Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Learning Sparse Task Relations in Multi-Task Learning

Yu Zhang, Qiang Yang
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
{zhangyu,qyang}@cse.ust.hk

Abstract

In multi-task learning, when the number of tasks is large, pairwise task relations exhibit sparse patterns since usually a task cannot be helpful to all of the other tasks and moreover, sparse task relations can reduce the risk of overfitting compared with the dense ones. In this paper, we focus on learning sparse task relations. Based on a regularization framework which can learn task relations among multiple tasks, we propose a SParse covAriance based mulTi-taSk (SPATS) model to learn a sparse covariance by using the ℓ_1 regularization. The resulting objective function of the SPATS method is convex, which allows us to devise an alternating method to solve it. Moreover, some theoretical properties of the proposed model are studied. Experiments on synthetic and real-world datasets demonstrate the effectiveness of the proposed method.

Introduction

Multi-task learning (Caruana 1997; Baxter 1997), which is inspired by human learning ability, aims to help improve the generalization performance of several tasks simultaneously by leveraging useful but hidden common knowledge among them. Multi-task learning has many applications in various areas, including data mining, computer vision, bioinformatics, and health informatics.

A key issue in multi-task learning is to understand the relations between tasks since this understanding can be incorporated into the learning process to improve the generalization performance of all the tasks. At the early stage, many multi-task methods focus on utilizing a priori information on the task relations. For example, based on the domain knowledge that all the tasks are similar to each other, Evgeniou and Pontil (Evgeniou and Pontil 2004) extend the support vector machine to the multi-task setting by proposing a regularizer to enforce the model parameters of all the tasks to be close to the average one and Evgeniou et al. (Evgeniou, Micchelli, and Pontil 2005) as well as Kato et al. (Kato et al. 2007) devise some Laplacian-based regularizers depending on the given task similarity graphs to make the model parameters corresponding to any pair of similar tasks close to each other. Moreover, Han et al. (Han et al. 2014) utilize the given hierarchical structure among different tasks as a priori information to help learn model parameters. However, in real-world applications, such a priori information on task relations may be not easy to obtain or even does not exist, bringing difficulties to the wide use of such approaches. In recent years, many multi-task methods are proposed to learn the task relations directly from data. Usually the task relations appear in different forms and hence different approaches are proposed to learn them. For example, the multi-task feature learning method proposed in (Argyriou, Evgeniou, and Pontil 2006) aims at learning a common subset of features for all the tasks based on group-sparsity regularizers after some linear transformation on the original feature representation, and the multi-task feature selection method (Obozinski, Taskar, and Jordan 2006) adopts the same idea by learning the common features based on the original feature representation. By assuming that the model parameters of all the tasks share a low-rank subspace, Ando and Zhang (Ando and Zhang 2005) directly learn the shared subspace as well as the task-specific spanning coefficients under a non-convex formulation and then Chen et al. (Chen et al. 2009) relax the non-convex formulation to a convex one, which is easier to solve. With a similar idea to (Ando and Zhang 2005), Pong et al. (Pong et al. 2010) use the trace norm regularization to learn a low-rank parameter matrix and then Han and Zhang (Han and Zhang 2016) extend it to the capped trace norm penalty. The task clustering approach such as (Xue et al. 2007; Jacob, Bach, and Vert 2008; Kang, Grauman, and Sha 2011; Jawanpuria and Nath 2012; Han and Zhang 2015a) can detect the cluster structure among tasks where tasks from a cluster are similar to each other in terms of model parameters or feature representations. All these methods in this approach can identify task clusters but in different ways, including utilizing the Dirichlet process in (Xue et al. 2007), devising some regularizer inspired by the k-means clustering in (Jacob, Bach, and Vert 2008), integer programming in (Kang, Grauman, and Sha 2011), group-sparsity regularizers in (Jawanpuria and Nath 2012; Han and Zhang 2015a) and so on. Interestingly, Han and Zhang (Han and Zhang 2015b) propose a method to learn hierarchical structure among tasks based on sequential constraints. Moreover, several methods directly learn pairwise relations among tasks, where the pairwise relations are represented by a covariance matrix (Bonilla, Chai, and Williams 2007; Zhang and Yeung 2010a; 2010b; Zhang and Schneider 2010) or just a square matrix (Zhang 2013). The task relations

reflected in the matrix can give us deep insight and understanding on the tasks, which can enhance the interpretation of the multi-task models. When the number of tasks is large, usually one task cannot be helpful to all of the other tasks, implying that the pairwise task relations could be sparse. Moreover, learning densely pairwise relations may lead to high model complexity and also increased risk for overfitting. In this sense, learning sparse task relations is an important problem.

In this paper, we propose a new method to learn the sparse task relations. Based on a regularization framework for multi-task learning to learn task relations, we propose a SParse covAriance based mulTi-taSk (SPATS) method for learning sparse task relations. Since the covariance used in the framework is to model the task relations, the proposed SPATS method learns a sparse covariance by placing a ℓ_1 regularization on it. The resulting objective function of the SPATS method is convex, which allows us to devise an alternating method to solve it. Some theoretical properties of the proposed SPATS method are studied. Experiments on synthetic and real-world datasets demonstrate the effectiveness of the proposed method.

A Framework to Learn Task Relations

Suppose we are given a set of m tasks $\{\mathcal{T}_i\}_{i=1}^m$. For task $\mathcal{T}_i$, its training set contains n_i data points $\{\mathbf{x}_j^i\}_{j=1}^{n_i}$ as well as their labels $\{y_j^i\}_{j=1}^{n_i}$ where $\mathbf{x}_j^i \in \mathbb{R}^d$ and $y_j^i \in \mathbb{R}$ for regression problems and $y_j^i \in \{-1, 1\}$ for classification problems. The learning function for task $\mathcal{T}_i$ is defined as $f_i(\mathbf{x}) = \mathbf{w}_i^T \phi(\mathbf{x}) + b_i$, where $\phi(\cdot)$ denotes the feature mapping from $\mathbb{R}^d$ to $\mathbb{R}^{\hat{d}}$. Some examples for $\phi(\cdot)$ are $\phi(\mathbf{x}) = \mathbf{x}$ and the feature mapping induced by some kernel function such that the dot product between $\phi(\mathbf{x}_1)$ and $\phi(\mathbf{x}_2)$ is equal to $k(\mathbf{x}_1, \mathbf{x}_2)$ where $k(\cdot, \cdot)$ denotes a kernel function.

In the following, we present a regularized framework for learning multiple tasks and modeling the task relations simultaneously:

$$\min_{\mathbf{W},\mathbf{b},\mathbf{\Omega}\succeq\mathbf{0}} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(\mathbf{w}_i^T \phi(\mathbf{x}_j^i) + b_i, y_j^i\right) + \frac{\lambda_1}{2} \mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T) + \lambda_2 g(\mathbf{\Omega}), \quad (1)$$

where $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_m)$, $\mathbf{b} = (b_1, \ldots, b_m)^T$, $l(\cdot, \cdot)$ denotes a loss function, $\mathbf{0}$ denotes a zero vector or matrix with appropriate size, $\mathbf{A} \succ (\succeq) \mathbf{B}$ for two square matrices $\mathbf{A}$ and $\mathbf{B}$ means that $\mathbf{A} - \mathbf{B}$ is positive definite (PD) (positive semidefinite (PSD)), $\mathrm{tr}(\cdot)$ denotes the trace of a square matrix, $\mathbf{M}^{-1}$ denotes the inverse or pseduoinverse of a square matrix depending on whether it is nonsingular or not, and λ_1, λ_2 are regularization parameters to control the trade-off among three terms in problem (1).

Problem (1) contains three terms. The first term measures the empirical loss on the training data based on the loss function $l(\cdot, \cdot)$. The second term is a regularizer on $\mathbf{W}$ based on $\mathbf{\Omega}$. For example, when $\mathbf{\Omega} \propto \mathbf{I}$ where $\mathbf{I}$ denotes an identity matrix with appropriate size, the second term is the squared Frobenius norm regularization on $\mathbf{W}$ and if $\mathbf{\Omega}$ is a diagonal matrix, it becomes the weighted Frobenius norm regularization for $\mathbf{W}$. Similar to (Zhang and Yeung 2010a), $\mathbf{\Omega}$, which is assumed to be PSD, can be viewed as a covariance matrix to describe the pairwise task relations. The function $g(\cdot)$ in the last term of problem (1) can be considered as a regularizer on $\mathbf{\Omega}$ to specify its structure. When $g(\cdot)$ is an indicator function for some set, problem (1) becomes a constrained problem with the constraints defining the structure of $\mathbf{\Omega}$. Problem (1) defines a framework for multi-task learning, which depends on the choice of the function $g(\cdot)$, to learn the task relations via $\mathbf{\Omega}$. It is not difficult to reveal that many existing multi-task methods (Evgeniou and Pontil 2004; Evgeniou, Micchelli, and Pontil 2005; Jacob, Bach, and Vert 2008; Pong et al. 2010; Zhang and Yeung 2010a; Zhang and Schneider 2010; Rai, Kumar, and Daume 2012; Zhang and Yeung 2014) can be viewed as concrete instances under this framework.

As proved in (Zhang and Yeung 2010a), the second term in problem (1) is jointly convex with respect to $\mathbf{W}$ and $\mathbf{\Omega}$. Given a convex loss function $l(\cdot, \cdot)$ and a convex function $g(\cdot)$, problem (1) is also convex, which can bring computational benefits.

Moreover, problem (1) can be viewed as a regularized multi-task framework as

$$\min_{\mathbf{W},\mathbf{b}} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(\mathbf{w}_i^T \phi(\mathbf{x}_j^i) + b_i, y_j^i\right) + \lambda_1 R(\mathbf{W}),$$

where $R(\mathbf{W})$ is defined as

$$R(\mathbf{W}) = \min_{\mathbf{\Omega}\succeq\mathbf{0}} \frac{1}{2} \mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T) + \frac{\lambda_2}{\lambda_1} g(\mathbf{\Omega}). \quad (2)$$

Different choices of $g(\cdot)$ lead to different regularizers $R(\cdot)$. It is easy to see that some simple $g(\cdot)$ can induce several well-known regularizers including the (squared) trace norm regularizer and the (squared) Schatten norm regularizer.

The SPATS Method

Recall that the goal here is to learn the sparse task relations. In the proposed framework (1), $\mathbf{\Omega}$ corresponds to the task relations and hence we expect to learn a sparse $\mathbf{\Omega}$, which is different from existing multi-task models which learn dense $\mathbf{\Omega}$'s.

Here we propose the SParse covAriance based mulTi-taSk (SPATS) method, which assumes that the task covariance is sparse. Specifically, the objective function of the SPATS method is formulated as

$$\min_{\mathbf{W},\mathbf{b},\mathbf{\Omega}\succeq\mathbf{0}} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(\mathbf{w}_i^T \phi(\mathbf{x}_j^i) + b_i, y_j^i\right) + \frac{\lambda_1}{2} \mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T) + \lambda_2 \|\mathbf{\Omega}\|_1, \quad (3)$$

where the ℓ_1 norm is used to enforce the sparsity of the task covariance $\mathbf{\Omega}$. Problem (3) is an instance of problem (1) by setting $g(\mathbf{\Omega})$ to be $\|\mathbf{\Omega}\|_1$. Since $\|\mathbf{\Omega}\|_1 = \mathrm{tr}(\mathbf{\Omega}) + \sum_{i=1}^m \sum_{j=1, j\neq i}^m |\omega_{ij}|$, problem (3) not only behaves similarly to the trace norm regularization by penalizing the trace of $\mathbf{\Omega}$ but also learns sparse task relations via the penalization of the off-diagonal entries in $\mathbf{\Omega}$. Note that problem (3) is different from the methods proposed in (Zhang and Schneider 2010; Rai, Kumar, and Daume 2012) whose $g(\cdot)$ takes the form of $g(\mathbf{\Omega}) = \|\mathbf{\Omega}^{-1}\|_1$ by assuming that $\mathbf{\Omega}^{-1}$ is sparse. Moreover, given the convex loss

function $l(\cdot,\cdot)$, problem (3) is jointly convex with respect to $\mathbf{W}$, $\mathbf{b}$, and $\mathbf{\Omega}$ according to (Zhang and Yeung 2010a) but the methods proposed in (Zhang and Schneider 2010; Rai, Kumar, and Daume 2012) are non-convex.

Properties

By placing the ℓ_1 regularization on $\mathbf{\Omega}$, we expect to learn a sparse $\mathbf{\Omega}$ and meanwhile restrict the complexity of $\mathbf{\Omega}$. The zero entries in $\mathbf{\Omega}$ implies the corresponding pairs of tasks are unrelated. To see this, we have the following theorem.

Theorem 1 *Suppose ω_{ij}, the (i,j)th element in $\mathbf{\Omega}$, is equal to 0. Then the optimal $\mathbf{w}_i$ is not spanned by $\phi(\mathbf{X}_j)$ and similarly $\mathbf{w}_j$ is not spanned by $\phi(\mathbf{X}_i)$, where $\phi(\mathbf{X}_j) = (\phi(\mathbf{x}_1^j),\ldots,\phi(\mathbf{x}_{n_j}^j))$ is the data matrix for the jth task.*

Proof. First we consider multi-task classification problems where the loss function $l(y_1,y_2)$ can be reformulated as a function of the margin y_1y_2, that is, $l\left(\mathbf{w}_i^T\phi(\mathbf{x}_j^i)+b_i, y_j^i\right) = c\left(y_j^i(\mathbf{w}_i^T\phi(\mathbf{x}_j^i)+b_i)\right)$ for some function $c(\cdot)$. In this case, we set the derivative of problem (3) with respect to $\mathbf{W}$ to be zero and get

$$\lambda_1\mathbf{W}\mathbf{\Omega}^{-1} = \sum_{p=1}^{m}\frac{-1}{n_p}\sum_{q=1}^{n_p} c'\left(y_q^p(\mathbf{w}_p^T\phi(\mathbf{x}_q^p)+b_p)\right)y_q^p\phi(\mathbf{x}_q^p)(\mathbf{e}_p^m)^T,$$

where $c'(\cdot)$ denotes the (sub)gradient of $c(\cdot)$ and $\mathbf{e}_i^m$ denotes the ith canonical basis of $\mathbb{R}^m$. By denoting $-\frac{1}{n_i}c'\left(y_j^i(\mathbf{w}_i^T\phi(\mathbf{x}_j^i)+b_i)\right)$ by α_j^i and utilizing a fact that $\mathbf{w}_i = \mathbf{W}\mathbf{e}_i^m$, we can obtain the representer theorem as

$$\begin{aligned}\mathbf{w}_i &= \frac{1}{\lambda_1}\sum_{p=1}^{m}\sum_{q=1}^{n_p}\alpha_q^p y_q^p\phi(\mathbf{x}_q^p)(\mathbf{e}_p^m)^T\mathbf{\Omega}\mathbf{e}_i^m\\ &= \frac{1}{\lambda_1}\sum_{p=1}^{m}\sum_{q=1}^{n_p}\alpha_q^p y_q^p\phi(\mathbf{x}_q^p)\omega_{pi}.\end{aligned}$$

When ω_{ij} is 0 which implies that $\omega_{ji}=0$ due to the symmetry of $\mathbf{\Omega}$, the above equation can be simplified as

$$\mathbf{w}_i = \frac{1}{\lambda_1}\sum_{\substack{p=1\\p\neq j}}^{m}\sum_{q=1}^{n_p}\alpha_q^p y_q^p\phi(\mathbf{x}_q^p)\omega_{pi}. \tag{4}$$

So $\mathbf{w}_i$ lies in a span of all the training data except those from the jth task. So does $\mathbf{w}_j$.

For multi-task regression problems, the loss function can be formulated as $l(y_1,y_2) = \hat{c}(y_2-y_1)$ for some function $\hat{c}(\cdot)$. The optimality condition for $\mathbf{W}$ gives $\lambda_1\mathbf{W}\mathbf{\Omega}^{-1} = \sum_{p=1}^{m}\sum_{q=1}^{n_p}\beta_q^p\phi(\mathbf{x}_q^p)(\mathbf{e}_p^m)^T$, where $\beta_q^p = \frac{1}{n_i}\hat{c}'\left(y_j^i - \mathbf{w}_p^T\phi(\mathbf{x}_q^p)+b_p\right)$. Similarly, when $\omega_{ij}=0$, we have

$$\mathbf{w}_i = \frac{1}{\lambda_1}\sum_{\substack{p=1\\p\neq j}}^{m}\sum_{q=1}^{n_p}\beta_q^p\phi(\mathbf{x}_q^p)\omega_{pi}, \tag{5}$$

in which we reach the conclusion. □

According to Theorem 1, we can see that when ω_{ij} is 0, the training data of the jth task are not used to compute $\mathbf{w}_i$, which verifies the unrelatedness of those two tasks. Moreover, based on Eqs. (4) and (5), the ℓ_1 regularization on $\mathbf{\Omega}$ leads to a 'sparse' representer theorem in Corollary 1, which is different from the conventional multi-task representer theorem (Argyriou, Micchelli, and Pontil 2009) where each $\mathbf{w}_i$ lies in the span of the training data from all the tasks.

Corollary 1 *For problem (3), the optimal solution satisfies the following representer theorem as*

$$\mathbf{w}_i = \frac{1}{\lambda_1}\sum_{p:\omega_{pi}\neq 0}\sum_{q=1}^{n_p}\gamma_q^p\phi(\mathbf{x}_q^p)\omega_{pi} \text{ for } i=1,\ldots,m,$$

where $\gamma_q^p\in\mathbb{R}$.

Moreover, we investigate the regularizer $R(\mathbf{W})$ defined in Eq. (2) for problem (3). That is, we need to solve the following problem

$$\min_{\mathbf{\Omega}\succeq\mathbf{0}}\ \mathrm{tr}(\mathbf{\Omega}^{-1}\mathbf{S}) + \tau\|\mathbf{\Omega}\|_1, \tag{6}$$

where $\mathbf{S} = \frac{1}{2}\mathbf{W}^T\mathbf{W}$ and $\tau = \frac{\lambda_2}{\lambda_1}$. By utilizing the dual norm of the ℓ_1 norm, we can rewrite problem (6) as

$$\min_{\mathbf{\Omega}\succeq\mathbf{0}}\max_{\|\mathbf{U}\|_\infty\leq\tau}\ \mathrm{tr}(\mathbf{\Omega}^{-1}\mathbf{S}) + \mathrm{tr}(\mathbf{U}^T\mathbf{\Omega}), \tag{7}$$

where $\mathbf{U}$ is a dual variable and $\|\cdot\|_\infty$, the ℓ_∞ norm of a vector or matrix, is equal to the maximum entry of the corresponding vector or matrix. Problem (7) is convex with respect to $\mathbf{\Omega}$ as proved in Theorem 3 and concave with respect to $\mathbf{U}$, leading to an equivalent formulation as

$$\max_{\|\mathbf{U}\|_\infty\leq\tau}\min_{\mathbf{\Omega}\succeq\mathbf{0}}\ \mathrm{tr}(\mathbf{\Omega}^{-1}\mathbf{S}) + \mathrm{tr}(\mathbf{U}^T\mathbf{\Omega}). \tag{8}$$

By solving the inner optimization problem with respect to $\mathbf{\Omega}$, we can obtain the relation between the primal variable $\mathbf{\Omega}$ and the dual variable $\mathbf{U}$ as

$$\mathbf{U} = \mathbf{\Omega}^{-1}\mathbf{S}\mathbf{\Omega}^{-1}. \tag{9}$$

According to this relation, the dual variable $\mathbf{U}$ is PSD since $\mathbf{S}$ is PSD. Moreover, for the optimal $\mathbf{\Omega}$, we have the following result.

Theorem 2 *The optimal $\mathbf{\Omega}$ of problem (6) satisfies $\mathbf{\Omega}\succeq\frac{\mu_m(\mathbf{W})}{\sqrt{2m\tau}}\mathbf{I}$.*

Proof. Based on Eq. (9), the smallest eigenvalue of $\mathbf{S}$, $\mu_m(\mathbf{S})$, satisfies

$$\begin{aligned}\mu_m(\mathbf{S}) &= \mu_m(\mathbf{\Omega}\mathbf{U}\mathbf{\Omega}) = \mu_m(\mathbf{U}\mathbf{\Omega}^2)\\ &\leq \mu_m(\mathbf{\Omega}^2)\mu_1(\mathbf{U}) = \mu_m^2(\mathbf{\Omega})\mu_1(\mathbf{U}),\end{aligned}$$

where the first equality holds due to Eq. (9), the second equality holds due to a fact that $\mathbf{AB}$ and $\mathbf{BA}$ have the same spectrum for two square matrices $\mathbf{A}$ and $\mathbf{B}$, the inequality holds because of a fact that $\mu_m(\mathbf{AB})\leq\mu_m(\mathbf{A})\mu_1(\mathbf{B})$ for any PSD matrices $\mathbf{A},\mathbf{B}\in\mathbb{R}^{m\times m}$, and the last equality holds since $\mu_i(\mathbf{M}^2)$ equals $\mu_i^2(\mathbf{M})$ for any PSD matrix $\mathbf{M}$. Since $\|\mathbf{U}\|_\infty\leq\tau$, then we have $\mu_1(\mathbf{U})\leq\mathrm{tr}(\mathbf{U})\leq m\tau$, where the first inequality holds since $\mathbf{U}$ is PSD and the second one holds since each diagonal element in $\mathbf{U}$ is no larger than τ. Combining the above two inequalities, we can get $\mu_m(\mathbf{\Omega})\geq\sqrt{\frac{\mu_m(\mathbf{S})}{m\tau}}$. Note that $\mathbf{S}=\frac{1}{2}\mathbf{W}^T\mathbf{W}$. We can get $\mu_m(\mathbf{S}) = \frac{1}{2}\mu_m^2(\mathbf{W})$ and hence we reach the conclusion. □

According to Theorem 2, we can see that the smallest eigenvalue of $\mathbf{\Omega}$ has a lower bound depending on the smallest singular value of $\mathbf{W}$. Therefore, when $\mathbf{W}$ is of rank m, the optimal $\mathbf{\Omega}$ is PD and otherwise PSD.

Optimization Procedure

In this section, we discuss how to solve problem (3). In order to study both multi-task classification and regression problems, we adopt the square loss, that is, $l(y_1, y_2) = (y_1 - y_2)^2$. The optimization procedure can be easily extended to other loss functions.

The objective function of the SPATS method with the square loss is formulated as

$$\min_{\mathbf{W},\mathbf{b},\mathbf{\Omega}\succeq\mathbf{0}} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \left(\mathbf{w}_i^T \phi(\mathbf{x}_j^i) + b_i - y_j^i\right)^2 + \frac{\lambda_1}{2}\mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T) + \lambda_2\|\mathbf{\Omega}\|_1. \quad (10)$$

In order to solve problem (10), we use an alternating method which first optimizes problem (10) with respect to $\mathbf{W}$ and $\mathbf{b}$ by fixing $\mathbf{\Omega}$ and then solves problem (10) with respect to $\mathbf{\Omega}$ given the fixed $\mathbf{W}$ and $\mathbf{b}$. In the following, we discuss these two steps in details.

When $\mathbf{\Omega}$ is fixed, the problem with respect to $\mathbf{W}$ and $\mathbf{b}$ is the same as problem (9) in (Zhang and Yeung 2010a) and hence we can obtain an analytical solution or use an SMO-style method to solve it iteratively.

When $\mathbf{W}$ and $\mathbf{b}$ are fixed, the problem with respect to $\mathbf{\Omega}$ is formulated as

$$\min_{\mathbf{\Omega}\succeq\mathbf{0}} \frac{\lambda_1}{2}\mathrm{tr}(\mathbf{\Omega}^{-1}\mathbf{R}) + \lambda_2\|\mathbf{\Omega}\|_1, \quad (11)$$

where $\mathbf{R} = \mathbf{W}^T\mathbf{W}$. Given that $\mathbf{R}$ is PSD, in the following theorem we can prove that problem (11) is convex.

Theorem 3 *When $\mathbf{R}$ is PSD, problem (11) is convex.*

Proof. We only need to prove that $\mathrm{tr}(\mathbf{\Omega}^{-1}\mathbf{R})$ is convex with respect to $\mathbf{\Omega}$ since both the second term in the objective function and the constraint are convex. We define $r(\mathbf{\Omega}) = \mathrm{tr}(\mathbf{\Omega}^{-1}\mathbf{R})$. For any PSD matrix $\mathbf{P} \in \mathbb{R}^{m\times m}$ and any $\alpha \in [0,1]$, we can easily have $(1-\alpha)\mathbf{I} + \alpha\mathbf{P}^{-1} \succeq ((1-\alpha)\mathbf{I} + \alpha\mathbf{P})^{-1}$ since the ith largest eigenvalue of $(1-\alpha)\mathbf{I} + \alpha\mathbf{P}^{-1}$, which is equal to $1-\alpha+\alpha\mu_{m-i}(\mathbf{P})^{-1}$, is larger than that of $((1-\alpha)\mathbf{I} + \alpha\mathbf{P})^{-1}$, which is equal to $(1-\alpha+\alpha\mu_{m-i}(\mathbf{P}))^{-1}$, and those two matrices share the same eigenvectors. By letting $\mathbf{P} = \mathbf{\Omega}_2^{-\frac{1}{2}}\mathbf{\Omega}_1\mathbf{\Omega}_2^{-\frac{1}{2}}$ where $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ are PSD and then left- and right-multiplying the inequality by $\mathbf{\Omega}_2^{-\frac{1}{2}}$, we can get $\alpha\mathbf{\Omega}_1^{-1} + (1-\alpha)\alpha\mathbf{\Omega}_2^{-1} \succeq (\alpha\mathbf{\Omega}_1 + (1-\alpha)\mathbf{\Omega}_2)^{-1}$. Then we have

$$\begin{aligned}&\alpha r(\mathbf{\Omega}_1) + (1-\alpha)r(\mathbf{\Omega}_2) - r(\alpha\mathbf{\Omega}_1 + (1-\alpha)\mathbf{\Omega}_2)\\ =&\mathrm{tr}\left(\left(\alpha\mathbf{\Omega}_1^{-1} + (1-\alpha)\alpha\mathbf{\Omega}_2^{-1} - (\alpha\mathbf{\Omega}_1 + (1-\alpha)\mathbf{\Omega}_2)^{-1}\right)\mathbf{R}\right) \geq 0,\end{aligned}$$

where the inequality holds since the trace of the product between two PSD matrices is nonnegative. Based on the definition of convex functions, we reach the conclusion. □

Based on Theorem 3, problem (11) is convex since $\mathbf{R} = \mathbf{W}^T\mathbf{W}$ is PSD and hence we can use the FISTA algorithm (Beck and Teboulle 2009) to solve problem (11). The FISTA algorithm aims to minimize a combination of two convex function as: $\min_{\mathbf{\Theta}\in\mathcal{C}_\theta} f(\mathbf{\Theta}) + h(\mathbf{\Theta})$, where $\mathcal{C}_\theta$ defines a set of constraints on $\mathbf{\Theta}$, $f(\mathbf{\Theta})$ is a differentiable Lipschitz function, and $h(\mathbf{\Theta})$ can be non-smooth. Then we define

$$Q_L(\mathbf{\Theta}, \hat{\mathbf{\Theta}}) = f(\hat{\mathbf{\Theta}}) + \bigtriangledown_{\mathbf{\Theta}} f(\hat{\mathbf{\Theta}})^T(\mathbf{\Theta} - \hat{\mathbf{\Theta}}) + \frac{L}{2}\mathcal{D}(\mathbf{\Theta}, \hat{\mathbf{\Theta}}) + h(\mathbf{\Theta}),$$

where $\bigtriangledown_{\mathbf{\Theta}} f(\hat{\mathbf{\Theta}})$ denotes the derivative of $f(\cdot)$ with respect to $\mathbf{\Theta}$ at $\mathbf{\Theta} = \hat{\mathbf{\Theta}}$ and $\mathcal{D}(\cdot,\cdot)$ measures the Euclidean distance between variables. We define $q_L(\hat{\mathbf{\Theta}}) = \arg\min_{\mathbf{\Theta}} Q_L(\mathbf{\Theta}, \hat{\mathbf{\Theta}})$.

In order to apply the FISTA algorithm to problem (11), we set $\mathbf{\Theta} = \{\mathbf{\Omega}\}$, $f(\mathbf{\Theta}) = \frac{\lambda_1}{2}\mathrm{tr}(\mathbf{\Omega}^{-1}\mathbf{R})$, $h(\mathbf{\Theta}) = \lambda_2\|\mathbf{\Omega}\|_1$, and $\mathcal{C}_\theta = \{\mathbf{\Omega} \succeq \mathbf{0}\}$. In the FISTA algorithm, we need to minimize $Q_L(\mathbf{\Theta}, \hat{\mathbf{\Theta}})$, which is formulated as

$$\min_{\mathbf{\Omega}} \frac{L}{2}\left\|\mathbf{\Omega} - \left(\hat{\mathbf{\Omega}} - \frac{1}{L}\bigtriangledown_{\mathbf{\Omega}} f(\hat{\mathbf{\Omega}})\right)\right\|_F^2 + \lambda_2\|\mathbf{\Omega}\|_1, \quad (12)$$

where $\bigtriangledown_{\mathbf{\Omega}} f(\hat{\mathbf{\Omega}}) = -\frac{\lambda_1}{2}\mathbf{\Omega}^{-1}\mathbf{R}\mathbf{\Omega}^{-1}$. The PSD constraint on $\mathbf{\Omega}$ is satisfied according to the numerical determination of L in step 6 of the FISTA algorithm and hence problem (12) does not include such constraint. Problem (12) is a Lasso problem with an analytical solution as $[\mathbf{\Omega}]_{ij} = \kappa\left(\left[\hat{\mathbf{\Omega}} - \frac{1}{L}\bigtriangledown_{\mathbf{\Omega}} f(\hat{\mathbf{\Omega}})\right]_{ij}, \frac{\lambda_2}{L}\right)$, where $[\cdot]_{ij}$ returns the (i,j)th element of a matrix, $\mathrm{sgn}(\cdot)$ defines the sign function, $|\cdot|$ gives the absolute value when the argument is a scalar, and $\kappa(a,b) = \begin{cases} 0 & \text{if } |a| \leq b \\ \mathrm{sgn}(a)\,(|a| - b) & \text{otherwise} \end{cases}$ is the soft-thresholding operator.

The above two steps will iterate until convergence. In our experiments, we find that the convergence of the alternating method is very fast compared with using the FISTA algorithm to solve problem (3) directly. Moreover, no matter whatever the function $g(\cdot)$ is, the learning of $\mathbf{W}$ and $\mathbf{b}$ keeps unchanged, which makes the implementation of the alternating method partially re-useable, which can speedup the implementation of different methods under the proposed framework.

Discussion

Under the proposed framework, we can devise more concrete learning models. For example, in some case, outlier tasks that have no relations to other tasks may exist among those tasks under investigation. In such situation, if the outlier tasks are still assumed to be helpful to other tasks, then they will definitely cause the performance of other tasks to deteriorate. Therefore, it is better to detect such outlier tasks during the learning process of multi-task learning. Under this setting, we expect one or more columns except the diagonal entries in $\mathbf{\Omega}$ is a zero vector and hence the function $g(\cdot)$ in problem (1) can be defined as $g(\mathbf{\Omega}) = \sum_{i=1}^{m}\|\boldsymbol{\omega}_i^c\|_2$, where $\boldsymbol{\omega}_i^c \in \mathbb{R}^{m-1}$ is the ith column of $\mathbf{\Omega}$ by excluding the ith entry in that column. Another advantage of the proposed framework (1) is that we can combine two or more instances of $g(\cdot)$ to form a new regularizer, which can aggregate the characteristics of individual regularizers, for $\mathbf{\Omega}$.

Experiments

In this section, we empirically test the performance of the proposed SPATS method.

We compare the proposed SPATS model with state-of-the-art models including the STL method which is the single-task model with the square loss or equivalent the ridge regression model, the MTL_Ω method (Evgeniou and Pontil 2004)

which is the multi-task model with a given $\mathbf{\Omega}$, the MTL-TNR method (Pong et al. 2010) which is the multi-task model with trace norm regularization, the MTL-STNR method (Argyriou, Evgeniou, and Pontil 2006) which is the multi-task model with the squared trace norm regularization, the MTL-CNR method (Jacob, Bach, and Vert 2008) which is the multi-task model with the cluster norm regularization, and the MTL-gLasso method (Zhang and Schneider 2010) which is the multi-task model with graphical Lasso to enforce the sparsity of the inverse of $\mathbf{\Omega}$.

Experiments on Synthetic Data

In this section, we test the SPATS method on some synthetic datasets. To begin with, we generate the sparse task covariance $\mathbf{\Omega}_*$ as follows. We first generate an $m \times m$ matrix $\mathbf{U}_\Omega^*$ with each of its entries sampling from the standard normal distribution. Each row in $\mathbf{U}_\Omega^*$ has 40% probability to be selected and 80% of the entries in each selected row of $\mathbf{U}_\Omega^*$ will set to be 0 with equal probability. After that, we can obtain a sparse $\mathbf{\Omega}_*$ as $\mathbf{\Omega}_* = \mathbf{U}_\Omega^*(\mathbf{U}_\Omega^*)^T$. After obtaining $\mathbf{\Omega}_*$, we can generate the parameter matrix $\mathbf{W}_* \in \mathbb{R}^{d\times m}$ with each of its rows sampling from a multivariate normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Omega}_*)$ independently. Each entry in the data matrix for the ith task, $\mathbf{X}_i \in \mathbb{R}^{d\times n_i}$, is sampled according to the standard normal distribution. Then the vector of labels in the ith task, $\mathbf{y}_i \in \mathbb{R}^{n_i}$, is generated as $\mathbf{y}_i = \mathbf{X}_i^T\mathbf{w}_i^* + \boldsymbol{\xi}_i$ where $\mathbf{w}_i^*$ is the ith column of $\mathbf{W}_*$ and $\boldsymbol{\xi}_i \in \mathbb{R}^{n_i}$ contains Gaussian noises each of which is sampled from $\mathcal{N}(0, 0.5)$.

We adopt two settings for (m, d), i.e., $(20, 10)$ and $(40, 10)$, to generate two synthetic datasets. For each task, we generate 50 data points for training, 50 data points for validation to choose the regularization parameter with the set of candidate values as $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}$, 100 data points for testing. Each setting will repeat for 10 times and the mean as well as the standard deviation is reported. The performance measure is the normalized mean squared error (NMSE) which equals the mean squared error divided by the variance of the ground truth. The experimental results are shown in Table 1 and the best results under the significant t-test with 95% confidence are shown in bold. According to the results, we can see that on the synthetic datasets, the MTL$_\Omega$ model performs worse than the STL model and one reason for that is that the assumption adopted by the MTL$_\Omega$ model does not hold in this dataset. Other multi-task models outperform the STL model and among them, the SPATS model has the best performance.

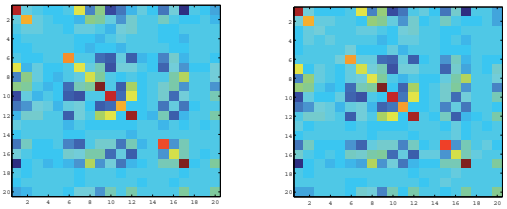
In Fig. 1, we plot the ground-truth of the task covariance in the first dataset as well as its estimation learned by the SPATS model. We can see that the difference between the ground-truth and the estimation is very small, which demonstrates the effectiveness of the SPATS method on the synthetic dataset to recover the sparse task relations.

Experiments on Real-World Datasets

Five benchmark datasets, including School, Parkinson, Sentiment, Landmine and MHC-I datasets, are used in the experiment. The School dataset contains examination scores of 15362 students from 139 secondary schools in London during years 1985, 1986 and 1987, hence, there are totally 139 tasks. The input consists of the year of the examination, four school-specific and three student-specific attributes. Following (Evgeniou, Micchelli, and Pontil 2005), we replace each categorical attribute with one binary variable for each possible attribute value and as a result, there are 27 input attributes. The Parkinson dataset is used to predict the Parkinson's disease symptom score for patients based on 16 biomedical features. The Parkinson dataset contains 5,875 observations for 42 patients and hence predicting the symptom score for each patient is treated as a regression task, leading to 42 regression tasks with the number of instances for each task ranging from 101 to 168. In the Sentiment dataset, there are four different products (tasks) from Amazon.com: books, DVDs, electronics, and kitchen appliances. For each task, there are 1000 positive and 1000 negative data points corresponding to positive and negative reviews, respectively and the goal is to classify the reviews of some products into two classes: positive and negative reviews. Each data point has 473856 feature dimensions. The Landmine dataset contains examples collected from 29 landmine fields. Each example contains nine numeric features and each of the 29 tasks is a binary classification problem to predict landmines (positive class) or clutters (negative class). The number of data points in each task varies from 445 to 690. The dataset is highly imbalance against the positive class. The MHC-I dataset contains binding affinities of various peptides with different MHC-I molecules and the goal is to predict whether a peptide binds a molecule. Each MHC-I molecule is considered as a

Table 1: Experimental results in terms of mean±standard deviation on the two synthetic datasets

Method	Synthetic Data 1	Synthetic Data 2
STL	0.1033±0.0118	0.1029±0.0263
MTL$_\Omega$	0.1287±0.0230	0.1173±0.0298
MTL-TNR	0.0766±0.0066	0.0401±0.0056
MTL-STNR	0.0763±0.0033	0.0691±0.0069
MTL-CNR	0.0765±0.0073	0.0382±0.0042
MTL-gLasso	0.1013±0.0149	0.0930±0.0157
SPATS	**0.0646±0.0052**	**0.0280±0.0042**



(a) $\mathbf{\Omega}_*$ (b) $\tilde{\mathbf{\Omega}}$

Figure 1: Comparison between the true task covariances and the estimations by the SPATS method on the first synthetic data. Here $\mathbf{\Omega}_*$ denotes the ground-truth and $\tilde{\mathbf{\Omega}}$ is the corresponding estimation.

Table 2: Experimental results on five real-world datasets. ↑ after the name of each dataset means that a larger value indicates better performance and ↓ implies that a smaller value corresponds to better performance.

Method	School↓	Parkinson↓	Sentiment↑	Landmine↑	MHC-I↑
STL	1.1453±0.0599	1.1327±0.0866	0.8303±0.0098	0.6854±0.0261	0.6677±0.0234
MTL_Ω	1.1004±0.0631	1.0828±0.0460	0.8237±0.0209	0.7015±0.0244	0.6879±0.0227
MTL-TNR	1.0247±0.0879	1.0744±0.0415	**0.8764±0.0078**	0.7236±0.0249	0.7076±0.0203
MTL-STNR	0.9048±0.0981	1.0207±0.0162	**0.8808±0.0085**	**0.7496±0.0287**	0.6943±0.0230
MTL-CNR	1.1040±0.0474	1.0203±0.0119	–	0.7228±0.0204	0.7070±0.0084
MTL-gLasso	1.1559±0.0538	1.1182±0.0617	0.8267±0.0279	0.7286±0.0202	**0.7232±0.0297**
SPATS	0.9110±0.0291	**0.9894±0.0229**	0.8576±0.0103	**0.7420±0.0121**	**0.7299±0.0240**

task and there are 35 tasks. The number of instances per task varies from 59 to 197 and the the dataset is biased against the positive class.

For all the datasets, we randomly choose 20% data for training, 20% for validation, and the rest for testing. The random split is repeated for 10 times and the mean as well as the standard deviation is reported for each dataset. Here the validation set is to choose the regularization parameters in all of the methods in comparison and the set of the candidate values for the regularization parameters is $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}$. Similar to the synthetic datasets, the performance measure used for multi-task regression problems is the NMSE and that for multi-task classification problems is the area under curve (AUC) for the receiver operating characteristic curve since some datasets (e.g., the Landmine and MHC-I data) are imbalance. So in the regression problems, the smaller the value reported, the better the performance, but for classification problems, a larger value indicates better performance.

We present the experimental results in Table 2, where the best results under the significant t-test with 95% confidence are shown in bold. Since the MTL-CNR method proposed in (Jacob, Bach, and Vert 2008) is a linear method which cannot handle high-dimensional text data, we do not include it in the comparison on the Sentiment dataset. According to the results, all the multi-task methods have better performance than the single-task method. Moreover, on the four datasets including the School, Parkinson, Landmine, and MHC-I datasets, the SPATS method achieves the best or nearly the best performance among all the methods in comparison. On the Sentiment dataset, the situation is slightly different. we can see that the MTL-TNR and MTL-STNR methods have the best performance and the SPATS method performs slightly worse than them. One reason for that is when the number of tasks is very small, each task will not receive enough knowledge transferred from other tasks if task relations are assumed to be sparse and hence under this situation, learning dense task relations is a better choice. Based on this observation, we believe that learning sparse task relations is a good strategy when there are lots of learning tasks.

Analysis on Learned Task Covariances

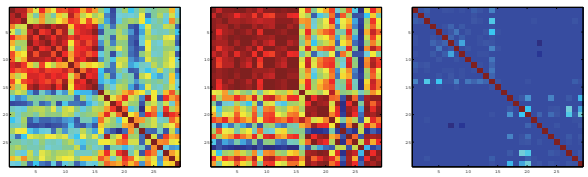
In this section, we present some analysis on the learned task covariances.

In order to study the sparse task covariance learned in the SPATS method, we record in Table 3 the average sparsity of the learned $\mathbf{\Omega}$ in the SPATS method on all the datasets except the Sentiment dataset which has only 4 tasks. According to Table 3, the learned $\mathbf{\Omega}$ is very sparse and this observation together with the good performance of our proposed methods reported in the last section verifies the motivation that learning sparse task relations is useful when the number of tasks is large.

Table 3: The average sparsity (in percentage) of the learned $\mathbf{\Omega}$ in the SPATS method on the four datasets.

School	Parkinson	Landmine	MHC-I
73.79%	77.62%	79.19%	81.14%

We plot in Figure 2 the task correlation matrices derived from the learned $\mathbf{\Omega}$'s in the SPATS, MTL-TNR, and MTL-STNR methods on the Landmine dataset, where darker colors indicate values closer to zero. According to Figure 2, we can see that the learned $\mathbf{\Omega}$'s in the SPATS method have more entries close to zero, which again verifies that the SPATS method can learn sparse task relations.



(a) MTL-TNR (b) MTL-STNR (c) SPATS

Figure 2: The comparison of the task correlation matrices learned by the MTL-TNR, MTL-STNR and SPATS methods on the Landmine dataset.

Conclusion

In this paper, we investigate the learning of sparse task relations. Based on the proposed framework, which can accommodate several state-of-the-art multi-task models as special instances, for multi-task learning, we devise the SPATS method to learn the sparse task relations. Through the experiments, we have shown that when the number of tasks is large, learning sparse task relations is helpful to improve

the performance. In our future study, we are interested in devising more multi-task models to learn different sparsity patterns in Ω such as the outlier task case discussed before.

Acknowledgement

This work is supported by National Grant Fundamental Research (973 Program) of China under Project 2014CB340304, Hong Kong CERG projects 16211214, 16209715 and 16244616, and Natural Science Foundation of China under Project 61305071.

References

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

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2006. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19*, 41–48.

Argyriou, A.; Micchelli, C. A.; and Pontil, M. 2009. When is there a representer theorem? vector versus matrix regularizers. *Journal of Machine Learning Research* 10:2507–2529.

Baxter, J. 1997. A Bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning* 28(1):7–39.

Beck, A., and Teboulle, M. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences* 2(1):183–202.

Bonilla, E.; Chai, K. M. A.; and Williams, C. 2007. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems 20*, 153–160.

Caruana, R. 1997. Multitask learning. *Machine Learning* 28(1):41–75.

Chen, J.; Tang, L.; Liu, J.; and Ye, J. 2009. A convex formulation for learning shared structures from multiple tasks. In *Proceedings of the 26th International Conference on Machine Learning*, 137–144.

Evgeniou, T., and Pontil, M. 2004. Regularized multi-task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 109–117.

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

Han, L., and Zhang, Y. 2015a. Learning multi-level task groups in multi-task learning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*.

Han, L., and Zhang, Y. 2015b. Learning tree structure in multi-task learning. In *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Han, L., and Zhang, Y. 2016. Multi-stage multi-task learning with reduced rank. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

Han, L.; Zhang, Y.; Song, G.; and Xie, K. 2014. Encoding tree sparsity in multi-task learning: A probabilistic framework. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 1854–1860.

Jacob, L.; Bach, F.; and Vert, J.-P. 2008. Clustered multi-task learning: a convex formulation. In *Advances in Neural Information Processing Systems 21*, 745–752.

Jawanpuria, P., and Nath, J. S. 2012. A convex feature learning formulation for latent task structure discovery. In *Proceedings of the 29th International Conference on Machine Learning*.

Kang, Z.; Grauman, K.; and Sha, F. 2011. Learning with whom to share in multi-task feature learning. In *Proceedings of the 28th International Conference on Machine Learning*, 521–528.

Kato, T.; Kashima, H.; Sugiyama, M.; and Asai, K. 2007. Multi-task learning via conic programming. In *Advances in Neural Information Processing Systems 20*, 737–744.

Obozinski, G.; Taskar, B.; and Jordan, M. 2006. Multi-task feature selection. Technical report, Department of Statistics, University of California, Berkeley.

Pong, T. K.; Tseng, P.; Ji, S.; and Ye, J. 2010. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization* 20(6):3465–3489.

Rai, P.; Kumar, A.; and Daume, H. 2012. Simultaneously leveraging output and task structures for multiple-output regression. In *Advances in Neural Information Processing Systems 25*, 3185–3193.

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

Zhang, Y., and Schneider, J. G. 2010. Learning multiple tasks with a sparse matrix-normal penalty. In *Advances in Neural Information Processing Systems 23*, 2550–2558.

Zhang, Y., and Yeung, D.-Y. 2010a. A convex formulation for learning task relationships in multi-task learning. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 733–742.

Zhang, Y., and Yeung, D.-Y. 2010b. Transfer metric learning by learning task relationships. In *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1199–1208.

Zhang, Y., and Yeung, D.-Y. 2014. A regularization approach to learning task relationships in multitask learning. *ACM Transactions on Knowledge Discovery from Data* 8(3):article 12.

Zhang, Y. 2013. Heterogeneous-neighborhood-based multi-task local learning algorithms. In *Advances in Neural Information Processing Systems 26*.