Semi-Supervised Multi-Task Learning with Task Regularizations

Fei Wang, Xin Wang, Tao Li
School of Computing and Information Sciences
Florida International University
Miami, FL 33199 USA
{*feiwang,xwang009,taoli*}*@cs.fiu.edu*

***Abstract*—Multi-task learning refers to the learning problem of performing inference by jointly considering multiple related tasks. There have already been many research efforts on supervised multi-task learning. However, collecting sufficient labeled data for each task is usually time consuming and expensive. In this paper, we consider the *semi-supervised multi-task learning* (*SSMTL*) problem, where we are given a small portion of labeled points together with a large pool of unlabeled data within each task. We assume that the different tasks can form some *task clusters* and the task in the same cluster share similar classifier parameters. The final learning problem is relaxed to a convex one and an efficient *gradient descent* strategy is proposed. Finally the experimental results on both synthetic and real world data sets are presented to show the effectiveness of our method.**

I. Introduction

In many practical problems, a learning task can usually involve multiple related tasks. Solving those related tasks together is expected to be more advantageous than solving them independently because the knowledge from different tasks can help to improve the generalization ability of the classifier constructed on each individual task. In recent years, the problem of learning multiple related tasks (usually referred to as *multi-task learning* (*MTL*)) has been investigated extensively [2][3][11][15][4][16], and the related approaches have been applied to a variety of application areas, such as marketing [5], computer vision [19] and bioinformatics[9].

Clearly, the exploration of task relationships is at the heart of multi-task learning. Many algorithms have been proposed to address this issue. Sheldon [18] and Kato *et al.* [15] assume task relationships are known beforehand; Argyriou *et al.* [3] assumes the data associated with those tasks share some common latent features and derive an optimization strategy to obtain these features; Evgeniou and Pontil [11] suppose that the classifiers of all these related tasks share a common part of parameters; Allenby and Rossi [1] and Arora *et al.* [5] applied a hierarchical Bayesian model with a common prior for the classifier parameters of all tasks.

As another research line, *semi-supervised learning* (*SSL*) [8] provides an alterative way to improve the classification performances. SSL assumes we are given a partially labeled data set, and it makes inferences by making use of both labeled and unlabeled points [2], [16]. There are also some works that integrate MTL and SSL together, where they assume that there are some labeled and unlabeled points available for each task, and they use all these data samples and the task relationships to perform multi-task learning.

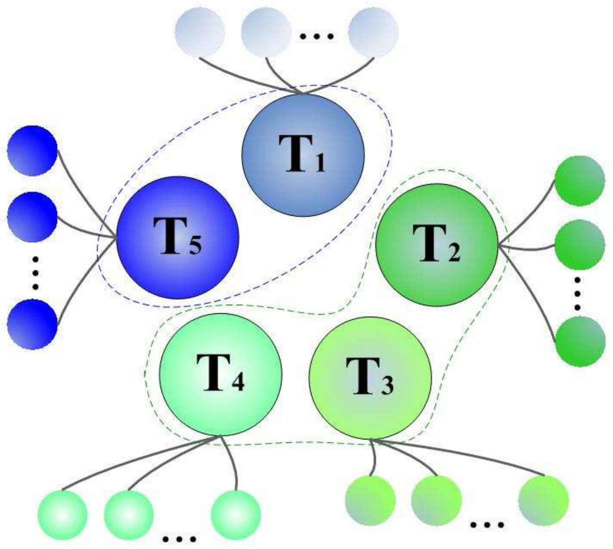


Figure 1. A graphical illustration of learning with task groups, where the large nodes correspond to tasks, and the connected small nodes are the data points within the tasks. The tasks in a dashed line form a task group.

Despite their empirical success and theoretical soundness, one of the major limitations of those traditional methods is that they assume all the tasks should be related with each other. In practice, a more reasonable assumption is that the tasks can form different *groups*, and the tasks in each group should share similar classification rules (see Fig.1 for a graphical illustration). There are some efforts based on this assumption [6], [13], however, they are all supervised approaches.

In this paper, we propose a novel *semi-supervised multi-task learning* (*SSMTL*) framework, which assumes that we are given a partially labeled set associated with each task. We construct a linear classifier for each task according to the underlying task-specific data manifolds. Then all the classification vectors will be integrated together by a K-means like *inter-task* regularization term which can help explore the task clusters. After some relaxations, we show that our problem is convex and an efficient gradient descent method is derived to solve it. Moreover, we also show that our algorithm can easily be kernelized to nonlinear cases. Finally the experimental results on both synthetic and real world data sets are presented to show the effectiveness of our method.

II. The Algorithm

In the multi-task setting, we are given a set of data points $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$. Each $\mathbf{x}_i$ is associated with a specific task t $(t = 1, 2, \cdots, m)$. We denote by $\mathcal{I}_t$ the set of indices of the data points associated to the task t. For semi-supervised multi-task learning, we assume that for each task t we have l_t labeled points, and we denote the indices of the labeled samples in task t as $\mathcal{I}_{\mathcal{L}_t}$, the indices of the unlabeled samples in task t as $\mathcal{I}_{\mathcal{U}_t}$, then $\mathcal{I}_t = \mathcal{I}_{\mathcal{L}_t} \bigcup \mathcal{I}_{\mathcal{U}_t}$. If $\mathbf{x}_i$ is labeled, we denote its label by y_i. For convenience, we assume that $\forall i \in \bigcup_t \mathcal{I}_{\mathcal{L}_t},\ y_i \in \{+1, -1\}$, and for each task we can learn a linear classifier $f_t(\mathbf{x}) = \mathbf{w}_t^\top \mathbf{x}$, such that the predicted label of $\mathbf{x}$ in task t can be determined by $\text{sign}(f(\mathbf{x}))$. We denote by $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_m] \in \mathbb{R}^{d\times m}$ the matrix whose columns are the successive vectors we want to estimate.

We use square loss to measure the prediction quality of the linear classifiers, i.e., for task t, the prediction loss would be

$$\mathcal{J}_{loss}^t = \sum_{i\in\mathcal{I}_{\mathcal{L}_t}} (\mathbf{w}_t^\top \mathbf{x}_i - y_i)^2 \tag{1}$$

Then the total prediction loss becomes

$$\mathcal{J}_{loss} = \sum_t \sum_{i\in\mathcal{I}_{\mathcal{L}_t}} (\mathbf{w}_t^\top \mathbf{x}_i - y_i)^2 \tag{2}$$

We define the data matrix associated with the t-th task as $\mathbf{X}_t \in \mathbb{R}^{d\times n_t}$. From the *representor theorem* [17], we can write $\mathbf{w}_t$ as a linear combination of the data points in task t, i.e.,

$$\mathbf{w}_t = \textstyle\sum_{i\in\mathcal{I}_t} h_i^t \mathbf{x}_i = \mathbf{X}_t \mathbf{h}_t \tag{3}$$

where $\mathbf{h}_t = [h_1^t, h_2^t, \cdots, h_{n_t}^t]^T$ is the combination coefficient vector. Define the label truncated identity matrix for task t as $\mathbf{J}_t \in \mathbb{R}^{n_t\times n_t}$, which is a diagonal matrix with

$$J_{ii} = \begin{cases} 1, & \text{if } i \in \mathcal{I}_{\mathcal{L}_t} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

and the initial label vector for task t as $\mathbf{y}_t \in \mathbb{R}^{n_t\times 1}$ with

$$y_t(i) = \begin{cases} y_i, & \text{if } i \in \mathcal{I}_{\mathcal{L}_t} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

We further define four concatenated matrices

$$\begin{aligned} \mathbf{X} &= [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_m] \in \mathbb{R}^{d\times n} && (6)\\ \mathbf{H} &= diag(\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_m) \in \mathbb{R}^{n\times m} && (7)\\ \mathbf{Y} &= diag(\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_m) \in \mathbb{R}^{n\times m} && (8)\\ \mathbf{G} &= diag(\mathbf{G}_1, \mathbf{G}_2, \cdots, \mathbf{G}_m) \in \mathbb{R}^{n\times n} && (9)\\ \mathbf{J} &= diag(\mathbf{J}_1, \mathbf{J}_2, \cdots, \mathbf{J}_m) \in \mathbb{R}^{n\times n} && (10) \end{aligned}$$

where $\mathbf{G}_t = \mathbf{X}_t^\top \mathbf{X}_t$ is the *Gram matrix* for task t. Then we have

$$\begin{aligned} \mathbf{W} &= \mathbf{XH} && (11)\\ \mathcal{J}_{loss} &= \|\mathbf{Y} - \mathbf{JGH}\|_F^2 && (12) \end{aligned}$$

A. Intra-Task Regularization

In our *Semi-Supervised Multi-Task Learning* (*SSMTL*) framework, we propose to explore both the intra-task and inter-task information to help determine the classification function. In this section we will introduce how to explore the intra-task information.

Specifically, within each task t, we first want the classification function to have a better generalization ability, which motivates us to punish the norm of $\mathbf{w}_t$, since it relates to the reciprocal of the classification margin [20]. Thus we can adopt the following regularization term to penalize the norm

$$\mathcal{C} = \sum_t \|\mathbf{w}_t\|^2 = \sum_t \|\mathbf{X}_t \mathbf{h}_t\|^2 = tr(\mathbf{H}^\top \mathbf{G}^\top \mathbf{H}) \tag{13}$$

Secondly, we want that for each task, the predicted labels should be sufficiently smooth with respect to its corresponding intrinsic data manifold. According to [7], such smoothness can be measured by

$$\mathcal{S}_t = \mathbf{w}_t^\top \mathbf{X}_t \mathbf{L}_t \mathbf{X}_t^\top \mathbf{w}_t \tag{14}$$

where $\mathbf{L}_t \in \mathbb{R}^{n_t\times n_t}$ is the *graph Laplacian* constructed for task t, whose (i, j)-th entry is [10]

$$L_t(i, j) = \begin{cases} -s_{ij}, & \text{if } i \neq j \\ d_{ii} - s_{ii}, & \text{otherwise} \end{cases} \tag{15}$$

where s_{ij} is the similarity between the i-th and j-th data points associated with the t-th task, which can usually be computed by the following Gaussian function

$$s_{ij} = \exp\left(-\frac{\|\mathbf{x}_i^t - \mathbf{x}_j^t\|^2}{2\sigma_t^2}\right) \tag{16}$$

where we use $\mathbf{x}_i^t$ to represent the i-th data point in task t. Therefore the total smoothness of the classification functions can be measured by

$$\begin{aligned} \mathcal{S} &= \sum_t \mathcal{S}_t = \sum_t \mathbf{w}_t^\top \mathbf{X}_t \mathbf{L}_t \mathbf{X}_t^\top \mathbf{w}_t \\ &= \sum_t \mathbf{h}_t^\top \mathbf{G}_t \mathbf{L}_t \mathbf{G}_t \mathbf{h}_t \\ &= tr\left(\mathbf{H}^\top \mathbf{GLGH}\right) && (17) \end{aligned}$$

where

$$\mathbf{L} = diag(\mathbf{L}_1, \mathbf{L}_2, \cdots, \mathbf{L}_m) \in \mathbb{R}^{n\times n} \tag{18}$$

is the concatenated Laplacian matrix.

Combining Eq.(13) and Eq.(17) we can construct a intra-task loss as

$$\begin{aligned} \mathcal{J}_{intra} &= tr\left(\mathbf{H}^\top \mathbf{G}\mathbf{H}\right) + \delta tr\left(\mathbf{H}^\top \mathbf{G}\mathbf{L}\mathbf{G}\mathbf{H}\right) \\ &= tr\left(\mathbf{H}^\top \left(\mathbf{G} + \delta\mathbf{G}\mathbf{L}\mathbf{G}\right)\mathbf{H}\right) \end{aligned} \tag{19}$$

B. Inter-Task Regularization

Besides exploring the information within each task, another important issue is how to explore the task relationships in multi-task learning. Some previous works assume that the inter-task relationships are known beforehand [18][15]. In this paper, we will not make such assumption and we will design a paradigm to explore such information automatically. Specifically, we assume there are hidden clusters among those tasks, and the tasks in similar clusters should share similar weight vectors. Using the k-means criterion, assuming the tasks can formulate k clusters, we can write the objective

$$\mathcal{J}_{inter} = \sum_{i=1}^{k} \sum_{\mathcal{T}_t \in \pi_i} \|\mathbf{w}_t - \widetilde{\mathbf{w}}_i\|^2 \tag{20}$$

where $\mathcal{T}_t$ represents the t-th task, π_i denotes the i-th task cluster. $\widetilde{\mathbf{w}}_i$ is the mean classification vector of the i-th task cluster.

Now we introduce an $m \times k$ task cluster indicator matrix $\mathbf{E}$ with its (i,j)-th entry

$$E_{ij} = \begin{cases} 1, & \text{if } \mathcal{T}_i \in \pi_j \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

Let

$$\mathbf{M} = \mathbf{I} - \mathbf{E}\left(\mathbf{E}^\top \mathbf{E}\right)^{-1}\mathbf{E}^\top \tag{22}$$

Then Eq.(20) can be rewritten as

$$\mathcal{J}_{inter} = tr(\mathbf{W}\mathbf{M}\mathbf{W}^\top) = tr(\mathbf{X}\mathbf{H}\mathbf{M}\mathbf{H}^\top\mathbf{X}^\top) \tag{23}$$

C. Semi-Supervised Multi-Task Learning with Task Regularizations

Combining all things together, we can derive the following objective for semi-supervised multi-task learning:

$$\begin{aligned} \mathcal{J} &= \mathcal{J}_{loss} + \lambda_1 \mathcal{J}_{intra} + \lambda_2 \mathcal{J}_{inter} \\ &= \|\mathbf{Y} - \mathbf{J}\mathbf{G}\mathbf{H}\|_F^2 + \lambda_1 tr\left(\mathbf{H}^\top\left(\mathbf{G} + \delta\mathbf{G}\mathbf{L}\mathbf{G}\right)\mathbf{H}\right) \\ &\quad + \lambda_2 tr(\mathbf{X}\mathbf{H}\mathbf{M}\mathbf{H}^\top\mathbf{X}^\top) \end{aligned} \tag{24}$$

which is convex with respect to both $\mathbf{H}$ and $\mathbf{M}$. Now we will analyze the constraints on $\mathbf{H}$ and $\mathbf{M}$.

- According to the definition of $\mathbf{H}$ in Eq.(7), there are no specific constraints on $\mathbf{H}$ except its block-diagonal structure. Such structure constraints are equivalent to restrict the non-diagonal elements of $\mathbf{H}$ to be zeros, which are thus linear and will not effect the convexity of the problem of minimizing $\mathcal{J}$.
- The constraint on $\mathbf{M}$ is a bit more complicated because of its special structure as defined in Eq.(22) and the integer values in $\mathbf{E}$ (see Eq.(21)). To relax it, we observe that (1) $\mathbf{M}$ is idempotent; (2) $\mathbf{M}$ is symmetric. Therefore $\mathbf{M}$ is an orthogonal projector, i.e., $\mathbf{M}$ and $\mathbf{I} - \mathbf{M}$ are all *positive semi-definite* (*PSD*). Moreover, from its definition we can easily see that $tr\mathbf{M} = m - k$. Thus we relax $\mathbf{M}$ to lie in the following set

$$\mathcal{M} = \{\mathbf{M} : \mathbf{0} \preceq \mathbf{M} \preceq \mathbf{I},\ tr\mathbf{M} = m - k\} \tag{25}$$

where we use $\mathbf{0} \preceq \mathbf{M} \preceq \mathbf{I}$ to denote that both $\mathbf{M}$ and $\mathbf{I} - \mathbf{M}$ are PSD.

Combining all above things together, we can derive that our semi-supervised multi-task learning framework aims to solve the following problem

$$\begin{aligned} \min_{\mathbf{H},\mathbf{M}} \quad & \|\mathbf{Y} - \mathbf{J}\mathbf{G}\mathbf{H}\|_F^2 + \lambda_1 tr\left(\mathbf{H}^\top\left(\mathbf{G} + \delta\mathbf{G}\mathbf{L}\mathbf{G}\right)\mathbf{H}\right) \\ & + \lambda_2 tr(\mathbf{X}\mathbf{H}\mathbf{M}\mathbf{H}^\top\mathbf{X}^\top) \\ s.t. \quad & \mathbf{H} \in \mathcal{H},\ \mathbf{M} \in \mathcal{M} \end{aligned} \tag{26}$$

where we use $\mathcal{H}$ to denote the set of block diagonal matrices taking the form as in Eq.(7). In this way, we got a problem that is convex in both $\mathbf{H}$ and $\mathbf{M}$, in the following section we will introduce an efficient optimization method to solve it.

D. Optimization

A most straightforward way to solve problem (26) is *alternative optimization*, which iteratively solves $\mathbf{M}$ (or $\mathbf{H}$) with $\mathbf{H}$ (or $\mathbf{M}$) fixed. Specifically, when $\mathbf{M}$ is fixed as $\mathbf{M} = \mathbf{M}^*$, we should solve the optimal $\mathbf{H}$ by

$$\begin{aligned} \min_{\mathbf{H}} \quad & \mathcal{J} = \|\mathbf{Y} - \mathbf{J}\mathbf{G}\mathbf{H}\|_F^2 + \lambda_1 tr\left(\mathbf{H}^\top\left(\mathbf{G} + \delta\mathbf{G}\mathbf{L}\mathbf{G}\right)\mathbf{H}\right) \\ & + \lambda_2 tr(\mathbf{X}\mathbf{H}\mathbf{M}^*\mathbf{H}^\top\mathbf{X}^\top) \\ s.t. \quad & \mathbf{H} \in \mathcal{H} \end{aligned} \tag{27}$$

First we ignoring the constraint on $\mathbf{H}$, then since $\mathcal{J}$ is convex in $\mathbf{H}$, then the optimal $\mathbf{H}$ can be obtained at the zero point of the following gradient

$$\frac{\partial \mathcal{J}}{\partial \mathbf{H}} = 2\Big((\lambda_1\mathbf{G} + \mathbf{G}(\mathbf{J} + \lambda_1\delta\mathbf{L})\mathbf{G})\mathbf{H} - \mathbf{G}\mathbf{Y} + \lambda_2\mathbf{X}^\top\mathbf{X}\mathbf{H}\mathbf{M}^*\Big)$$

By setting $\partial\mathcal{J}/\partial\mathbf{H} = 0$, we can get that

$$(\lambda_1\mathbf{G} + \mathbf{G}(\mathbf{J} + \lambda_1\delta\mathbf{L})\mathbf{G})\mathbf{H} + \lambda_2\mathbf{X}^\top\mathbf{X}\mathbf{H}\mathbf{M}^* = \mathbf{G}\mathbf{Y} \tag{28}$$

This is a *generalized Sylvester equation* which can be solved with $O(n+m)^3$ [14], which is prohibitive when n and m become large. Therefore we give up such idea and apply the simple gradient descent to solve the optimal $\mathbf{H}$ for problem (26).

Starting with an initial point $\mathbf{H}^0$, the gradient descent method successively updates $\mathbf{H}$ by

$$\mathbf{H}^t = \mathbf{H}^{t-1} + \alpha \left.\frac{\partial \mathcal{J}}{\partial \mathbf{H}}\right|_{\mathbf{H}=\mathbf{H}^{t-1}} \tag{29}$$

where the superscripts correspond to the numbers of iteration steps, α is the step size, and $\partial\mathcal{J}/\partial\mathbf{H}$ can be computed by Eq.(28)[1], where $\mathbf{M}^*$ should be computed by

$$\mathbf{M}^* = arg \min_{\mathbf{M}\in\mathcal{M}} tr(\mathbf{X}\mathbf{H}\mathbf{M}\mathbf{H}^\top\mathbf{X}^\top) \tag{30}$$

Considering the eigenvalue decomposition of $\mathbf{M}$:

$$\mathbf{M} = \mathbf{\Pi}\mathbf{\Lambda}\mathbf{\Pi}^\top \tag{31}$$

where $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2, \cdots, \lambda_m) \in \mathbb{R}^{m\times m}$ is a diagonal matrix with λ_i's the eigenvalues of $\mathbf{M}$. Then

$$\begin{aligned} &\min\nolimits_{\mathbf{M}\in\mathcal{M}} tr(\mathbf{X}\mathbf{H}\mathbf{M}\mathbf{H}^\top\mathbf{X}^\top) \qquad (32)\\ &= \min\nolimits_{0\le\lambda_i\le1, \sum_i \lambda_i = m-k, \mathbf{\Pi}\in\mathcal{O}^m} tr(\mathbf{X}\mathbf{H}\mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top\mathbf{H}^\top\mathbf{X}^\top)\end{aligned}$$

where $\mathcal{O}^m$ is a set of orthogonal matrices in $\mathbb{R}^{m\times m}$. We further consider the *singular value decomposition* (*SVD*) of $\mathbf{X}\mathbf{H}$:

$$\mathbf{X}\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \tag{33}$$

Then

$$tr(\mathbf{X}\mathbf{H}\mathbf{M}\mathbf{H}^\top\mathbf{X}^\top) = tr(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{\Pi}\mathbf{\Lambda}\mathbf{\Pi}^\top\mathbf{V}\mathbf{\Sigma}U^\top) \tag{34}$$

when $\mathbf{\Pi} = \mathbf{V}$, then

$$tr(\mathbf{X}\mathbf{H}\mathbf{M}\mathbf{H}^\top\mathbf{X}^\top) = tr(\mathbf{U}\mathbf{\Sigma}\mathbf{\Lambda}\mathbf{\Sigma}\mathbf{U}^\top) = \sum_{i=1}^m \lambda_i\sigma_i^2 \tag{35}$$

Therefore

$$\min_{\mathbf{M}\in\mathcal{M}} tr(\mathbf{X}\mathbf{H}\mathbf{M}\mathbf{H}^\top\mathbf{X}^\top) = \min_{\substack{0\le\lambda_i\le1,\\ \sum_i\lambda_i=m-k}} \sum_{i=1}^m \lambda_i\sigma_i^2 \tag{36}$$

Then problem (30) becomes an eigenvalue optimization problem as

$$\begin{aligned} \min\nolimits_{\{\lambda_i\}} \quad & \textstyle\sum_{i=1}^m \lambda_i\sigma_i^2 \\ s.t. \quad & \forall\ i = 1,\cdots,m,\ 0\le\lambda_i\le1 \\ & \textstyle\sum_{i=1}^m \lambda_i = m-k \end{aligned} \tag{37}$$

which is a *linear programming* problem that can be solved efficiently. After we get the optimal λ_i's, we can get $\mathbf{M}^*$ and proceeds with the gradient calculation. Note that since $\mathbf{H}$ has a special block-diagonal structure, we only need to update the non-zero entries during the gradient descent process.

[1]In fact, we should compute $\nabla_{\mathbf{H}}\mathcal{J} = \partial_{\mathbf{H}}\mathcal{J} + \partial_{\mathbf{M}}\mathcal{J}\partial_{\mathbf{H}}\mathbf{M}$, however, the constraints on $\mathbf{M}$ makes $\mathcal{J}$ non-differentiable on $\mathbf{M}$, thus we use $\partial_{\mathbf{H}}\mathcal{J}$ instead.

E. Kernelization

In this section, we will present the kernelized form of our algorithm. Specifically, we assume that there are some nonlinear map $\phi : \mathbb{R}^d \to \mathbb{F}$ that maps the data points from the space $\mathbb{R}^d$ to some high-dimensional (possibly infinite) feature space $\mathbb{F}$, and then we will run our task regularization method in $\mathbb{F}$. Let

$$\mathbf{\Phi}_t = [\phi(\mathbf{x}_1^t), \phi(\mathbf{x}_2^t), \cdots, \phi(\mathbf{x}_{n_t}^t)] \tag{38}$$

where $\mathbf{x}_i^t$ is the i-th data point in the t-th task. Then from the representor theorem we know that the classification vector for task t can be written as

$$\mathbf{w}_t^\phi = \mathbf{\Phi}_t\boldsymbol{\alpha}_t \tag{39}$$

where $\boldsymbol{\alpha}_t \in \mathbb{R}^{n_t\times1}$ is the combination coefficient vector for task t. We denote $\mathbf{K}_t$ as the kernel matrix for task t with

$$\mathbf{K}_t(i,j) = \phi(\mathbf{x}_i^t)^\top\phi(\mathbf{x}_j^t) \tag{40}$$

and define two concatenated matrices

$$\begin{aligned} \mathbf{A}^\phi &= diag(\boldsymbol{\alpha}_1^\phi, \boldsymbol{\alpha}_2^\phi, \cdots, \boldsymbol{\alpha}_m^\phi) \in \mathbb{R}^{n\times m} && (41)\\ \mathbf{K} &= diag(\mathbf{K}_1, \mathbf{K}_2, \cdots, \mathbf{K}_m) \in \mathbb{R}^{n\times n} && (42)\end{aligned}$$

Then

$$\begin{aligned} \mathcal{J}_{loss}^\phi &= \|\mathbf{Y} - \mathbf{J}\mathbf{K}\mathbf{A}^\phi\|_F^2 && (43)\\ \mathcal{J}_{intra}^\phi &= tr\left((\mathbf{A}^\phi)^\top(\mathbf{K}+\delta\mathbf{K}\mathbf{L}\mathbf{K})\mathbf{A}^\phi\right) && (44)\\ \mathcal{J}_{inter}^\phi &= tr\left(\mathbf{\Phi}\mathbf{A}^\phi\mathbf{M}(\mathbf{A}^\phi)^\top\mathbf{\Phi}^\top\right) && (45)\end{aligned}$$

where $\mathbf{Y} \in \mathbb{R}^{n\times m}$, $\mathbf{J} \in \mathbb{R}^{n\times n}$, $\mathbf{L} \in \mathbb{R}^{n\times n}$ are defined in Eq.(8), Eq.(10) and Eq.(18) respectively, and $\mathbf{\Phi} = [\mathbf{\Phi}_1, \mathbf{\Phi}_2, \cdots, \mathbf{\Phi}_m]$ is the concatenated data matrix in the feature space. However, in general we cannot compute $\mathbf{\Phi}$ explicitly. Therefore we can transform $\mathcal{J}_{inter}^\phi$ to

$$\mathcal{J}_{inter}^\phi = tr\left(\mathbf{\Phi}\mathbf{A}^\phi\mathbf{M}(\mathbf{A}^\phi)^\top\mathbf{\Phi}^\top\right) = tr\left(\mathbf{A}^\phi\mathbf{M}(\mathbf{A}^\phi)^\top\mathbf{\Phi}^\top\mathbf{\Phi}\right)$$

We can define the *composite* kernel matrix $\widetilde{\mathbf{K}} \in \mathbb{R}^{n\times n}$ as

$$\widetilde{\mathbf{K}} = \begin{bmatrix} \mathbf{K}_1 & \mathbf{K}_{12} & \cdots & \mathbf{K}_{1m} \\ \mathbf{K}_{21} & \mathbf{K}_2 & \cdots & \mathbf{K}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K}_{m1} & \mathbf{K}_{m2} & \cdots & \mathbf{K}_m \end{bmatrix} \tag{46}$$

where $\mathbf{K}_{ij} \in \mathbb{R}^{n_i\times n_j} = \mathbf{\Phi}_i^\top\mathbf{\Phi}_j$ is the kernel matrix between the data points in task i and j. We can perform Choleskey decomposition to $\mathbf{K}$ as

$$\mathbf{K} = \mathbf{F}\mathbf{F}^\top \tag{47}$$

with $\mathbf{F} \in \mathbb{R}^{n\times n}$, then

$$\mathcal{J}_{inter}^\phi = tr\left(\mathbf{F}^\top\mathbf{A}^\phi\mathbf{M}(\mathbf{A}^\phi)^\top\mathbf{F}\right) \tag{48}$$

In this way, we can minimize $\mathcal{J} = \mathcal{J}_{loss} + \lambda_1\mathcal{J}_{intra} + \lambda_2\mathcal{J}_{inter}$ to get the optimal $\mathbf{A}^\phi$ by the same approach as in section II-D.

III. Experiments

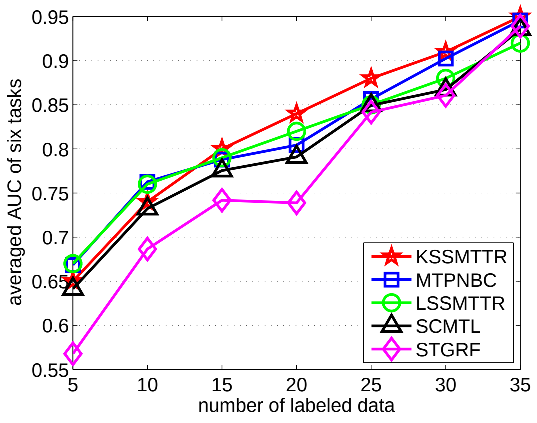
In this section, we will introduce a set of experimental results to show the effectiveness of our method.

A. A Toy Example

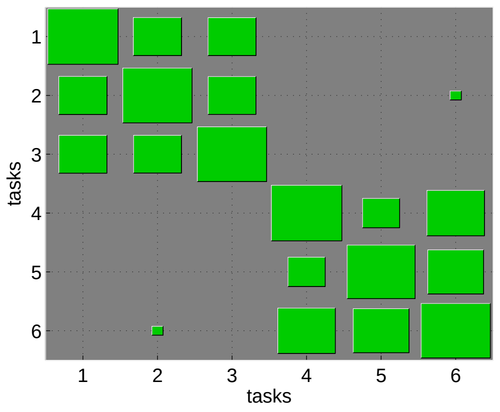
We first construct a toy example to evaluate our method. There are totally 6 tasks with binary classes. The data from each class was generated from a *Gaussian mixture model* (*GMM*). For Tasks 1, 2, and 3 the GMM parameters for class one (denoted with a blue star) are drawn from a three-component mixture defined as follows. Mixture weights (three components): $(0.3, 0.3, 0.4)$; respective two-dimensional means: $(1, 1), (3, 3)$ and $(5, 5)$; and respective covariance matrices $\Sigma_1 = \begin{bmatrix} 0.3 & 0.7 \\ 0.7 & 3.0 \end{bmatrix}$, $\Sigma_2 = \begin{bmatrix} 3.0 & 0.0 \\ 0.0 & 3.0 \end{bmatrix}$ and $\Sigma_3 = \begin{bmatrix} 3.0 & -0.5 \\ -0.5 & 0.3 \end{bmatrix}$. Data for class 2 in Tasks 1, 2, and 3 are drawn from a single Gaussian with mean $(2.5, 1.5)$ and diagonal covariance with symmetric variance 0.5 . For Tasks 4, 5 and 6 the GMM parameters for class one (denoted with blue star) are drawn from a three-component mixture defined as follows. The three mixture weights are $(0.3, 0.3, 0.4)$; the respective two-dimensional means are $(0.5, 0.5)$, $(3, 2)$, and $(5, 5)$ and the respective covariances are Σ_2, Σ_3, Σ_1. In Tasks 4, 5, 6 the data for class two are drawn from a single Gaussian with mean $(2, 3)$, and diagonal covariance with symmetric variance 0.5.

Besides our method, we also implemented the *Multi-Task Parameterized Neighborhood-Based Classification* (*MTPNBC*) method [16], the *Supervised Cluster-based Multi-Task Learning* (*SCMTL*) method [13] and the *Single-Task Gaussian Random Field* (*STGRF*) [21]. Note that MTPNBC is a semi-supervised multi-task learning method with probabilistic classifiers, SCMTL is a supervised multi-task learning method, and STGRF is a semi-supervised single-task learning method. For our *Semi-Supervised Multi-Task Learning with Task Regularization* (*SSMTTR*) method, the regularization parameters λ_1 and λ_2 are all set by 5-fold cross validation from the exponential grid $2^{[-4:1:4]}$. The graph Laplacian for each task is constructed in the same way as in [21]. For the *Kernel SSMTTR* (*KSSMTTR*) method, we just use the Gaussian kernel with its width set by 5-fold cross validation from $2^{[-4:1:4]}$.

Each curve in Figure 2(a) represents the mean *Area Under the ROC Curve* (*AUC*) score over 50 independent trials as a function of the number of labeled data, from which we can clearly observe the superiority of our methods. To gain a better understanding of the clustered structure that SSMTTR finds for the six tasks, we plot the Hinton diagram [12] of the between-task sharing matrix found by the *Linear SSMTTR* (*LSSMTTR*), averaged over the 50 trials. The (i, j)-th element of sharing matrix is equal to $\exp(-\|\mathbf{w}_i - \mathbf{w}_j\|^2/2)$, which is represented by a square in the Hinton diagram, with a larger square indicating a larger value of the corresponding element. The Hinton diagram in Figure 2(b) also shows the agreement of the sharing mechanism of the SSMTTR with the similarity between the tasks.



(a) Averaged AUC vs. number of labeled data



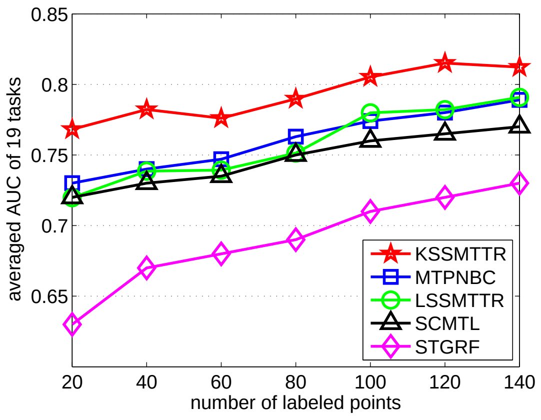
(b) Hinton diagram

Figure 2. (a) Performance of various multi-task learning algorithms on Tasks 1-6 for the data in Figure 1. The horizontal axis is the number of labeled data in each task. The vertical axis is the AUC averaged over the six tasks and 50 independent trials (b) The Hinton diagram of between-task sharing found by our linear semi-supervised multi-task learning method, averaged over 50 independent trials.

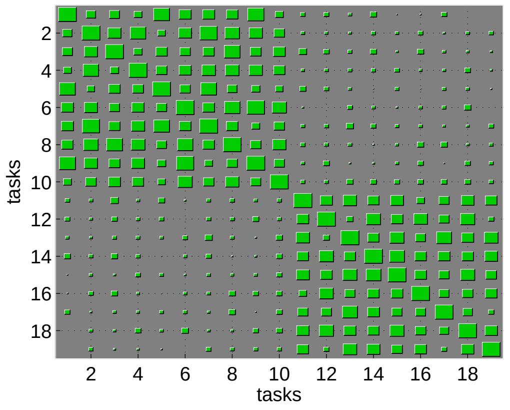
B. Application in Landmine Detection Based on Airborne Radar Data

We consider the remote sensing problem considered in [16], based on data collected from a real landmine field[2]. In this problem there are a total of 19 sets of data, collected from various landmine fields by an actual synthetic-aperture radar system. Each data point is represented by a 9-dimensional feature vector extracted from the data. The class label is binary (mine or false mine).

[2]The data set is downloaded from http://www.ee.duke.edu/~lcarin/LandmineData.zip.



(a) Averaged AUC vs. number of labeled data



(b) Hinton diagram

Figure 3. (a) Performance of various multi-task learning algorithms; (b) The Hinton diagram of between-task sharing found by our linear semi-supervised multi-task learning method, averaged over 100 independent trials.

Each of the 19 data sets defines a task, in which we aim to find landmines with a minimum number of false alarms. Of the 19 data sets, 1-10 are collected at foliated regions and 11-19 are collected at regions that are bare earth or desert. Therefore we expect two dominant clusters of tasks, corresponding to the two different types of ground surface conditions.

We perform 100 independent trials, in each of which we randomly select a subset of data for which labels are assumed available, train the SSMTTR and test the classifiers on the remaining data. The AUC averaged over the 19 tasks is presented in Figure 3(a), as a function of the number of labeled data, where each curve represents the mean calculated from the 100 independent trials (the parameters in our algorithms are also set by cross validation from the same grid as the experiments in last section). The results of MTPNBC, SCMTL and STGRF are also reported, which shows that our algorithm outperforms the competitors significantly. The Hinton diagram of the between-task sharing matrix found by the *Linear SSMTTR* (*LSSMTTR*), averaged over the 100 trials, is shown in Figure 3(b), where the (i, j)-th element of sharing matrix is equal to $\exp(-\|\mathbf{w}_i - \mathbf{w}_j\|^2/2)$. From the figure we can easily discover the task-cluster structure.

IV. Conclusions

In this paper we propose a novel semi-supervised multi-task learning algorithm based on task regularizations. Our algorithm assumes that the multiple tasks form several task clusters and the tasks in the same cluster will share similar classification vectors. After relaxations, we show that our algorithm can be cast into a convex optimization problem and can thus be efficiently solved. Finally the experimental results on both toy and real world problems are presented to show the effectiveness of out method.

Acknowledgement

The work is partially supported by NSF grants NSF grants DMS-0844513 and CCF-0830659.

References

[1] G. M. Allenby and P. E. Rossi. Marketing models of consumer heterogeneity. *Journal of Econometrics*, 89(1-2):57–78, 1998.

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

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

[4] A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying. A spectral regularization framework for multi-task structure learning. In *Advances in Neural Information Processing Systems 20*, pages 25–32. 2008.

[5] N. Arora, G. M. Allenby, and J. Ginter. A hierarchical bayes model of primary and secondary demand. *Marketing Science*, 17(1):29–44, 1998.

[6] B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 4:83–89, 2003.

[7] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[8] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[9] R. Chari, W. W. Lockwood, B. P. Coe, A. Chu, D. Macey, A. Thomson, J. J. Davies, C. Macaulay, and W. L. Lam. Sigma: A system for integrative genomic microarray analysis of cancer genomes. *BMC Genomics*, 7:324, 2006.

[10] F. R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, 1997.

[11] T. Evgeniou and M. Pontil. Regularized multi-task learning. In *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117, 2004.

[12] G. E. Hinton and T. J. Sejnowski. Learning and relearning in boltzmann machines. *Parallel distributed processing: explorations in the microstructure of cognition*, 1:282–317, 1986.

[13] L. Jacob, F. Bach, and J.-P. Vert. Clustered multi-task learning: A convex formulation. In *Advances in Neural Information Processing Systems 21*, 2008.

[14] B. Kågström and P. Poromaa. Lapack-style algorithms and software for solving the generalized sylvester equation and estimating the separation between regular matrix pairs. *ACM Trans. Math. Software*, 22:78–103, 1996.

[15] T. Kato, H. Kashima, M. Sugiyama, and K. Asai. Multi-task learning via conic programming. In *Advances in NIPS 20*, pages 737–744, 2008.

[16] Q. Liu, X. Liao, and L. Carin. Semi-supervised multitask learning. In *NIPS 20*, pages 937–944. 2008.

[17] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[18] D. Sheldon. Graphical multi-task learning. In *NIPS Workshop on Structured Input - Structured Output*, 2008.

[19] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 762–769, 2004.

[20] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag., New York, NY, USA, 1995.

[21] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian random fields and harmonic functions. In *Proceedings of the International Conference on Machine Learning*, pages 912–919, 2003.