A Convex Formulation for Learning Shared Structures from Multiple Tasks

Jianhui Chen JIANHUI.CHEN@ASU.EDU
Lei Tang L.TANG@ASU.EDU
Jun Liu J.LIU@ASU.EDU
Jieping Ye JIEPING.YE@ASU.EDU
Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287, USA

Abstract

Multi-task learning (MTL) aims to improve generalization performance by learning multiple related tasks simultaneously. In this paper, we consider the problem of learning shared structures from multiple related tasks. We present an improved formulation (iASO) for multi-task learning based on the non-convex alternating structure optimization (ASO) algorithm, in which all tasks are related by a shared feature representation. We convert iASO, a non-convex formulation, into a relaxed convex one, which is, however, not scalable to large data sets due to its complex constraints. We propose an alternating optimization (cASO) algorithm which solves the convex relaxation efficiently, and further show that cASO converges to a global optimum. In addition, we present a theoretical condition, under which cASO can find a globally optimal solution to iASO. Experiments on several benchmark data sets confirm our theoretical analysis.

1. Introduction

The problem of multi-task learning (Caruana, 1997) has recently received broad attention in areas such as machine learning, data mining, computer vision, and bioinformatics (Heisele et al., 2001; Ando & Zhang, 2005; Ando, 2007; Xue et al., 2007; Yu et al., 2007; Argyriou et al., 2008). Multi-task learning aims to improve generalization performance by learning from multiple related tasks. This can be achieved by learning the tasks simultaneously, and meanwhile exploiting their intrinsic relatedness, based on which the informative domain knowledge is allowed to be shared across the tasks, thus facilitating individual task learning. It is particularly desirable to share such knowledge across the tasks when there are a number of related tasks but only limited training data for each one is available.

Multi-task learning has been investigated by many researchers from different perspectives such as sharing hidden units of neural networks among similar tasks (Caruana, 1997; Baxter, 2000), modeling task relatedness using the common prior distribution in hierarchical Bayesian models (Bakker & Heskes, 2003; Schwaighofer et al., 2004; Yu et al., 2005; Zhang et al., 2005), learning the parameters of Gaussian Process covariance from multiple tasks (Lawrence & Platt, 2004), extending kernel methods and regularization networks to multi-task learning (Evgeniou et al., 2005), and multi-task learning with clustered tasks (Jacob et al., 2008). Recently, there is growing interest in studying multi-task learning in the context of feature learning and selection (Ando & Zhang, 2005; Obozinski et al., 2006; Amit et al., 2007; Argyriou et al., 2008). Specifically, Ando and Zhang (2005) propose the alternating structure optimization (ASO) algorithm to learn the predictive structure from multiple tasks. In ASO, a separate linear classifier is trained for each of the tasks and dimension reduction is applied on the predictor space, finding low-dimensional structures with the highest predictive power. This framework has been applied successfully in several applications (Ando, 2007; Quattoni et al., 2007). However, it is non-convex and the alternating optimization procedure is not guaranteed to find a global optimum (Ando & Zhang, 2005).

In this paper, we consider the problem of learning a shared structure from multiple related tasks following the approach in (Ando & Zhang, 2005). We present an improved ASO formulation (called iASO) using a novel regularizer. The improved formulation is non-convex;

Appearing in *Proceedings of the 26^{th} International Conference on Machine Learning*, Montreal, Canada, 2009.

we show that it can be converted into a (relaxed) convex formulation, whose globally optimal solution approximates the one to iASO. However, this convex formulation is not scalable to large data sets due to its positive semidefinite constraints. We propose a convex alternating structure optimization (called cASO) algorithm to solve the convex relaxation efficiently, in which the optimization variables are updated iteratively. The proposed cASO algorithm is similar in spirit to the block coordinate descent method (Bertsekas, 1999) in unconstrained optimization, and it can be shown to converge to a global optimum of the convex relaxation. In addition, we present a theoretical condition, under which cASO finds a globally optimal solution to iASO. We have performed experiments on benchmark data sets. The reported experimental results are consistent with our theoretical analysis. Results also demonstrate the effectiveness of the proposed multi-task learning algorithm.

Notations Denote $\mathbb{N}_n = \{1, \cdots, n\}$. Let $\mathbb{S}_+^d$ ($\mathbb{S}_{++}^d$) be the subset of positive semidefinite (positive definite) matrices. Denote $A \preceq B$ if and only if $B - A$ is positive semidefinite. Let $\mathrm{tr}(X)$ be the trace, and X^{-1} be the inverse of a matrix X.

2. Multi-Task Learning Framework

Assume that we are given m supervised (binary-class) learning tasks. Each of the learning tasks is associated with a predictor f_ℓ and training data $\{(x_1^\ell, y_1^\ell), \cdots, (x_{n_\ell}^\ell, y_{n_\ell}^\ell)\} \subset \mathbb{R}^d \times \{-1, 1\}$ ($\ell \in \mathbb{N}_m$). We focus on linear predictors $f_\ell(x) = u_\ell^\mathsf{T} x$, where u_ℓ is the weight vector for the ℓth task.

Ando and Zhang (2005) propose the alternating structure optimization (ASO) algorithm for learning predictive functional structures from multiple related tasks, that is, learning all m predictors $\{f_\ell\}_{\ell=1}^m$ simultaneously by exploiting a shared feature space in a simple linear form of low-dimensional feature map Θ across the m tasks. Formally, the prediction function f_ℓ can be expressed as:

$$f_\ell(x) = u_\ell^\mathsf{T} x = w_\ell^\mathsf{T} x + v_\ell^\mathsf{T} \Theta x, \tag{1}$$

where the structure parameter Θ takes the form of an $h \times d$ matrix with orthonormal rows, i.e., $\Theta\Theta^\mathsf{T} = I$, and u_ℓ, w_ℓ, and v_ℓ are the weight vectors for the full feature space, the high-dimensional one, and the shared low-dimensional one, respectively. Mathematically, ASO can be formulated as the following optimization problem:

$$\min_{\{u_\ell, v_\ell\}, \Theta\Theta^\mathsf{T} = I} \sum_{\ell=1}^m \left(\frac{1}{n_\ell} \sum_{i=1}^{n_\ell} L(u_\ell^\mathsf{T} x_i^\ell, y_i^\ell) + \alpha \|w_\ell\|^2 \right), \tag{2}$$

where L is the loss function, $\|w_\ell\|^2$ is the regularization term ($w_\ell = u_\ell - \Theta^\mathsf{T} v_\ell$) controlling the task relatedness among m tasks, and α is the pre-specified parameter.

The optimization problem in Eq. (2) is non-convex due to its orthonormal constraints and the regularization term in terms of u_ℓ, v_ℓ, and Θ (the loss function L is assumed to be convex). We present an improved ASO formulation (called iASO) given by:

$$(\mathbf{F_0}) \quad \min_{\{u_\ell, v_\ell\}, \Theta\Theta^\mathsf{T} = I} \sum_{\ell=1}^m \left(\frac{1}{n_\ell} \sum_{i=1}^{n_\ell} L(u_\ell^\mathsf{T} x_i^\ell, y_i^\ell) + g_\ell(u_\ell, v_\ell, \Theta) \right), \tag{3}$$

where $g_\ell(u_\ell, v_\ell, \Theta)$ is the regularization function defined as:

$$g_\ell(u_\ell, v_\ell, \Theta) = \alpha \|u_\ell - \Theta^\mathsf{T} v_\ell\|^2 + \beta \|u_\ell\|^2. \tag{4}$$

The regularization functions in Eq. (4) controls the task relatedness (via the first component) as well as the complexity of the predictor functions (via the second component) as commonly used in traditional regularized risk minimization formulation for supervised learning. Note that α and β are pre-specified coefficients, indicating the importance of the corresponding regularization component. For simplicity, we use the same α and β parameters for all tasks. However, the discussion below can be easily extended to the case where α and β are different for different tasks.

The iASO formulation (F_0 in Eq. (3)) subsumes several multi-task learning algorithms as special cases: it reduces to the ASO algorithm in Eq. (2) by setting $\beta = 0$ in Eq. (4); and it reduces to m independent support vector machines (SVM) by setting $\alpha = 0$. It is worth noting that F_0 is non-convex. In the next section, we convert F_0 into a (relaxed) convex formulation, which admits a globally optimal solution.

3. A Convex Multi-Task Learning Formulation

In this section, we consider a convex relaxation of the non-convex problem F_0 (iASO) in Eq. (3).

The optimal $\{v_\ell\}_{\ell=1}^m$ to F_0 can be expressed in the form of a function on Θ and $\{u_\ell\}_{\ell=1}^m$. Let $U = [u_1, \cdots, u_m] \in \mathbb{R}^{d \times m}$ and $V = [v_1, \cdots, v_m] \in \mathbb{R}^{h \times m}$. It can be verified that $g_\ell(u_\ell, v_\ell, \Theta)$ in Eq. (4) is minimized when $v_\ell = \Theta u_\ell$ ($\ell \in \mathbb{N}_m$), and hence $V = \Theta U$. Therefore we can denote

$$\begin{aligned} G_0(U, \Theta) &= \min_V \sum_{\ell=1}^m g_\ell(u_\ell, v_\ell, \Theta) \\ &= \alpha \ \mathrm{tr}\left(U^\mathsf{T} \left((1+\eta)I - \Theta^\mathsf{T}\Theta\right) U\right), \end{aligned} \tag{5}$$

where $\eta = \beta/\alpha > 0$. Moreover, it can be verified that the following equality holds

$$(1+\eta)I - \Theta^\mathsf{T}\Theta = \eta\,(1+\eta)\left(\eta I + \Theta^\mathsf{T}\Theta\right)^{-1}. \quad (6)$$

We can then reformulate $G_0(U,\Theta)$ in Eq. (5) into an equivalent form given by

$$G_1(U,\Theta) = \alpha\,\eta\,(1+\eta)\,\mathrm{tr}\left(U^\mathsf{T}\left(\eta I + \Theta^\mathsf{T}\Theta\right)^{-1}U\right). \quad (7)$$

Since the loss term in Eq. (3) is independent of $\{v_\ell\}_{\ell=1}^m$, F_0 can be equivalently transformed into the following optimization problem F_1 with optimization variables Θ and U as:

$$(\mathbf{F_1})\ \min_{U,\Theta\Theta^\mathsf{T}=I}\sum_{\ell=1}^m\left(\frac{1}{n_\ell}\sum_{i=1}^{n_\ell}L(u_\ell^\mathsf{T}x_i^\ell, y_i^\ell)\right) + G_1(U,\Theta), \quad (8)$$

where $G_1(U,\Theta)$ is defined in Eq. (7).

3.1. Convex Relaxation

The orthonormality constraints in Eq. (8) is non-convex, so is the optimization problem F_1. We propose to convert F_1 into a convex formulation by relaxing its feasible domain into a convex set. Let the set $\mathcal{M}_e$ be defined as:

$$\mathcal{M}_e = \left\{M_e \mid M_e = \Theta^\mathsf{T}\Theta,\ \Theta\Theta^\mathsf{T} = I,\ \Theta \in \mathbb{R}^{h\times d}\right\}. \quad (9)$$

It has been shown in (Overton & Womersley, 1993) that the convex hull (Boyd & Vandenberghe, 2004) of $\mathcal{M}_e$ can be precisely expressed as the convex set $\mathcal{M}_c$ given by

$$\mathcal{M}_c = \left\{M_c \mid \mathrm{tr}(M_c) = h,\ M_c \preceq I,\ M_c \in \mathbb{S}_+^d\right\}, \quad (10)$$

and each element in $\mathcal{M}_e$ is referred to as an extreme point of $\mathcal{M}_c$. Since $\mathcal{M}_c$ consists of all convex combinations of the elements in $\mathcal{M}_e$, $\mathcal{M}_c$ is the smallest convex set that contains $\mathcal{M}_e$, and hence $\mathcal{M}_e \subseteq \mathcal{M}_c$.

To convert the non-convex problem F_1 into a convex formulation, we replace $\Theta^\mathsf{T}\Theta$ with M in Eq. (8), and relax the (feasible) problem domain to a convex set based on the relationship between $\mathcal{M}_e$ and $\mathcal{M}_c$ presented above; this results in a convex formulation F_2 defined as:

$$(\mathbf{F_2})\ \min_{U,M}\ \sum_{\ell=1}^m\left(\frac{1}{n_\ell}\sum_{i=1}^{n_\ell}L(u_\ell^\mathsf{T}x_i^\ell, y_i^\ell)\right) + G_2(U,M)$$
$$\text{subject to}\quad \mathrm{tr}(M) = h,\ M \preceq I,\ M \in \mathbb{S}_+^d, \quad (11)$$

where $G_2(U,M)$ is defined as:

$$G_2(U,M) = \alpha\,\eta\,(1+\eta)\,\mathrm{tr}\left(U^\mathsf{T}\left(\eta I + M\right)^{-1}U\right). \quad (12)$$

Note that F_2 is a convex relaxation of F_1 since the optimal M to F_2 is not guaranteed to occur at the extreme points of $\mathcal{M}_c$. The optimal Θ to F_1 can be approximated using the first h eigenvectors (corresponding to the largest h eigenvalues) of the optimal M computed from F_2.

The convexity in F_2 can be readily verified. We add variables $\{t_\ell\}_{\ell=1}^m$ and enforce $u_\ell^\mathsf{T}(\eta I + M)^{-1}u_\ell \le t_\ell$ $(\forall \ell \in \mathbb{N}_m)$; it follows from the Schur complement Lemma (Golub & Loan, 1996) that we rewrite F_2 as:

$$(\mathbf{F_3})\ \min_{U,M,\{t_\ell\}}\sum_{\ell=1}^m\left(\frac{1}{n_\ell}\sum_{i=1}^{n_\ell}L(u_\ell^\mathsf{T}x_i^\ell, y_i^\ell)\right) + \alpha\eta(1+\eta)\sum_{\ell=1}^m t_\ell$$
$$\text{subject to}\quad \begin{pmatrix} \eta I + M & u_\ell \\ u_\ell^\mathsf{T} & t_\ell \end{pmatrix} \succeq 0,\ \forall \ell \in \mathbb{N}_m,$$
$$\mathrm{tr}(M) = h,\ M \preceq I,\ M \in \mathbb{S}_+^d. \quad (13)$$

Given that the loss function L is convex, the optimization problem F_3 is convex. However, it is not scalable to high-dimensional data (with a large d) due to its positive semidefinite constraints. If L is the SVM hinge loss, F_3 is a semidefinite program (SDP) (Boyd & Vandenberghe, 2004). Note that many off-the-shelf optimization solvers such as SeDuMi[1] can be used for solving SDP, which can only handle several hundreds of optimization variables.

4. Convex Alternating Structure Optimization

The optimization problem F_3 in Eq. (13) is convex, thus resulting in a globally optimal solution. However, this formulation does not scale well in practice. In this section, we propose a convex alternating structure optimization (called *c*ASO) algorithm to efficiently solve the optimization problem F_2 in Eq. (11). *c*ASO is similar to the block coordinate descent method (Bertsekas, 1999), in which one of the two optimization variables (U and M) is fixed, while the other one can be optimized in terms of the fixed one. The pseudo-code of the *c*ASO algorithm is presented in Algorithm 1.

4.1. Computation of U for a Given M

For a given M, the optimal U can be computed by solving the following problem:

$$\min_U\ \sum_{\ell=1}^m\left(\frac{1}{n_\ell}\sum_{i=1}^{n_\ell}L(u_\ell^\mathsf{T}x_i^\ell, y_i^\ell) + \hat{g}(u_\ell)\right), \quad (14)$$

where $\hat{g}(u_\ell) = \alpha\,\eta\,(1+\eta)\,\mathrm{tr}\left(u_\ell^\mathsf{T}\left(\eta I + M\right)^{-1}u_\ell\right)$. Given any convex loss function L, we can verify that

[1] http://sedumi.ie.lehigh.edu/

Algorithm 1 cASO for multi-task learning

Input: $\{(x_i^\ell, y_i^\ell)\}$, $i \in \mathbb{N}_{n_\ell}$, $\ell \in \mathbb{N}_m$, $h \in \mathbb{N}$.
Output: U, V, and Θ.
Parameter: α and β.
Initialize M subject to the constraints in Eq. (11).
repeat
  Update U via Eq. (14).
  Compute the SVD of $U = P_1 \Sigma P_2^\mathsf{T}$.
  Update M via Eq. (17) and Theorem 4.1.
until convergence criterion is satisfied.
Construct Θ using the top h eigenvectors of M.
Construct V as $V = \Theta U$.
Return U, V and Θ.

the objective function in Eq. (14) is strictly convex, and hence this optimization problem admits a unique minimizer. Note that if L is the SVM hinge loss, the problem in Eq. (14) decouples into m independent quadratic programs (QP), thus the optimal $\{u_\ell\}_{\ell=1}^m$ can be computed separately.

4.2. Computation of M for a Given U

For a given U, the optimal M can be computed by solving the following problem:

$$\begin{aligned} \min_{M} \quad & \operatorname{tr}\left(U^\mathsf{T}\left(\eta I + M\right)^{-1} U\right) \\ \text{subject to} \quad & \operatorname{tr}(M) = h, M \preceq I, M \in \mathbb{S}_+^d. \end{aligned} \tag{15}$$

This problem can be recast into an SDP problem, which is computationally expensive to solve. We propose an efficient approach to find its optimal solution, in which we only solve a simple eigenvalue optimization problem.

4.2.1. Efficient Computation of M

Given any $U \in \mathbb{R}^{d\times m}$ in Eq. (15), let $U = P_1 \Sigma P_2^\mathsf{T}$ be its SVD (Golub & Loan, 1996), where $P_1 \in \mathbb{R}^{d\times d}$ and $P_2 \in \mathbb{R}^{m\times m}$ are orthogonal, and $\operatorname{rank}(U) = q$. Note that in general $q \le m \le d$. It follows that

$$\begin{aligned} \Sigma = \operatorname{diag}(\sigma_1, \cdots, \sigma_m) \in \mathbb{R}^{d\times m}, \\ \sigma_1 \ge \cdots \ge \sigma_q > 0 = \sigma_{q+1} = \cdots = \sigma_m. \end{aligned} \tag{16}$$

We consider the following convex optimization problem (Boyd & Vandenberghe, 2004):

$$\begin{aligned} \min_{\{\gamma_i\}_{i=1}^q} \quad & \sum_{i=1}^q \frac{\sigma_i^2}{\eta + \gamma_i} \\ \text{subject to} \quad & \sum_{i=1}^q \gamma_i = h,\ 0 \le \gamma_i \le 1,\ \forall i \in \mathbb{N}_q, \end{aligned} \tag{17}$$

where $\{\sigma_i\}$ are the singular values of U defined in Eq. (16). The optimal $\{\gamma_i^*\}_{i=1}^q$ to Eq. (17) must satisfy $\gamma_1^* \ge \gamma_2^* \cdots \ge \gamma_q^*$. Hence

$$\frac{1}{\eta + \gamma_1^*} \le \frac{1}{\eta + \gamma_2^*} \le \cdots \le \frac{1}{\eta + \gamma_q^*}. \tag{18}$$

This can be verified by contradiction: assuming $\gamma_i^* < \gamma_{i+1}^*$ and given $\sigma_i > \sigma_{i+1}$, we can construct another feasible solution by switching γ_i^* and γ_{i+1}^*, and obtain a smaller objective value in Eq. (17). Note that the problem in Eq. (17) can be solved via many existing algorithms such as the projected gradient descent method (Boyd & Vandenberghe, 2004).

We show that the optimal solution to Eq. (15) can be obtained by solving Eq. (17). We first present the following lemma, which will be useful for proving the theorem followed.

Lemma 4.1. *For any matrix $Z \in \mathbb{S}_+^d$, let $Z = \hat{U}\hat{\Sigma}_z\hat{U}^\mathsf{T}$ be its SVD, where $\hat{U} \in \mathbb{R}^{d\times d}$ is orthogonal, $\hat{\Sigma}_z = \operatorname{diag}(\hat{\sigma}_1, \cdots, \hat{\sigma}_d)$, and $\hat{\sigma}_1 \ge \cdots \ge \hat{\sigma}_d \ge 0$. Let $\{Z_i\}_{i=1}^d$ be the diagonal entries of Z, and $\Pi = \{\pi_1, \cdots, \pi_p\} \subseteq \mathbb{N}_d$ be any integer subset with p distinct elements. Then $\sum_{i=1}^p Z_{\pi_i} \le \sum_{j=1}^p \hat{\sigma}_j$.*

Proof. Denote the i-th row-vector of $\hat{U} \in \mathbb{R}^{d\times d}$ by $\hat{U}_i = [\hat{u}_{i1}, \cdots, \hat{u}_{id}]$. For any integer subset $\Pi = \{\pi_1, \cdots, \pi_p\}$, we have

$$0 \le \sum_{k=1}^p \hat{u}_{\pi_k j}^2 \le 1,\ \sum_{j=1}^d \hat{u}_{\pi_k j}^2 = 1,\ \forall j \in \mathbb{N}_d,\ \forall k \in \mathbb{N}_p.$$

The i-th diagonal entry of Z can be expressed as $Z_i = \sum_{j=1}^d \hat{\sigma}_j \hat{u}_{ij}^2$. It follows that

$$\begin{aligned} \sum_{i=1}^p Z_{\pi_i} &= \sum_{j=1}^d \left(\hat{\sigma}_j \hat{u}_{\pi_1 j}^2 + \cdots + \hat{\sigma}_j \hat{u}_{\pi_p j}^2\right) \\ &= \sum_{j=1}^d \sum_{k=1}^p \left(\hat{\sigma}_j \hat{u}_{\pi_k j}^2\right) = \sum_{j=1}^d \left(\hat{\sigma}_j \sum_{k=1}^p \hat{u}_{\pi_k j}^2\right) \le \sum_{j=1}^p \hat{\sigma}_j, \end{aligned}$$

where the last equality (the maximum) above is attained when the set $\{\hat{u}_{\pi_1 j}^2, \cdots, \hat{u}_{\pi_p j}^2\}$ ($\forall j \in \mathbb{N}_d$) has only one non-zero element of value one or $p = d$. This completes the proof of this lemma. □

We summarize the main result of this subsection in the following theorem.

Theorem 4.1. *Let $\{\lambda_i^*\}_{i=1}^q$ be optimal to Eq. (17), and denote $\Lambda^* = \operatorname{diag}(\lambda_1^*, \cdots, \lambda_q^*, 0) \in \mathbb{R}^{d\times d}$. Let $P_1 \in \mathbb{R}^{d\times d}$ be orthogonal consisting of the left singular vectors of U. Then $M^* = P_1 \Lambda^* P_1^\mathsf{T}$ is an optimal solution to Eq. (15). Moreover, the problem in Eq. (17)*

attains the same optimal objective value as the one in Eq. (15).

Proof. For any feasible M in Eq. (15), let $M = Q\Lambda Q^\mathsf{T}$ be its SVD, where $Q \in \mathbb{R}^{d\times d}$ is orthogonal, $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_d)$, and $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$. The problem in Eq. (15) can be rewritten as:

$$\begin{aligned} \min_{Q,\Lambda} \quad & \mathrm{tr}\left((\eta I + \Lambda)^{-1} Q^\mathsf{T} P_1 \Sigma\Sigma^\mathsf{T} P_1^\mathsf{T} Q\right) \\ \text{subject to} \quad & QQ^\mathsf{T} = Q^\mathsf{T}Q = I,\ \Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_d), \\ & \textstyle\sum_{i=1}^d \lambda_i = h,\ 1 \geq \lambda_1 \geq \cdots \geq \lambda_d \geq 0, \quad (19) \end{aligned}$$

where $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_q, 0)$ is defined in Eq. (16). Note that the reformulated problem in Eq. (19) is equivalent to the one in Eq. (15) and has two separate optimization variables Q and Λ.

We show that the optimization variable Q can be factored out from Eq. (19). Let $D = Q^\mathsf{T} P_1 \Sigma\Sigma^\mathsf{T} P_1^\mathsf{T} Q$ and denote its diagonal entries by $\{D_i\}_{i=1}^d$. It follows from Eq. (16) that D is a positive semidefinite matrix with non-zero singular values $\{\sigma_i^2\}_{i=1}^q$. Given any feasible Λ in Eq. (19), we have

$$\begin{aligned} & \min_{Q^\mathsf{T}Q = QQ^\mathsf{T} = I} \mathrm{tr}\left((\eta I + \Lambda)^{-1} Q^\mathsf{T} P_1 \Sigma\Sigma^\mathsf{T} P_1^\mathsf{T} Q\right) \\ = & \min_{D \in \mathbb{S}_+^d : D \sim \Sigma\Sigma^T} \sum_{i=1}^d \frac{D_i}{\eta + \lambda_i}, \quad (20) \end{aligned}$$

where $D \sim \Sigma\Sigma^\mathsf{T}$ indicates that the eigenvalues of D are given by the diagonal elements of $\Sigma\Sigma^\mathsf{T}$, and the equality above means that these two problems attain the same optimal objective value. Following the non-decreasing order of $1/(\eta+\lambda_i)$ ($i \in \mathbb{N}_d$) and $\sum_{i=1}^p D_{\pi_i} \leq \sum_{j=1}^p \sigma_j^2$ for any integer subset $\{\pi_i\}_{i=1}^p$ (Lemma 4.1), we can verify that the optimal objective value to Eq. (20) is given by

$$\sum_{i=1}^q \frac{\sigma_i^2}{\eta + \lambda_i} + \sum_{i=q+1}^d \frac{0}{\eta + \lambda_i} = \sum_{i=1}^q \frac{\sigma_i^2}{\eta + \lambda_i}, \quad (21)$$

where this optimum can be attained when $Q^\mathsf{T} P_1 = I$ (Golub & Loan, 1996) and $D = \Sigma\Sigma^\mathsf{T}$. It follows from Eq. (21) that the optimal $\{\lambda_i^*\}_{i=1}^d$ to Eq. (19) satisfy $\lambda_{q+1}^* = \cdots = \lambda_d^* = 0$.

In summary, the optimal objective value to Eq. (19) or equivalently Eq. (15) can be obtained via minimizing Eq. (21) subject to the constraints on $\{\lambda_i\}$ or equivalently Eq. (17). Since Eq. (20) is minimized when $Q = P_1$, we conclude that $M^* = P_1 \Lambda^* P_1^\mathsf{T}$ is optimal to Eq. (15). This completes the proof. □

Note that the optimization problem (not strictly convex) in Eq. (15) may have multiple global minimizers yet with the same objective value, while the formulation in Eq. (17) can find one of those global minimizers.

4.3. Convergence Analysis

The alternating optimization procedure employed in Algorithm 1 (cASO) is widely used for for solving many optimization problems efficiently. However, such a procedure does not generally guarantee the global convergence. We summarize the global convergence property of cASO algorithm in the following theorem.

Theorem 4.2. *Algorithm 1 converges to the global minimizer of the optimization problem F_2 in Eq. (11).*

Proof. The proof follows similar arguments in (Argyriou et al., 2007; Argyriou et al., 2008). □

5. Computation of an Optimal Solution to iASO

Recall that F_2 in Eq. (11) is a convex relaxation of iASO in Eq. (3). In this section, we present a theoretical condition under which a globally optimal solution to iASO can be obtained via cASO.

We first present the following lemma, which is the key building block of the analysis in this section.

Lemma 5.1. *Let $\{\sigma_i\}_{i=1}^m$ be defined in Eq. (16) and $\{\gamma_i^*\}_{i=1}^q$ be optimal to Eq. (17). For any $h \in \mathbb{N}_q$, if $\sigma_h/\sigma_{h+1} \geq 1 + 1/\eta$, then $\gamma_1^* = \cdots = \gamma_h^* = 1$ and $\gamma_{h+1}^* = \cdots = \gamma_q^* = 0$.*

Proof. Prove by contradiction. Assume that $\gamma_1^* = \cdots = \gamma_h^* = 1$ and $\gamma_{h+1}^* = \cdots = \gamma_q^* = 0$ do not hold; this contrary leads to $\gamma_h^* \neq 1$ and hence $0 < \gamma_{h+1}^* \leq \gamma_h^* < 1$, since $\sum_{i=1}^q \gamma_i^* = h$ and γ_i^* is non-increasing with i. We show that there exists a feasible solution $\{\zeta_i^*\}_{i=1}^m$ such that $\sum_{i=1}^m \sigma_i^2/(\eta+\gamma_i^*) > \sum_{i=1}^m \sigma_i^2/(\eta+\zeta_i^*)$, thus reaching a contradiction.

Let γ_a^* be the element in $\{\gamma_i^*\}_{i=1}^q$ with the smallest index $a \in \mathbb{N}_h$, satisfying $\gamma_a^* \neq 1$. Let γ_b^* be the element in $\{\gamma_i^*\}_{i=1}^q$ with the largest index $b \in \mathbb{N}_q$, satisfying $\gamma_b^* \neq 0$. Note that it can be verified that $a \leq h$ and $h+1 \leq b$. For any $0 < \delta < \min(1-\gamma_a^*, \gamma_b^*)$, we can construct a feasible solution $\{\zeta_i^*\}_{i=1}^m$ to Eq. (17) as:

$$\zeta_i^* = \begin{cases} \gamma_i^* & i \in \mathbb{N}_q,\ i \neq a,\ i \neq b \\ \gamma_a^* + \delta & i = a \\ \gamma_b^* - \delta & i = b \end{cases}$$

such that $1 \geq \zeta_1^* \geq \cdots > \zeta_a^* > \cdots \geq \zeta_h^* > \cdots > \zeta_b^* >$

$0 = \cdots = 0$. Moreover, we have

$$\begin{aligned}
&\left(\frac{\sigma_a^2}{\eta+\gamma_a^*}+\frac{\sigma_b^2}{\eta+\gamma_b^*}\right)-\left(\frac{\sigma_a^2}{\eta+\zeta_a^*}+\frac{\sigma_b^2}{\eta+\zeta_b^*}\right)\\
&=\delta\left(\frac{\sigma_a^2}{(\eta+\gamma_a^*)(\eta+\gamma_a^*+\delta)}-\frac{\sigma_b^2}{(\eta+\gamma_b^*)(\eta+\gamma_b^*-\delta)}\right)\\
&\geq\sigma_{h+1}^2\delta\left(\frac{(1+1/\eta)^2}{(\eta+\gamma_a^*)(\eta+\gamma_a^*+\delta)}-\frac{1}{(\eta+\gamma_b^*)(\eta+\gamma_b^*-\delta)}\right)\\
&>\sigma_{h+1}^2\delta\left(\frac{(1+1/\eta)^2}{(\eta+1)(\eta+1)}-\frac{1}{\eta^2}\right)=0,
\end{aligned}$$

where the first inequality follows from $\sigma_h/\sigma_{h+1} \geq 1 + 1/\eta$, $\sigma_a \geq \sigma_h \geq (1+1/\eta)\,\sigma_{h+1}$, and $\sigma_{h+1} \geq \sigma_b$; the second (strict) inequality follows from $1 > \gamma_a^*, \gamma_b^* > 0$, and $1 \geq \gamma_a^* + \delta, \gamma_b^* - \delta \geq 0$. Therefore $\sum_{i=1}^m \sigma_i^2/(\eta + \gamma_i^*) > \sum_{i=1}^m \sigma_i^2/(\eta+\zeta_i^*)$. This completes the proof. □

We conclude this section using the following theorem.

Theorem 5.1. *Let the problems F_1 and F_2 be defined in Eqs. (8) and (11), respectively, and let (U^*, M^*) be the optimal solution to F_2. Let $P_1 \in \mathbb{R}^{d\times d}$ be orthogonal consisting of the left singular vectors of U^*, and $\{\sigma_i\}_{i=1}^q$ be the corresponding non-zero singular values of U^* in non-increasing order. Let Θ^* consist of the first h column-vectors of P_1 corresponding to the largest h singular values. If $\sigma_h/\sigma_{h+1} \geq 1 + 1/\eta$, then the optimal solution to F_1 is given by (U^*, Θ^*).*

Proof. Since (U^*, M^*) is optimal to F_2, it follows from Theorem 4.1 that M^* can be expressed as $M^* = P_1 \Lambda P_1^\mathsf{T}$, where $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_d) \in \mathbb{R}^{d\times d}$ can be computed via Eq. (17). Given $\sigma_h/\sigma_{h+1} \geq 1 + 1/\eta$, we can verify that $\lambda_i = 1$ if $i \in \mathbb{N}_h$, and 0 otherwise (Lemma 5.1); therefore $M^* = {\Theta^*}^\mathsf{T}\Theta^*$, where $\Theta^* \in \mathbb{R}^{d\times h}$ corresponds to the first h column-vectors of P_1. Moreover, given a fixed $U \in \mathbb{R}^{d\times m}$ in F_1 and F_2 respectively, we have

$$\min_{\Theta^\mathsf{T}\Theta\in\mathcal{M}_e,\Theta\Theta^\mathsf{T}=I} G_1(U,\Theta) \geq \min_{M\in\mathcal{M}_c} G_2(U,M), \qquad (22)$$

where $G_1(U,\Theta)$ and $G_2(U.M)$ are defined in Eqs. (7) and (12) respectively, and $\mathcal{M}_e$ and $\mathcal{M}_c$ are defined in Eqs. (9) and (10) respectively. The equality in Eq. (22) is attained when the optimal M to the right side of Eq. (22) is an extreme point of the set $\mathcal{M}_c$, i.e., belong to the set $\mathcal{M}_e$. For a given U^*, if $\sigma_h/\sigma_{h+1} \geq 1 + 1/\eta$ is satisfied, Θ^* minimizes $G_1(U^*, \Theta)$ and the equality in Eq. (22) can be attained. Hence, (U^*, Θ^*) is the optimal solution to F_1. This completes the proof. □

6. Experiments

In this section, we evaluate the proposed *c*ASO algorithm on the tasks of multi-topic web pages categorization using the Yahoo web pages data sets (Ueda & Saito, 2002).

The Yahoo data sets consist of 11 top-level categories (each category corresponds to one data set), and each category is further divided into a set of second-level categories (each sub-category corresponds to a topic in one data set). We preprocess the data sets by removing topics with fewer than 100 web pages. The TF-IDF scheme is used to represent the web pages, and the obtained feature vectors are normalized to unit length. In the experiments, we use the SVM hinge loss; quadratic programs (QP) are solved via MOSEK[2], and SVM is solved via LIBSVM package (Chang & Lin, 2001).

Performance Comparsion We compare the proposed *c*ASO algorithm with SVM (independent SVM for multi-task learning), ASO (alternating structure optimization) (Ando & Zhang, 2005), and *c*MTFL (convex multi-task feature learning) (Argyriou et al., 2008) on the tasks of web pages categorization.

We use Macro F1 and Micro F1 (Lewis et al., 2004) as the performance measures. The parameters in the competing algorithms (the penalty parameter C of SVM, the regularization parameters of ASO, *c*ASO and *c*MTFL) are determined via 3-fold cross-validation. In ASO, *c*ASO and *c*MTFL, the stopping criterion is set as that the relative change of the objective value is smaller than 10^{-4}. We randomly sample 1500 data points from each data set as training data, and the remaining are used as test data.

The experimental results (averaged over 5 random repetitions) as well as the standard deviation are presented in Tables 1 and 2. We can observe that *c*ASO is competitive with other competing algorithms on all of the 11 data sets. Moreover, *c*ASO outperforms ASO (in this supervised setting) on 9 data sets in terms of both Macro F1 and Micro F1. This superiority may be due to the flexibility in the *c*ASO formulation (via the regularization term), and its guaranteed global optimal solution. The reason for the low performance of SVM probably lies in that it does not utilize the relationship of the multiple learning tasks.

Efficiency Comparison We compare the efficiency of *c*ASO with other competing algorithms in terms of computation time (in seconds) on the Arts data set. We increase the training sample size from 500 to 2500, and record the computation time. From Figure 1, we can observe that SVM is the fastest algorithm, while ASO is the slowest one. Note that in practice, early stopping can be applied in ASO as in (Ando & Zhang, 2005). *c*ASO performs faster than *c*MTFL

[2] http://www.mosek.com/

Table 1. Performance comparison of competing algorithms on six Yahoo data sets. The statistics of the data sets are presented in the second row, where n, d, and m denotes the sample size, dimensionality, and the number of topics (tasks), respectively. In ASO and *c*ASO, the shared feature dimensionality h is set as $\lfloor (m-1)/5 \rfloor \times 5$.

Data (n, d, m)		Arts (7441, 17973, 19)	Business (9968, 16621, 17)	Computers (12317, 25259, 23)	Education (11817, 20782, 14)	Entertainment (12691, 27435, 14)	Health (9209, 18430, 14)
Macro F1	SVM	33.93 ± 1.07	44.43 ± 0.56	30.09 ± 1.10	39.00 ± 2.42	46.88 ± 0.47	56.14 ± 2.58
	ASO	**37.93 ± 1.57**	44.64 ± 0.40	28.33 ± 0.67	36.93 ± 1.98	47.46 ± 0.37	57.63 ± 0.74
	*c*ASO	37.35 ± 0.60	**45.79 ± 0.69**	**33.35 ± 0.84**	**41.28 ± 0.90**	49.66 ± 0.97	**61.16 ± 1.70**
	*c*MTFL	37.06 ± 0.75	40.90 ± 1.66	32.50 ± 0.90	40.17 ± 0.55	**50.94 ± 1.06**	58.66 ± 2.22
Micro F1	SVM	43.99 ± 1.23	77.51 ± 0.51	**55.36 ± 0.63**	48.03 ± 1.56	55.69 ± 2.45	61.40 ± 4.76
	ASO	43.96 ± 0.03	**78.08 ± 0.25**	54.43 ± 0.40	46.97 ± 0.37	57.71 ± 0.33	65.90 ± 0.39
	*c*ASO	**47.69 ± 0.47**	77.44 ± 0.94	54.54 ± 1.07	**49.50 ± 0.57**	57.90 ± 1.38	**68.19 ± 1.01**
	*c*MTFL	46.31 ± 0.32	69.00 ± 1.01	49.38 ± 4.22	48.56 ± 0.40	**58.25 ± 0.76**	66.83 ± 1.72

Table 2. Performance comparison of competing algorithms on five Yahoo data sets. Explanation can be found in Table 1.

Data Set (n, d, m)		Recreation (12797, 25095, 18)	Reference (7929, 26397, 15)	Science (6345, 24002, 22)	Social (11914, 32492, 21)	Society (14507, 29189, 21)
Macro F1	SVM	43.01 ± 1.44	39.37 ± 1.15	41.80 ± 1.45	35.87 ± 0.79	30.68 ± 0.94
	ASO	43.63 ± 1.29	37.46 ± 0.27	39.26 ± 0.82	35.29 ± 0.67	29.42 ± 0.30
	*c*ASO	**47.12 ± 0.73**	42.11 ± 0.60	**45.46 ± 0.50**	**39.30 ± 1.28**	**34.84 ± 1.05**
	*c*MTFL	46.13 ± 0.58	**43.25 ± 0.81**	42.52 ± 0.59	38.94 ± 1.88	33.79 ± 1.43
Micro F1	SVM	49.15 ± 2.32	55.11 ± 3.16	49.27 ± 4.64	63.05 ± 2.45	40.07 ± 3.42
	ASO	50.68 ± 0.18	57.72 ± 0.51	49.05 ± 0.57	62.77 ± 3.59	46.13 ± 2.33
	*c*ASO	**53.34 ± 0.90**	**59.39 ± 0.39**	**53.32 ± 0.45**	**66.04 ± 0.62**	**49.27 ± 0.55**
	*c*MTFL	52.52 ± 0.92	58.49 ± 0.51	50.60 ± 0.76	65.60 ± 0.63	46.46 ± 0.87

when training with a fixed regularization parameter. *c*ASO and *c*MTFL have comparable efficiency, when training with parameter tuning, and their computation time is close to that of SVM.

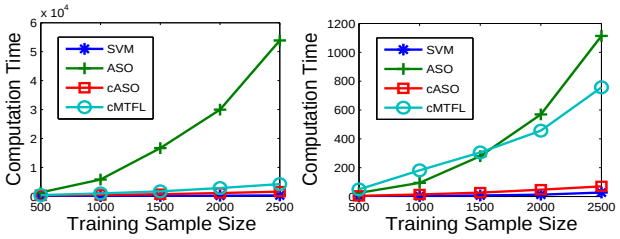

Figure 1. Comparison of computation time (in seconds) (a) training with parameter tuning; (b) training with a fixed penalty/regularization parameter (the best one obtained from tuning).

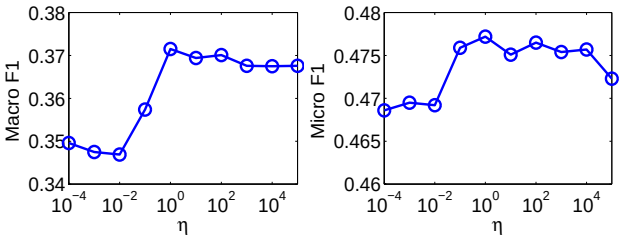

Figure 2. Sensitivity study of η on Micro/Micro F1.

Sensitivity Study We study the effect of η on the performance of *c*ASO on Arts data. Recall that $\eta = \beta/\alpha$; we fix $\alpha = 1$ and vary β from 10^{-4} to 10^5, and record the obtained Macro/Micro F1. The experimental results are presented in Figure 2. We can observe that a small η (equivalently small β) leads to lower F1, while $\eta \approx 1$ (equivalently $\alpha \approx \beta$) leads to the highest F1. In the experiments, we also observe that *c*ASO requires more computation time for convergence using a small η, while less computation time is required for a large η.

Table 3. Comparison of the optimal objective values of F_0 and F_2 with different choices of η.

η	1000	100	10	1	0.1	0.01	0.001
$1 + 1/\eta$	1.001	1.01	1.1	2	11	101	1001
σ_h/σ_{h+1}	1.23	1.25	1.34	1.75	3.07	13.79	89.49
OBJ_{F_0}	52.78	52.65	51.37	40.73	22.15	5.95	0.69
OBJ_{F_2}	52.78	52.65	51.37	40.71	20.73	4.11	0.41

Relationship Study on F_0 and F_2 We study the relationship between the problems F_0 and F_2 as well as the condition presented in Theorem 5.1. We vary the parameter η (by fixing $\alpha = 1$ and varying β accordingly) from 10^3 to 10^{-3}. F_2 is solved via Algorithm 1, and the optimal objective value OBJ_{F_2} and the value of σ_h/σ_{h+1} are recorded. Similarly, F_0 is solved via alternating optimization and its optimal objective value OBJ_{F_0} is recorded. From the experimental results in Table 3, we can observe that when $\eta \in \{1000, 100, 10\}$, the condition $\sigma_h/\sigma_{h+1} > 1 + 1/\eta$ is satisfied and hence $\mathrm{OBJ}_{F_0} = \mathrm{OBJ}_{F_2}$; otherwise, we observe $\mathrm{OBJ}_{F_0} > \mathrm{OBJ}_{F_2}$. This empirical result is consistent with our theoretical analysis in Theorem 5.1.

7. Conclusion and Future Work

We present a multi-task learning formulation called *i*ASO for learning a shared feature representation from

multiple related tasks. Since iASO is non-convex, we convert it into a relaxed convex formulation, and then develop the cASO algorithm to solve the convex relaxation efficiently. Our convergence analysis shows that the cASO algorithm converges to a globally optimal solution to the convex relaxation. We also present a theoretical condition, under which cASO can find a globally optimal solution to iASO.

We have conducted experiments on benchmark data sets; the experimental results are consistent with our theoretical analysis. We also observe that cASO with a non-zero η, tends to either increase or at least keep the generalization performance compared with ASO, while significantly reducing the computational cost. We are currently investigating how the solutions of cASO depend on the parameters involved in the formulation as well as their estimation. We plan to compare the presented iASO formulation with the multi-task learning formulation using the trace-norm regularization. We also plan to apply the cASO algorithm to applications such as the automatic processing of biomedical texts for tagging the gene mentions (Ando, 2007).

Acknowledgments

This work was supported by NSF IIS-0612069, IIS-0812551, CCF-0811790, NIH R01-HG002516, and NGA HM1582-08-1-0016.

References

Amit, Y., Fink, M., Srebro, N., & Ullman, S. (2007). Uncovering shared structures in multiclass classification. *ICML*.

Ando, R. K. (2007). BioCreative II gene mention tagging system at IBM Watson. *Proceedings of the Second BioCreative Challenge Evaluation Workshop*.

Ando, R. K., & 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., & Pontil, M. (2008). Convex multi-task feature learning. *Machine Learning*, *73*, 243–272.

Argyriou, A., Micchelli, C. A., Pontil, M., & Ying, Y. (2007). A spectral regularization framework for multi-task structure learning. *NIPS*.

Bakker, B., & Heskes, T. (2003). Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, *4*, 83–99.

Baxter, J. (2000). A model of inductive bias learning. *Journal of Artificial Intelligence Research*, *12*, 149–198.

Bertsekas, D. P. (1999). *Nonlinear programming*. Athena Scientific.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

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

Chang, C.-C., & Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*.

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

Golub, G. H., & Loan, C. F. V. (1996). *Matrix computations*. Johns Hopkins University Press.

Heisele, B., Serre, T., Pontil, M., Vetter, T., & Poggio, T. (2001). Categorization by learning and combining object parts. *NIPS*.

Jacob, L., Bach, F., & Vert, J.-P. (2008). Clustered multi-task learning: A convex formulation. *NIPS*.

Lawrence, N. D., & Platt, J. C. (2004). Learning to learn with the informative vector machine. *ICML*.

Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, *5*, 361–397.

Obozinski, G., Taskar, B., & Jordan, M. I. (2006). Multi-task feature selection. *Technical report, Dept. of Statistics, UC Berkeley*.

Overton, M. L., & Womersley, R. S. (1993). Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrics. *Mathematical Programming*, *62*, 321–357.

Quattoni, A., Collins, M., & Darrell, T. (2007). Learning visual representations using images with captions. *CVPR*.

Schwaighofer, A., Tresp, V., & Yu, K. (2004). Learning gaussian process kernels via hierarchical bayes. *NIPS*.

Ueda, N., & Saito, K. (2002). Parametric mixture models for multi-labeled text. *NIPS*.

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

Yu, K., Tresp, V., & Schwaighofer, A. (2005). Learning gaussian processes from multiple tasks. *ICML*.

Yu, S., Tresp, V., & Yu, K. (2007). Robust multi-task learning with t-processes. *ICML*.

Zhang, J., Ghahramani, Z., & Yang, Y. (2005). Learning multiple related tasks using latent independent component analysis. *NIPS*.