observation is represented by 19 continuous variables including age, gender, time interval, and voice measurements. We use 75% of the observations as a training set and the remaining 25% as a test set.
- **Computer survey dataset** [3] [24]: This multi-output regression dataset is obtained from a survey of 190 ratings from people about their likelihood of purchasing each of the 20 different personal computers. We have 190 tasks and 20 observations shared for all tasks, where each task and observation correspond to a prediction of the integer ratings of a person on a scale of 0 to 10 and a computer. Each observation is represented by 13 binary variables, including its specification. We insert an additional variable to account for the bias term and use 75% of the observations as a training set and the remaining 25% as a test set.
- **MNIST dataset**[4] [22]: This multi-class classification dataset is obtained from 10 handwritten digits. We have 10 tasks, 60,000 training observations and 10,000 test observations, where each task and observation correspond to a prediction of the digit and an image, respectively. Each observations is represented by 28×28 variables and reduced to 64 dimensions using PCA. Train, validation and test set are generated by randomly selecting 1,000 observations from the train set of

[1] http://ttic.uchicago.edu/~argyriou/code/index.html

[2] http://archive.ics.uci.edu/ml/datasets/Parkinsons+Telemonitoring

[3] https://github.com/probml/pmtk3/tree/master/data/conjointAnalysisComputerBuyers

[4] http://yann.lecun.com/exdb/mnist/

Table 2: Results form the real datasets for the RMSE and ER over 10 repetitions. The statistically best models are highlighted in bold.

Dataset	Measure	LASSO	L1+Trace	MMTFL	CMTL	GO-MTL	VSTG-MTL k=opt
School exam	RMSE	12.0483	10.5041	10.1303	10.0170	10.1924	**9.8931**
		± 0.1738	± 0.1432	± 0.1291	± 0.1979	± 0.1331	± **0.1103**
Parkinson		2.9177	1.0481	1.1079	1.0408	1.0231	1.0077
		± 0.0960	± 0.0243	± 0.0182	± 0.0229	± 0.0285	± 0.0191
Computer survey		2.3119	4.9493	1.7525	2.7562	1.9067	**1.6993**
		± 0.3997	± 2.1592	± 0.1237	± 0.6336	± 0.1864	± **0.1053**
MNIST	ER	13.0200	17.9800	12.6000	12.3400	12.8400	**11.7000**
		± 0.7084	± 1.7574	± 0.8641	± 0.0199	1.2989	± **1.4461**
USPS		12.8800	16.0200	11.3600	12.4400	12.9000	11.4800
		± 1.5061	± 1.2874	± 1.1462	± 0.0099	± 1.0842	± 1.0379

and two sets of 500 observations from the test set, similar to the procedure of Kang et al. [17].

- **USPS dataset**[5] [14]: This multi-class classification dataset is also obtained from the 10 handwritten digits. We have 10 tasks, 7,291 training observations and 2,007 test observations, where each task and observation correspond to a prediction of the digit and an image, respectively. Each observation is represented by 16×16 variables and reduced to 87 dimensions using PCA. We follow the same procedure of that used in the MNIST dataset to generate train, validation and test set, resulting in 1000, 500, and 500 observations, respectively.

Table 2 summarizes the results for the five real datasets over 10 repetitions. **VSTG-MTL** k**=opt** outperforms the benchmark methods except the USPS dataset. This is especially true for the school exam, the computer survey and the MNIST dataset, where the proposed method shows statistically significant improvements over the benchmark methods.

7 CONCLUSION

This paper proposes a novel algorithm of VSTG-MTL, which simultaneously performs variable selection and learns an overlapping group structure among tasks. VSTG-MTL factorizes a coefficient matrix into the product of low-rank matrices and impose sparsities on them while considering possible correlations. The resulting bi-convex constrained problem is transformed to a regularized problem that is solved by alternating optimization. We provide the upper bound on the excess risk of the proposed method. The experimental results show that the proposed VSTG-MTL method outperforms the benchmark methods on synthetic as well as real datasets.

Future work would be improving the computational efficiency of the proposed method. We can replace an alternating optimization with a proximal alternating linearized minimization [6]. The optimization of $\mathbf{v}_j$ can be improved by adopting a fully corrective Frank-Wolfe Method [25].

ACKNOWLEDGMENTS

This study was supported by National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No 2017R1A2B4005450)

[5] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

REFERENCES

[1] Rie Kubota Ando and Tong Zhang. 2005. A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. *Journal of Machine Learning Research* 6 (Dec. 2005), 1817–1853.

[2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2008. Convex Multi-Task Feature Learning. *Machine Learning* 73, 3 (01 Dec 2008), 243–272. https://doi.org/10.1007/s10994-007-5040-8

[3] Andreas Argyriou, Rina Foygel, and Nathan Srebro. 2012. Sparse Prediction with the k-Support Norm. In *Advances in Neural Information Processing Systems 25 (NIPS'12)*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1457–1465.

[4] Aviad Barzilai and Koby Crammer. 2015. Convex Multi-Task Learning by Clustering. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS'2015)*, Guy Lebanon and S. V. N. Vishwanathan (Eds.), Vol. 38. PMLR, San Diego, California, USA, 65–73.

[5] Amir Beck and Marc Teboulle. 2009. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences* 2, 1 (2009), 183–202. https://doi.org/10.1137/080716542

[6] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. 2014. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming* 146, 1 (01 Aug 2014), 459–494. https://doi.org/10.1007/s10107-013-0701-9

[7] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends® in Machine Learning* 3, 1 (Jan. 2011), 1–122. https://doi.org/10.1561/2200000016

[8] Rich Caruana. 1997. Multitask Learning. *Machine Learning* 28, 1 (01 Jul 1997), 41–75. https://doi.org/10.1023/A:1007379606734

[9] Lisha Chen and Jianhua Z.Huang. 2012. Sparse Reduced-Rank Regression for Simultaneous Dimension Reduction and Variable Selection. *J. Amer. Statist. Assoc.* 107, 500 (2012), 1533–1545. https://doi.org/10.1080/01621459.2012.734178

[10] Harvey Goldstein. 1991. Multilevel Modelling of Survey Data. *Journal of the Royal Statistical Society. Series D (The Statistician)* 40, 2 (1991), 235–244. http://www.jstor.org/stable/2348496

[11] Lei Han and Yu Zhang. 2015. Learning Tree Structure in Multi-Task Learning. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)*. ACM, New York, NY, USA, 397–406. https://doi.org/10.1145/2783258.2783393

[12] Lei Han and Yu Zhang. 2016. Multi-Stage Multi-Task Learning with Reduced Rank. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*. AAAI Press, 1638–1644.

[13] Daniel Hernandez-Lobato, Jose Miguel Hernandez-Lobato, and Zoubin Ghahramani. 2015. A Probabilistic Model for Dirty Multi-task Feature Selection. In *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, Francis Bach and David Blei (Eds.), Vol. 37. PMLR, Lille, France, 1073–1082.

[14] Jonathan J. Hull. 1994. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16, 5 (May 1994), 550–554. https://doi.org/10.1109/34.291440

[15] Laurent Jacob, Francis Bach, and Jean-Philippe Vert. 2008. Clustered Multi-Task Learning: A Convex Formulation. In *Advances in Neural Information Processing Systems 21 (NIPS'08)*. Curran Associates, Inc., USA, 745–752.

[16] Ali Jalali, Pradeep Ravikumar, Sujay Sanghavi, and Chao Ruan. 2010. A Dirty Model for Multi-Task Learning. In *Advances in Neural Information Processing Systems 23 (NIPS'10)*. Curran Associates, Inc., USA, 964–972.

[17] Zhuoliang Kang, Kristen Grauman, and Fei Sha. 2011. Learning with Whom to Share in Multi-task Feature Learning. In *Proceedings of the 28th International*

Conference on Machine Learning (ICML'11). Omnipress, USA, 521–528.

[18] Zhuoliang Kang, Kristen Grauman, and Fei Sha. 2011. Learning with Whom to Share in Multi-task Feature Learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML'11)*. Omnipress, USA, 521–528.

[19] Seyoung Kim and Eric P. Xing. 2010. Tree-guided Group Lasso for Multi-task Regression with Structured Sparsity. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML'10)*. Omnipress, USA, 543–550. http://dl.acm.org/citation.cfm?id=3104322.3104392

[20] Abhishek Kumar and Hal Daumé III. 2012. Learning Task Grouping and Overlap in Multi-Task Learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML'12)*, John Langford and Joelle Pineau (Eds.). Omnipress, NY, USA, 1383–1390.

[21] Hanjiang Lai, Yan Pan, Canyi Lu, Yong Tang, and Shuicheng Yan. 2014. Efficient k-Support Matrix Pursuit. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 617–631.

[22] Yann Lecun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (Nov 1998), 2278–2324. https://doi.org/10.1109/5.726791

[23] Giwoong Lee, Eunho Yang, and Sung Hwang. 2016. Asymmetric Multi-task Learning Based on Task Relatedness and Loss. In *Proceedings of the 33rd International Conference on Machine Learning (ICML'16)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.), Vol. 48. PMLR, New York, USA, 230–238.

[24] Peter J. Lenk, Wayne S. DeSarbo, Paul E. Green, and Martin R. Young. 1996. Hierarchical Bayes Conjoint Analysis: Recovery of Partworth Heterogeneity from Reduced Experimental Designs. *Marketing Science* 15, 2 (1996), 173–191. https://doi.org/10.1287/mksc.15.2.173

[25] Bo Liu, Xiao-Tong Yuan, Shaoting Zhang, Qingshan Liu, and Dimitris N. Metaxas. 2016. Efficient K-support-norm Regularized Minimization via Fully Corrective frank-Wolfe Method. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16)*. AAAI Press, 1760–1766.

[26] Han Liu, Mark Palatucci, and Jian Zhang. 2009. Blockwise Coordinate Descent Procedures for the Multi-Task Lasso, with Applications to Neural Semantic Basis Discovery. In *Proceedings of the 26th International Conference on Machine Learning (ICML'09)*. ACM, New York, NY, USA, 649–656.

[27] Sulin Liu and Sinno Jialin Pan. 2017. Adaptive Group Sparse Multi-task Learning via Trace Lasso. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. AAAI Press, 2358–2364.

[28] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. 2016. The Benefit of Multitask Representation Learning. *Journal of Machine Learning Research* 17, 81 (2016), 1–32.

[29] Andrew M. McDonald, Massimiliano Pontil, and Dimitris Stamos. 2016. New Perspectives on K-support and Cluster Norms. *Journal of Machine Learning Research* 17, 1 (Jan. 2016), 5376–5413.

[30] Bamdev Mishra, Gilles Meyer, Francis Bach, and Rodolphe Sepulchre. 2013. Low-Rank Optimization with Trace Norm Penalty. *SIAM Journal on Optimization* 23, 4 (2013), 2124–2149. https://doi.org/10.1137/110859646

[31] Robert Nishihara, Laurent Lessard, Benjamin Recht, Andrew Packard, and Michael I. Jordan. 2015. A General Analysis of the Convergence of ADMM. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37 (ICML'15)*. PMLR, 343–352.

[32] Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization (2nd. ed.)*. Springer-Verlag New York. https://doi.org/10.1007/978-0-387-40065-5

[33] Guillaume Obozinski, Ben Taskar, and Michael I. Jordan. 2010. Joint Covariate selection and Joint Subspace Selection for Multiple Classification Problems. *Statistics and Computing* 20, 2 (01 Apr 2010), 231–252. https://doi.org/10.1007/s11222-008-9111-x

[34] Neal Parikh and Stephen Boyd. 2014. Proximal Algorithms. *Foundations and Trends® in Optimization* 1, 3 (2014), 127–239. https://doi.org/10.1561/2400000003

[35] Emile Richard, Francis BACH, and Jean-Philippe Vert. 2013. Intersecting Singularities for Multi-Structured Estimation. In *Proceedings of the 30th International Conference on Machine Learning (ICML'13)*, Sanjoy Dasgupta and David McAllester (Eds.), Vol. 28. PMLR, Atlanta, Georgia, USA, 1157–1165.

[36] Emile Richard, Pierre-André Savalle, and Nicolas Vayatis. 2012. Estimation of Simultaneously Sparse and Low Rank Matrices. In *Proceedings of the 29th International Conference on Machine Learning (ICML'12)*. Omnipress, USA, 51–58.

[37] Athanasios Tsanas, Max A. Little, Patrick E. McSharry, and Lorraine O. Ramig. 2010. Accurate Telemonitoring of Parkinson's Disease Progression by Noninvasive Speech Tests. *IEEE Transactions on Biomedical Engineering* 57, 4 (April 2010), 884–893. https://doi.org/10.1109/TBME.2009.2036000

[38] Xin Wang, Jinbo Bi, Shipeng Yu, Jiangwen Sun, and Minghu Song. 2016. Multiplicative Multitask Feature Learning. *Journal of Machine Learning Research* 17, 80 (2016), 1–33.

[39] Yu Zhang and Qiang Yang. 2017. A Survey on Multi-Task Learning. *CoRR* abs/1707.08114 (2017). arXiv:1707.08114 http://arxiv.org/abs/1707.08114

[40] Jiayu Zhou, Jianhui Chen, and Jieping Ye. 2011. Clustered Multi-Task Learning via Alternating Structure Optimization. In *Advances in Neural Information Processing Systems 24 (NIPS'11)*. Curran Associates, Inc., 702–710.

[41] Qiang Zhou and Qi Zhao. 2016. Flexible Clustered Multi-Task Learning by Learning Representative Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 2 (Feb 2016), 266–278. https://doi.org/10.1109/TPAMI.2015.2452911

[42] Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67, 2 (2005), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x