What shall I share and with Whom? - A Multi-Task Learning Formulation using Multi-Faceted Task Relationships

Sunil Gupta* † Santu Rana * ‡ Dinh Phung * § Svetha Venkatesh * ¶

Abstract

Multi-task learning is a learning paradigm that improves the performance of "related" tasks through their joint learning. To do this each task answers the question "Which other task should I share with"? This task relatedness can be complex - a task may be related to one set of tasks based on one subset of features and to other tasks based on other subsets. Existing multi-task learning methods do not explicitly model this reality, learning a single-faceted task relationship over all the features. This degrades performance by forcing a task to become similar to other tasks even on their unrelated features. Addressing this gap, we propose a novel multi-task learning model that learns multi-faceted task relationship, allowing tasks to collaborate differentially on different feature subsets. This is achieved by simultaneously learning a low dimensional subspace for task parameters and inducing task groups over each latent subspace basis using a novel combination of L_1 and pairwise L_∞ norms. Further, our model can induce grouping across both positively and negatively related tasks, which helps towards exploiting knowledge from all types of related tasks. We validate our model on two synthetic and five real datasets, and show significant performance improvements over several state-of-the-art multi-task learning techniques. Thus our model effectively answers for each task: What shall I share and with whom?

1 Introduction

Multi-task learning is an established framework to improve predictive performance of related tasks through joint modeling, and has been applied in diverse areas successfully[4, 5, 25, 31]. These techniques rely on sharing useful knowledge between the tasks via inductive transfer [8]. The main idea is to represent each learning task as an instance in a common hypothesis space because learning in a common hypothesis space leads to a better generalization performance (in an average sense) than if learning was performed on tasks independently [3]. This idea is translated in many ways, and some examples include using the same generative processes for task parameters [11, 28, 21], representing the task parameters in a common low-dimensional subspace [2, 18, 19, 14], sharing a small distance from a common classification vector [13], or by modeling task parameters as a combination of a common low rank matrix with a task independent sparse vector [9, 10]. An important assumption is that only "related" tasks should be included. However, in a task pool, not *all* tasks may be related to each other. Thus identifying the "related" set is crucial in multi-task learning.

Early works in MTL naïvely included all tasks for joint learning with a prior assumption that all of them are related. For example, Argyriou et al. [2] proposed a multi-task feature learning model that represents the task parameters in a low-dimensional subspace and uses a mixed norm (L_2/L_1) penalty on the parameters' representations. The L_2 norm helps in avoiding model over-fitting whilst L_1 norm ensures that dimensionality of the subspace is kept small. Similar attempts using a manifold or an infinite subspace were made by [1, 23]. None of these works compute any form of task relatedness. In practice, the assumption of all tasks being related is rarely true and its violation may lead to degradation in performance. To overcome this problem, recent works on MTL compute some form of *task relatedness* and confine the joint learning only within the set of the related tasks. For example, Kang et al. [18] proposed a model that iteratively learns groups of related tasks via integer programming and then construct group-specific subspaces based on multi-task feature learning (as in [2]). To learn the task groups, it requires the number of groups to be specified in advance, which is usually unknown. Further, it is not be able to exploit the knowledge from the negatively related tasks as they may lie in separate groups. In a different approach, Zhang et al. [29] use a covariance matrix to learn the relatedness between tasks, successfully exploiting the knowledge across negatively related tasks. However, both of the above models use a form of task relatedness that we call *'single-faceted task relatedness'*- that is, a task relatedness measure that is computed over all the features and represented by a single number, thus presenting a single view of the relationship between any two tasks.

Consider spam classifiers using keywords as features. Two individuals (akin to two tasks) may share some common

*Center for Pattern Recognition and Data Analytics, Deakin University, Geelong Waurn Ponds Campus, Victoria, Australia.
†sunil.gupta@deakin.edu.au
‡santu.rana@deakin.edu.au
§dinh.phung@deakin.edu.au
¶svetha.venkatesh@deakin.edu.au

interests but differ in others. The presence of some keywords (features) might indicate spam emails for both in their shared interests; however, in areas where their interests are not congruent, their notion of spam may be divergent, and thus keywords indicating spam will be different. A single-faceted task relatedness measure would treat all these keywords (features) in a similar manner, resulting in a value between 0 and 1 (partially related) for these two tasks. Ideally, a multi-task learning method should combine the two individual's classifiers only along the set of keywords where their spam labeling is correlated. However, existing MTL methods, due to their use of single-faceted task relatedness, are not able to treat the two sets of keywords *differentially* and impose joint learning along all the keywords according to the single aggregated task relatedness. One way to solve this problem is to first learn the two subsets of keywords - the first set groups both tasks together, the second subset keep them separate. Then, task relatedness for each of the two subsets needs to be computed separately. Generalizing from this example, each subset of features that yields a unique grouping of tasks is called a *facet*. Several facets may be identified across tasks, giving rise to the concept of *'multi-faceted task relatedness'*. This gives us capacity to group tasks at a finer levels because we can treat the facets differentially in joint learning.

Building on the idea of exploiting multi-faceted task relatedness, we propose a new multi-task learning model that allows the tasks to collaborate differently on different feature subsets. To achieve this, we represent task parameters in a subspace where each basis corresponds to a set of correlated features. To capture intuitive basis vectors (so that they separate positively correlated features from negative ones), we use non-negativity constraints on basis vectors. For each basis, tasks are partitioned automatically into groups of related tasks, and the partitions can vary from one basis to another, resulting in multi-faceted task relationships. The task partition is learnt using a combination of L_1 and pair-wise L_∞ regularization on the task parameters. Task groupings are induced automatically without the need to specify the number of groups. A key feature of our model is its ability to group both positively-related and negatively-related tasks together. This helps towards exploiting knowledge even from negatively-related tasks. This is a direct result of using the pairwise L_∞ norm. On the other hand, the use of L_1 norm ensures that the subspace dimension is kept small. These regularizations are introduced in the OSCAR model [6] that was applied for feature selection. We instead use the regularizations over task parameters in a MTL setting, calling our model OSCAR-MTL. This model is formulated as an optimization problem that combines the least-square loss function with the above-mentioned regularizations. We provide an efficient solution to this optimization problem using alternating direction method of multipliers (ADMM). We validate our model on two synthetic and five real datasets, and show significant performance improvements by our model over several state-of-the-art multi-task learning techniques. Our main contributions are:

- Proposal of the novel concept of multi-faceted task relatedness and utilizing it to construct a new multi-task learning model that allows tasks to collaborate differently on different facets (feature subsets).
- Formulation of the proposed model as an optimization problem, and provision of an efficient solution using alternating direction method of multipliers (ADMM).
- Validation of the effectiveness of our model in several multi-task classification and regression applications, and demonstrating that OSCAR-MTL achieves significant performance improvements over many state-of-the-art multi-task learning techniques.

The significance of our work lies in the fact that we extend the scope of traditional multi-task learning techniques that solve "learning- with whom to share?" to a wider scope of "learning- with whom to share and what?". This way we offer a richer multi-task learning framework.

2 Background

OSCAR: Bondell and Reich [6] proposed a new method of feature selection that uses a combination of L_1 and pairwise L_∞ norms to encourage both sparsity of features and equality of feature weights for correlated features. Due to the use of these norms, the constrained region used by this method takes an octagonal shape. Motivated by this shape and the equality inducing property, this method is known as Octagonal Shrinkage and Clustering Algorithm for Regression (OSCAR). Given an input data matrix $\mathbf{X} \in \mathbb{R}^{N\times M}$ (where each row being a data point) and the outcome vector $\mathbf{y} \in \mathbb{R}^N$, OSCAR learns a sparse regression model by minimizing the following constrained least-square cost function

$$\hat{\beta} = \arg\min_{\beta} ||\mathbf{y} - \mathbf{X}\beta||^2, \quad \text{s. t. } ||\beta||_1 + c\sum_{j<k} \max\left\{|\beta_j|, |\beta_k|\right\} \le h, \tag{2.1}$$

where $c \ge 0$ and $h > 0$ are tuning constants with c controlling the relative weighting of the norms and h controlling the magnitude of the weights. One of the important properties of OSCAR is that it encourages parsimony in the regression model using only a small set of features. This property makes it useful for feature selection. Another important property of OSCAR is that it encourages weights of equal magnitudes for the correlated features. The sign of the weights depends on whether they are positively correlated or negatively correlated. This leads to automatic grouping or clustering of correlated features. In this paper, we use

a similar regularization to that in OSCAR to *automatically* learn the task-grouping for multi-task learning.

3 OSCAR-MTL

We develop a novel multi-task learning (MTL) model with a goal to address the following fundamental issues in real-world data with multiple tasks:

1. Most real-worlds tasks are only *partially* related to each other and thus need to collaborate differentially on different feature subsets.

2. Task relatedness in real-world data varies: tasks can be positively related, negatively related or unrelated. A successful MTL model should exploit *positive* and *negative* relatedness equally and automatically separate the unrelated tasks from joint learning. Doing this gets harder in presence of the *first* problem.

Our model addresses the *first* issue by learning a *multi-faceted* task relationship. For this, it simultaneously learns a low dimensional subspace for task parameters and induces independent task-grouping/partitioning for each latent subspace basis (*facet*). Due to this property, our model offers finer and flexible control in learning different aspects of a problem from different tasks. The *second* issue is dealt by inducing a grouping that tries to bring both positively and negatively related tasks closer (or equal) in their magnitude while retaining their signs.

Let us assume we have T_0 tasks, indexed as $t = 1,\ldots,T_0$. The labeled examples of t-th task are denoted as $\{(\mathbf{x}_{ti}, y_{ti}) \mid i = 1,\ldots,N_t\}$ where feature vector $\mathbf{x}_{ti} \in \mathbb{R}^M$, the output $y_{ti} \in \mathbb{R}$ (for regression) and $y_{ti} \in \{-1,+1\}$ (for classification). Collectively, we denote the data of t-th task by $\mathbf{X}_t = (\mathbf{x}_{t1},\ldots,\mathbf{x}_{tN_t})^T$ and $\mathbf{y}_t = (y_{t1},\ldots,y_{tN_t})^T$ and its task parameter by β_t. Since our goal is to model T_0 tasks jointly, we first represent their task parameters in a low-dimensional subspace, which is spanned by columns of a matrix $\mathbf{U}$. Formally, β_t is represented in the subspace as $\beta_t = \mathbf{U}\theta_t$. We stack the representation vectors θ_t's in a matrix as $\Theta = (\theta_1,\ldots,\theta_{T_0})$. We denote the k-th row of matrix Θ as $\theta_{(k)}$. Each column of the matrix $\mathbf{U}$ serves as a basis for the subspace and captures a set of features along which the weights of the task parameters $\beta_{1:T_0}$ are correlated. We also impose a non-negativity constraint on $\mathbf{U}$ to obtain intuitive topic-like bases similar to non-negative matrix factorization (NMF). We note that subspace learning can be guided by data such that each basis captures an unique *facet* of the problem. Collectively, multiple bases provide *multiple facets* (or sets of features) along which tasks can collaborate differentially.

To induce facet-specific task-grouping, our model imposes a regularization on each row of Θ (i.e. $\theta_{(k)}$) using a combination of L_1 and pairwise L_∞ penalties. The pairwise L_∞ regularization is used to induce grouping of both positively-related and negatively-related tasks. This helps towards exploiting knowledge even from negatively-related tasks. The L_1 norm (acting along both rows and columns of Θ) is used to ensure that unrelated tasks are separated out of joint learning and the subspace dimension is kept small. We formulate our model as the following objective function

$$\min_{\mathbf{U}\geq 0,\theta_{1:T_0}} J\left(\mathbf{U},\theta_{1:T_0}\right) = \frac{1}{2}\sum_t ||\mathbf{X}_t\mathbf{U}\theta_t - \mathbf{y}_t||^2 + \eta||\mathbf{U}||_F^2 + \sum_{k=1}^{K}\left[\lambda_1||\theta_{(k)}||_1 + \lambda_2 \sum_{t,t':t<t'} \max\left(|\theta_{kt}|,|\theta_{kt'}|\right)\right] \tag{3.2}$$

where η,λ_1,λ_2 are regularization parameters and collectively denoted as $\wr = \{\eta,\lambda_1,\lambda_2\}$. The regularization term of the above formulation is motivated from OSCAR feature selection where this regularization is used to automatically discover correlated feature groups [6]. An extension for undirected graphs is proposed in [27]. We note that *none of these formulations* have considered such regularization for multi-task learning. Since our model uses octagonal shrinkage as in OSCAR, we call it as **"OSCAR-MTL"**.

The cost function of (3.2) has terms involving pairwise L_∞ norms. With the use of simple algebra ([6, 27]) we can reduce these terms to a transformed L_1 norm term leading to the following cost function

$$J\left(\mathbf{U},\theta_{1:T_0}\right) = \frac{1}{2}\sum_t ||\mathbf{X}_t\mathbf{U}\theta_t - \mathbf{y}_t||^2 + \eta||\mathbf{U}||_F^2 + R(\Theta) \tag{3.3}$$

where we define $R(\Theta) \triangleq \sum_{k=1}^{K}\left[\lambda_1||\theta_{(k)}||_1 + \lambda_2||\mathbf{D}\theta_{(k)}||_1\right]$ and the matrix $\mathbf{D}$ arises due to the following decomposition property of the max operator:

$$\max\left(|\theta_{kt}|,|\theta_{kt'}|\right) = 0.5\left(|\theta_{kt}+\theta_{kt'}| + |\theta_{kt}-\theta_{kt'}|\right)$$

The right hand side terms can be written as $|\mathbf{f}^T\theta_{(k)}| + |\mathbf{g}^T\theta_{(k)}|$ where $\mathbf{f},\mathbf{g}$ (both of size $T_0 \times 1$) are sparse vectors with all zero elements except the t, t' elements, which are $\mathbf{f}_t = \mathbf{f}_{t'} = 0.5$ and similarly, $\mathbf{g}_t = -\mathbf{g}_{t'} = 0.5$. The matrix $\mathbf{D}$ is obtained by row-stacking the pair of vectors $[\mathbf{f},\mathbf{g}]^T$ culminating in a matrix sized $T_0\left(T_0-1\right)\times T_0$. This matrix does not depend on data or model parameters and can be constructed *in advance*.

3.1 Learning and Optimization The cost function of (3.3) contains a least-square loss term that is used to learn task parameters from data and the regularization terms that are used to couple task parameters of multiple tasks. To learn the proposed model, we need to estimate the matrix $\mathbf{U}$ and the task parameters $\theta_{1:T_0}$. The cost function of (3.3) is jointly non-convex in $\mathbf{U}$ and $\theta_{1:T_0}$. However, it reduces to a convex function in $\mathbf{U}$ when $\theta_{1:T_0}$ are fixed or *vice versa*. Given $\theta_{1:T_0}$, the optimal solution of $\mathbf{U}$ takes a closed form expression if there are no non-negative constraints. However, since

our formulation uses non-negative constraints on $\mathbf{U}$, we optimize it in an iterative fashion. Given $\mathbf{U}$, the cost function involving $\theta_{1:T_0}$ becomes a quadratic function along with non-smooth L_1 penalty. There are several ways to optimize this cost function. A direct way is to convert the L_1 regularizations to appropriate constraints and use optimization solvers such as 'quadratic programming with inequality contraints'. However, the computational cost of such techniques is often high. Another promising way is to use the accelerated gradient descent (AGD) for OSCAR [30]. However, due to the simultaneous involvement of both row and columns of Θ matrix in the optimization, it is not straight-forward to use this method. This leads us to use the popular alternating direction method of multipliers (ADMM) [7]. ADMM is a general method of decomposing a large optimization problem into a set of smaller problems and then combining their solutions to get the solution of the main problem. In the following, we detail the optimization process.

Optimizing U given $\theta_{1:T_0}$ Given $\theta_{1:T_0}$, the optimization problem involving $\mathbf{U}$ can be re-written as

$$\min_{\mathbf{U}\geq 0} \frac{1}{2}\sum_t ||\mathbf{X}_t\mathbf{U}\theta_t - \mathbf{y}_t||^2 + \eta||\mathbf{U}||_F^2. \tag{3.4}$$

A naïve way to solve the above optimization problem is by using constrained optimization methods such as Lagrange multiplier or projected gradients. However, these methods usually require a step size for the gradient descent. Choice of a step size can be easily avoided by following a multiplication update for $\mathbf{U}$ using an optimization strategy similar to NMF [20] or semi-NMF [12]. Using this strategy, we derive a multiplicative update of $\mathbf{U}$. The Lagrangian of the cost function of (3.4) can be written as

$$L(\mathbf{U}) = \frac{1}{2}\sum_t ||\mathbf{X}_t\mathbf{U}\theta_t - \mathbf{y}_t||^2 + \frac{\eta}{2}||\mathbf{U}||_F^2 + \mathrm{tr}\left(\Lambda\mathbf{U}^T\right),$$

where Λ_{mk} is the Lagrange multiplier for the constraint $\mathbf{U}_{mk} \geq 0$. The derivative of L with respect to $\mathbf{U}$ is

$$\nabla_{\mathbf{U}} L = \sum_t \mathbf{X}_t^T\left(\mathbf{X}_t\mathbf{U}\theta_t - \mathbf{y}_t\right)\theta_t^T + \eta\mathbf{U} + \Lambda.$$

Using the *Karush-Kuhn-Tucker* conditions $\Lambda_{mk}\mathbf{U}_{mk} = 0$, we get the following equation for $\mathbf{U}_{mk}$

$$\left[\sum_t \mathbf{X}_t^T\left(\mathbf{X}_t\mathbf{U}\theta_t - \mathbf{y}_t\right)\theta_t^T + \eta\mathbf{U}\right]_{mk}\mathbf{U}_{mk} = 0.$$

Applying 'vec' operator of a matrix, we can write

$$\left[\left(\sum_t \mathbf{A}_t\mathbf{U}\mathbf{B}_t\right)_{vec} - \left(\sum_t \mathbf{C}_t\right)_{vec} + \eta\mathbf{U}_{vec}\right]_j [\mathbf{U}_{vec}]_j = 0,$$

where we use the notation, $\mathbf{U}_{vec}$ to denote vec$(\mathbf{U})$ and define $\mathbf{A}_t \triangleq \mathbf{X}_t^T\mathbf{X}_t$, $\mathbf{B}_t \triangleq \theta_t\theta_t^T$ and $\mathbf{C}_t \triangleq \mathbf{X}_t^T\mathbf{y}_t\theta_t^T$. The index j is one-dimensional index of the matrix elements after using the vec operator and we have $j = M(k-1)+m$. The above expression can be further simplified as

$$\left[\mathbf{P}\mathbf{U}_{vec} - \mathbf{Q}_{vec} + \eta\mathbf{U}_{vec}\right]_j [\mathbf{U}_{vec}]_j = 0,$$

where we define $\mathbf{P} \triangleq \sum_t \mathbf{A}_t \otimes \mathbf{B}_t$, $\mathbf{Q} \triangleq \sum_t \mathbf{C}_t$. Using a notation, $\mathbf{v}_j = \mathbf{v}_j^+ - \mathbf{v}_j^-$, where $\mathbf{v}_j^+ = 0.5\left(|\mathbf{v}_j| + \mathbf{v}_j\right)$ and $\mathbf{v}_j^- = 0.5\left(|\mathbf{v}_j| - \mathbf{v}_j\right)$ (both non-negative matrices), the above equation can be written as

$$\left[-\mathbf{P}^-\mathbf{U}_{vec} - \mathbf{Q}_{vec}^+ + \mathbf{P}^+\mathbf{U}_{vec} + \mathbf{Q}_{vec}^- + \eta\mathbf{U}_{vec}\right]_j [\mathbf{U}_{vec}]_j = 0,$$

which leads to the following update for $\mathbf{U}$

$$[\mathbf{U}_{vec}]_j \leftarrow [\mathbf{U}_{vec}]_j \sqrt{\frac{\left[\mathbf{P}^-\mathbf{U}_{vec} + \mathbf{Q}_{vec}^+\right]_j}{\left[\mathbf{P}^+\mathbf{U}_{vec} + \mathbf{Q}_{vec}^- + \eta\mathbf{U}_{vec}\right]_j}} \tag{3.5}$$

THEOREM 3.1. *The cost function $J\left(\mathbf{U}, \theta_{1:T_0}\right)$ of (3.3) is monotonically decreasing under the update rule of (3.5) for a fixed $\theta_{1:T_0}$.*

Proof. The poof closely follows along the lines of the proof of Theorem 1(A) in semi-NMF [12]. □

Optimizing $\theta_{1:T_0}$ given U Given $\mathbf{U}$, the cost function involving $\theta_{1:T_0}$ becomes a quadratic function along with non-smooth L_1 penalty and can be written as

$$J\left(\mathbf{U}, \theta_{1:T_0}\right) = \frac{1}{2}\sum_t ||\mathbf{X}_t\mathbf{U}\theta_t - \mathbf{y}_t||^2 + R(\Theta) \tag{3.6}$$

The term $R(\Theta)$ is non-smooth in $\theta_{1:T_0}$ due to the L_1 penalty terms. To deal with this, we use ADMM, which can decouple non-smooth constraints from a smooth loss function. ADMM uses an augmented Lagrangian, which in our case is

$$L_\rho(\Theta, W, Z, \mu, \nu) = \frac{1}{2}\sum_t ||\mathbf{X}_t\mathbf{U}\theta_t - \mathbf{y}_t||^2 + R(\Theta, W, Z, \mu, \nu)$$

where the augmented regularization terms, denoted as $R(\Theta, W, Z, \mu, \nu)$, are given by

$$R(\Theta, W, Z, \mu, \nu) = \lambda_1||\mathbf{W}_{(k)}||_1 + \lambda_2||\mathbf{Z}_{(k)}||_1 + \mu_{(k)}^T\left(\mathbf{W}_{(k)} - \theta_{(k)}\right)$$
$$+ \nu_{(k)}^T\left(\mathbf{Z}_{(k)} - \mathbf{D}\theta_{(k)}\right) + \frac{\rho}{2}||\mathbf{W}_{(k)} - \theta_{(k)}||^2 + \frac{\rho}{2}||\mathbf{Z}_{(k)} - \mathbf{D}\theta_{(k)}||^2$$

In the above equation, we have introduced auxiliary variables W, Z with the constraint that $\mathbf{W}_{(k)} - \theta_{(k)} = 0$ and $\mathbf{Z}_{(k)} - \mathbf{D}\theta_{(k)} = 0$. Further, we have $\mu_{(k)} \in \mathbb{R}^{T_0}$ and $\nu_{(k)} \in \mathbb{R}^{T_0(T_0-1)}$, which are the augmented Lagrange multipliers and a non-negative parameter ρ, which is used to enhance the numerical stability of the algorithm. In our implementation, ρ is set to 1. ADMM finds the solution by iteratively optimizing the above Lagrangian over Θ, W, Z, μ and ν. In the following we present update rules for $\theta_{1:T_0}$ and other auxiliary variables.

Algorithm 1 The proposed OSCAR-MTL

Input: Multi-task data $\{\mathbf{X}_t, \mathbf{y}_t\}_{t=1}^{T_0}$, parameter set $\wr$ and subspace dimension K.
Output: Task parameters $\beta_{1:T_0}$, matrix $\mathbf{U}$ and matrix Θ.
Initialization: Initialize nonnegative matrix $\mathbf{U}$ and matrix Θ randomly. Set $\mathbf{W} = \Theta$ and $\mathbf{Z} = \Theta\mathbf{D}^T$.
repeat
  update $\mathbf{U}$ using Eq. (3.5) until it converges.
  for $k = 1 : K$ **do**
    update $\theta_{(k)}$ by solving Eq. (3.7).
    update $\mathbf{W}_{(k)}, \mathbf{Z}_{(k)}$ using Eq. (3.8)-(3.9).
    update $\mu_{(k)}, \nu_{(k)}$ using Eq. (3.10)-(3.11).
  end for
until convergence

- **Update $\theta_{(k)}$ given other variables:** Optimum solution of $\theta_{(k)}$ given other variables has a closed form and is obtained by solving the following equation

$$[\Gamma + \mathrm{diag}(\mathbf{e}_k)]\,\theta_{(k)} = \mathbf{r}_{(k)} - \mathbf{s}_{(k)} \tag{3.7}$$

where we first define $\Gamma \triangleq \rho\left(\mathbf{I} + \mathbf{D}^T\mathbf{D}\right)$, $\mathbf{E}^t \triangleq \mathbf{U}^T\mathbf{X}_t^T\mathbf{X}_t\mathbf{U}$, $\mathbf{b}^t \triangleq \mathbf{U}^T\mathbf{X}_t^T\mathbf{y}_t$ and then use $\mathbf{E}^t$, $\mathbf{b}^t$ to further define $\mathbf{e}_k \triangleq \left[E^1_{kk}, \ldots, E^{T_0}_{kk}\right]^T$, $\mathbf{s}_{(k)} \triangleq \left[\left(E^1\theta_1\right)_k, \ldots, \left(E^{T_0}\theta_{T_0}\right)_k\right]^T - \left(\theta_{(k)} \odot \mathbf{e}_k\right)$, and $\mathbf{r}_{(k)} \triangleq \left[b^1_k, \ldots, b^{T_0}_k\right]^T + \mu_{(k)} + \mathbf{D}^T\nu_{(k)} + \rho\left(\mathbf{W}_{(k)} + \mathbf{D}^T\mathbf{Z}_{(k)}\right)$. We note that the matrix $\Gamma + \mathrm{diag}(\mathbf{e}_k)$ is diagonal and has full rank. Therefore, Eq. (3.7) decouples in all variables and does not involve any matrix inverse.

- **Update $\mathbf{W}, \mathbf{Z}$ given other variables:** Optimum solution of $\mathbf{W}$ can be obtained by minimizing the following

$$\min_{\mathbf{W}_{(k)}} \lambda_1 ||\mathbf{W}_{(k)}||_1 + \frac{\rho}{2}||\mathbf{W}_{(k)} - \theta_{(k)}||^2 + \mu_{(k)}^T\left(\mathbf{W}_{(k)} - \theta_{(k)}\right),$$

which can be reduced to the following standard form that is amenable to using proximal gradient operators

$$\min_{\mathbf{W}_{(k)}} \frac{\lambda_1}{\rho}||\mathbf{W}_{(k)}||_1 + \frac{1}{2}||\mathbf{W}_{(k)} - \left(\theta_{(k)} - \frac{\mu_{(k)}}{\rho}\right)||^2.$$

The closed form solution to the above non-smooth minimization problem using proximal operators is given by

$$\mathbf{W}_{(k)} = \mathscr{H}_{\lambda_1/\rho}\left(\theta_{(k)} - \frac{\mu_{(k)}}{\rho}\right), \tag{3.8}$$

where $\mathscr{H}_\lambda(\theta)$ is the *soft-thresholding* operator given by

$$\mathscr{H}_\lambda(\theta) = \max(|\theta| - \lambda, 0)\,\mathrm{sign}(\theta).$$

The optimum solution of $\mathbf{Z}$ given other variables can be obtained similarly and is given as

$$\mathbf{Z}_{(k)} = \mathscr{H}_{\lambda_2/\rho}\left(\mathbf{D}\theta_{(k)} - \frac{\nu_{(k)}}{\rho}\right) \tag{3.9}$$

- **Update μ, ν given other variables:** The Lagrange multipliers μ, ν can be updated as following

$$\mu_{(k)} \leftarrow \mu_{(k)} + \rho\left(\mathbf{W}_{(k)} - \theta_{(k)}\right) \tag{3.10}$$

$$\nu_{(k)} \leftarrow \nu_{(k)} + \rho\left(\mathbf{Z}_{(k)} - \mathbf{D}\theta_{(k)}\right) \tag{3.11}$$

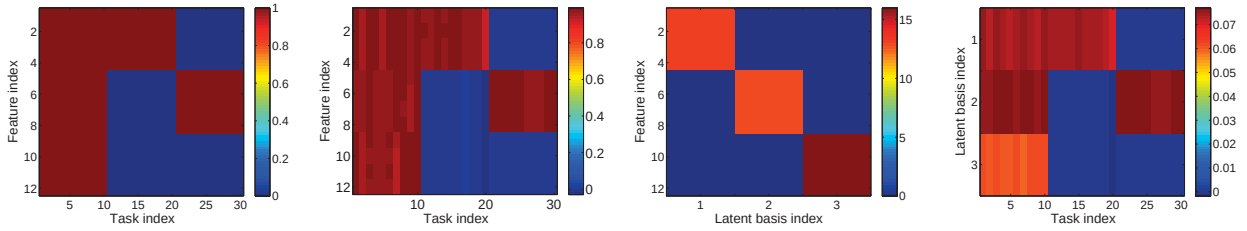
A step-by-step procedure of the proposed OSCAR-MTL is provided in Algorithm 1. We analyze the computational cost of the algorithm. Computational complexity of computing $\mathbf{X}_t^T\mathbf{X}_t$ and $\mathbf{X}_t^T\mathbf{y}_t$ is $\mathscr{O}\left(M^2N_t\right)$ and is required only once at the start of the algorithm. The costliest computation is incurred for updating $\mathbf{U}$, which is $\mathscr{O}\left(KM^2T_0^2\right)$ per iteration. The complexity for updating Θ is $\mathscr{O}\left(KT_0\right)$ per iteration. Thus the total computational complexity of Algorithm 1 per iteration is $\mathscr{O}\left(KM^2T_0^2\right)$.

4 Experiments

We perform extensive experiments using both synthetic and real datasets to validate the effectiveness of OSCAR-MTL. Synthetic data in a controlled scenario is used to illustrate the behavior of the model with different configurations of task-grouping with respect to the underlying subsets of features. A total of 2 classification and 3 regression real datasets are used to measure the effectiveness of our model for real world tasks. Performance is compared with the following state-of-the-art multi-task learning models:

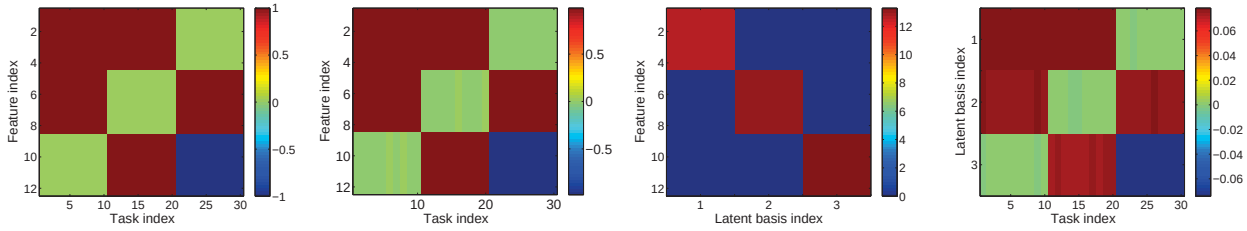
- Multi-task Feature Learning (**MTFL**) [2]: This method assumes that all the tasks are related and represents the task parameters in a low dimensional subspace. There is no grouping of tasks.
- Grouped Multi-task Learning (**GMTL**) [18]: This method extends MTFL by clustering the tasks into a set of groups and then modeling tasks of each group jointly. Unlike OSCAR-MTL, it requires the number of groups to be pre-specified. Further, it learns *a single* task-grouping irrespective of feature subsets. In addition, it can not exploit the knowledge from negatively related tasks as such tasks are not grouped together.
- Multi-task Relationship Learning (**MTRL**) [29]: This method first learns the relationship among the tasks through a covariance matrix and then uses the task relationship for jointly learning task parameters. This method can learn both positive and negative task relatedness. However, this method only learns a single-faceted relationship between the tasks.

In addition to the above baselines, comparison is also performed with single-task learning (**STL**), which learns each task independent of other tasks and a baseline that we call **ATL** (All Task Learning), which pools data from all tasks together and learns a single model for all the tasks. For all the models including OSCAR-MTL, the best parameters are learnt using a grid search through 5-fold cross validation.



(a) (b) (c) (d)

Figure 1: Experimental results for Synthetic-I: (a) task parameters (*true*) (b) task parameters (*estimated*) (c) matrix $\mathbf{U}$ (*estimated*) (d) multi-faceted task-grouping (*estimated*). In (c) and (d), latent bases are permuted for easy viewing.



(a) (b) (c) (d)

Figure 2: Experimental results for Synthetic-II: (a) task parameters (*true*) (b) task parameters (*estimated*) (c) matrix $\mathbf{U}$ (*estimated*) (d) multi-faceted task-grouping (*estimated*). In (c) and (d), latent bases are permuted for easy viewing.

We evaluate the performance on classification tasks using the *area under ROC curve* (AUC) and on regression tasks using *root mean square error* (RMSE). AUC is chosen as the measure since our classification datasets are imbalanced and it is known that AUC is a more robust measure than Accuracy for imbalanced datasets[17].

4.1 Experiments with Synthetic Data We create 30 synthetic tasks and divide them into 3 task groups such that tasks '1-10' belong to task group-1, tasks '11-20' belong to task group-2 and the tasks '21-30' belong to task group-3. The parameters (β_t) for all the tasks are defined in a 12-dimensional space. Following this, we create *two* synthetic datasets. Both the datasets simulate scenarios where task relatedness varies depending on subsets of features i.e. task relations are multi-faceted. The first dataset includes *positively related* or *unrelated* tasks but does not contain any *negatively related* tasks. The second dataset simulates a more general scenario having all types of task relatedness (*positively related, unrelated and negatively related*).

- **Synthetic-I:** Along the first 4 features the task group-1 is related to the task group-2 (but unrelated with the task group-3), along the next 4 features the task group-1 is related to the task group-3 (but unrelated with the task group-2) while along the last 4 features the task group-2 and task group-3 are related but the task group-1 is unrelated to them. This dataset simulates a situation where task relations are multi-faceted *i.e.* relatedness varies depending on the feature subsets.
- **Synthetic-II:** This is identical to the Synthetic-I along the first 8 features. However, along the last 4 features, the task group-2 and task group-3 are now *negatively* related and the task group-1 is unrelated.

For both datasets, the feature matrix for t-th task, i.e. $\mathbf{X}_t = \left[\mathbf{x}_1^t, \ldots, \mathbf{x}_{N_t}^t\right]$ is generated using a multi-variate normal distribution as $\mathbf{x}_i^t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Given i-th data vector $\mathbf{x}_i^t$ and its task parameter β_t, the label y_i^t is generated as

$$y_i^t = \text{sign}\left(\beta_t^T \mathbf{x}_i^t + \varepsilon_i^t\right),\ \varepsilon_i^t \sim \mathcal{N}(0, 0.1). \tag{4.12}$$

For all our synthetic experiments, we randomly split the data in two sets: 70% for training and the remainder for test. The results are averaged over 40 random training-test splits. As seen in Figure 3, OSCAR-MTL usually converges in 20-30 iterations. Figure 1 (a-b) shows the *true* task parameters and those estimated by OSCAR-MTL for Synthetic-I dataset. We can see that upto some proportionality constant, these parameters are estimated accurately. Figure 1 (c-d) shows the latent subspace bases and the subspace representation of the task parameters estimated by OSCAR-MTL. In Figure 1 (c), we can see that the features are correctly segmented, i.e. the first basis is mostly about the first 4 features, the next basis is mostly about the next 4 features and the third basis is mostly about the last 4 features. When looking at the subspace representation of the task parameters (in Figure 1 (d)), we can see that along the first subspace basis (i.e. the first 4 features), the weights of the first 10 tasks are very similar to the next 10 tasks. The correct recovery is also obtained for the other two bases. Figure 2 shows a

Method	*Synthetic-I*	*Synthetic-II*
STL	0.827 (0.003)	0.802 (0.005)
ATL	0.884 (0.002)	0.823 (0.003)
MTFL	0.876 (0.002)	0.856 (0.004)
GMTL	0.868 (0.002)	0.847 (0.004)
MTRL	0.883 (0.002)	0.858 (0.004)
OSCAR-MTL	**0.945** (0.002)	**0.949** (0.004)

Table 1: Comparison of classification performance of OSCAR-MTL with various methods in terms of the *area under ROC curve* (AUC) for *Synthetic* data. AUC performance is averaged over 40 random training-test splits and the numbers in parenthesis are the respective standard errors.

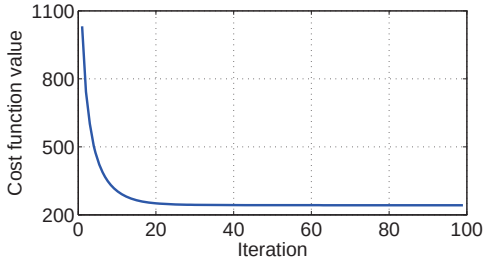


Figure 3: Convergence plot of OSCAR-MTL for Synthetic-I.

similar results for Synthetic-II dataset. We note that both task parameters and their subspace representations (see Figures 2 (b) and (d)) correctly capture the negative task relatedness.

In Table 1 we present a comparison between OSCAR-MTL and the baseline methods. Clearly, OSCAR-MTL significantly outperforms all the baselines. Comparing between the results of Synthetic-I and Synthetic-II, we note that all the baseline methods including MTFL, GMTL and MTRL suffer a degradation in their performance due to the negative correlation between task group-2 and task group-3 along the last 4 features. In contrast, the performance of OSCAR-MTL is marginally improved indicating that it can exploit both positive and negative relatedness equally due to the use of multi-faceted task relationship. Although MTRL claims to learn negative task relatedness, it fails to capture true task relations due to using a single-faceted measure.

4.2 Experiments with Real Data We use the following *classification* and *regression* datasets.

Landmine Data (Classification): This dataset is created from radar images collected from 19 landmine fields, used previously in [26]. Each data instance is a 9-dimensional representation of each image formed by concatenating different image based features. The task is to detect images with landmines. Treating each landmine field as a task, we jointly model them via multi-task learning.

Cancer Data (Classification): This dataset is collected from a hospital in Australia (Ethics approval #12/83). It contains records of patients who visited the hospital during 2010-2012 and suffered from one or many of 11 different cancer types (cf. Figure 5(b)). The records include a variety of information such as demographics, medical conditions using WHO ICD-10 codes and tumor-specific information. This information leads to a total of 528 features. The total number of patients over all cancer types is 668[16]. The task is to build a 1-year mortality prediction model for each cancer type. Since the number of instances for many of the cancer types are small, the predictive performance of independently trained models is poor. To improve the performance, we jointly model them via multi-task learning.

Computer Survey Data (Regression): This dataset was introduced in [22] and later used by [2] for multi-task learning. This dataset contains ratings of 20 computers by 190 students. Each computer was rated by students based on 13 binary features (cf. Figure 4(a)) and assigned a score on a scale of 0-10 indicating their likelihood of buying the product. We treat ratings by each student as a task, giving rise to 190 tasks. As these tasks are related, we jointly model them under the setting of multi-task learning.

School Data (Regression): This dataset comes from the Inner London Education Authority and has been widely used [2, 19]. It consists of examination scores of 15362 students from 139 secondary schools in London during 1985-1987. Each student is represented as a 26-dimensional feature vector containing year of examination, school and student-specific attributes. We treat the prediction of examination score for each school as a task.

Parkinson Data (Regression): This dataset consists of a range of biomedical voice measurements taken from 42 people with early-stage Parkinson's disease [24]. A total of 19 features includes subject's age, gender, time interval from baseline recruitment date, and 16 voice measures. There are a total of 5875 instances collected over time from all patients. The task is to accurately predict Parkinson's disease symptom score (motor UPDRS). We model it using multi-task learning treating each subject as a task.

For all classification datasets, we randomly split the data in two parts: 70% instances for training and the remainder for testing. The results are averaged over 40 random training-test splits. For school and computer dataset, following [15], we use 70% instances for training and the remainder for testing. For Parkinson, since the number of examples per task is not small, to show the efficacy of multi-task learning, we use a smaller training set: 20% instances for training and the remainder for testing. The performance is averaged over 40 trials. Convergence plots of OSCAR-MTL for all real datasets are similar to that of synthetic-I and not shown.

4.2.1 Experimental Results Table 2 presents a comparison of our proposed OSCAR-MTL with STL, ATL and three other MTL algorithms. OSCAR-MTL consistently and clearly outperforms all the baselines for both regression

Method	AUC (Classification)		RMSE (Regression)		
	Landmine	*Cancer*	*Computer*	*School*	*Parkinson*
STL	0.761 (0.02)	0.692 (0.005)	2.22 (0.00)	10.07 (0.00)	1.70 (0.05)
ATL	0.741 (0.01)	0.793 (0.004)	2.21 (0.00)	10.29 (0.00)	7.36 (0.09)
MTFL	0.762 (0.02)	0.703 (0.004)	2.06 (0.00)	9.81 (0.00)	2.85 (0.08)
GMTL	0.755 (0.02)	0.724 (0.004)	2.07 (0.01)	9.74 (0.02)	2.93 (0.10)
MTRL	**0.781** (0.01)	0.750 (0.004)	2.04 (0.00)	9.83 (0.00)	1.37 (0.02)
OSCAR-MTL	0.779 (0.01)	**0.827 (0.004)**	**1.71 (0.05)**	**9.73 (0.03)**	**1.14 (0.07)**

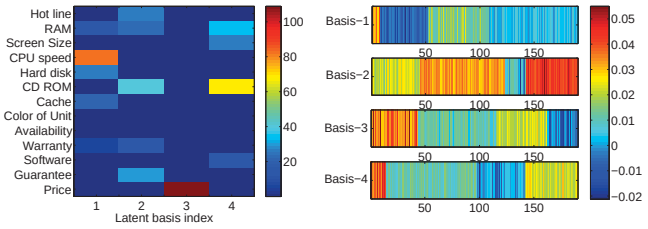
Table 2: Experimental results for real datasets: The results are reported in terms of *area under ROC curve* (AUC) for classification datasets and *root mean square error* (RMSE) for regression datasets. Performance is averaged over 40 random training-test splits and the numbers in parenthesis are their respective standard errors.

and classification applications except the Landmine dataset where MTRL achieves the best performance and OSCAR-MTL is a close second (the standard error bars for OSCAR-MTL and MTRL overlaps). Usually, MTRL is the closest contender to OSCAR-MTL for most of the datasets. One of the reasons is that similar to OSCAR-MTL, MTRL is also a capable model that can exploit negative task relatedness. However, since MTRL uses a single-faceted task relatedness computed using all the features, it may sometimes bring the tasks closer even along unrelated features causing degradations in performance. The performance of MTFL and GMTL is usually lower than that of MTRL and OSCAR-MTL, except for the school dataset where GMTL sits at the second place after OSCAR-MTL. The performance of GMTL and MTFL are usually similar. Interestingly, GMTL, in spite of learning task-grouping, fails to perform better than MTFL. This may be the result of task-grouping derived using a limited single-faceted view of task relations. In contrast, OSCAR-MTL induces multi-faceted task-grouping and achieves superior performance.

To illustrate further, we use the results from the Computer dataset and show how different task-grouping exist for different facets of the problem. We show the subspace matrix $\mathbf{U}$ and the *magnitude* of task representations Θ in Figure 4. The matrix $\mathbf{U}$ depicted in Figure 4 (a) shows 4 different facets discovered by OSCAR-MTL. The first facet is mainly about 'CPU speed', 'Hard disk' and 'Cache'. The second facet is about 'CD-ROM', along with 'Guarantee' and 'Hot line'. The third facet is about 'Price'. The fourth and the last facet is about 'CD-ROM', 'RAM', 'Screen size' and 'Software'. We can see that these facets are quite different from each other and are intuitively meaningful. From Figure 4 (b), we clearly see that there exist different grouping of students for each facet, implying that joint learning needs to be done with different students for different facets. Similar task-grouping for other datasets are shown in Figure 5. As seen, tasks collaborate differentially on different facets.

5 Conclusion

We propose a novel multi-task learning method exploiting multi-faceted task relationships. Our model simultaneously learns a low dimensional subspace for task parameters and induces task-grouping on each basis separately. The task groupings are learnt by using a novel combination of L_1 and pairwise L_∞ norms. Further, the model is able to group both positively and negatively-related tasks together, enabling learning from all types of related tasks. We formulate the model as an optimization problem and provide an efficient solution. We validate our model on five real datasets, and show significant performance improvements over several state-of-the-art multi-task learning methods.



(a) (b)

Figure 4: The latent bases ($K = 4$) and the basis-specific task grouping discovered by OSCAR-MTL for *computer* dataset. In (b), for easy viewing of the different task groups, we have re-ordered the tasks along each latent basis.

References

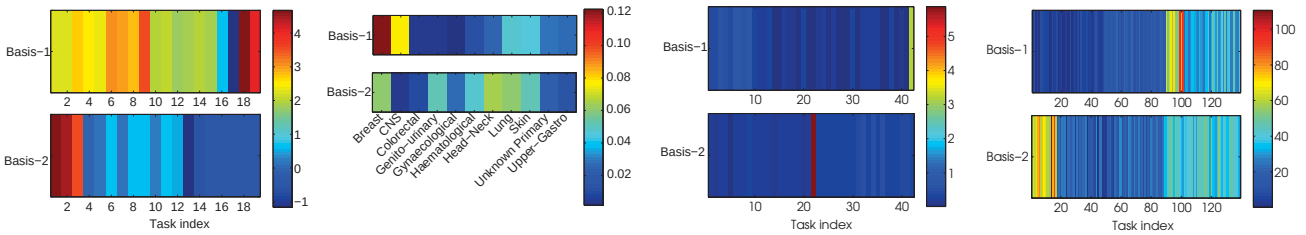
[1] A. Agarwal, S. Gerber, and H. Daume. Learning multiple tasks using manifold regularization. In *Advances in neural information processing systems*, pages 46–54, 2010.

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

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

[4] J. Bi, T. Xiong, S. Yu, M. Dundar, and R B. Rao. An improved multi-task learning approach with applications in medical diagnosis. In *Machine Learning and Knowledge Discovery in Databases*, pages 117–132. Springer, 2008.

[5] S. Bickel, J. Bogojeska, T. Lengauer, and Tobias S. Multi-task learning for hiv therapy screening. In *International conference on Machine learning*, pages 56–63. ACM, 2008.



(a) Landmine (b) Cancer (c) Parkinson (d) School

Figure 5: Basis-specific task-grouping discovered by OSCAR-MTL for various datasets used in this paper. For easy viewing of the different task groups, we have re-ordered the tasks along each latent basis except for the Cancer dataset .

[6] H. D Bondell and B. J Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64(1):115–123, 2008.

[7] S Boyd, N Parikh, E Chu, B Peleato, and J Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

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

[9] J. Chen, J. Liu, and J. Ye. Learning incoherent sparse and low-rank patterns from multiple tasks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):22, 2012.

[10] J. Chen, J. Zhou, and J. Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 42–50. ACM, 2011.

[11] H. Daumé III. Bayesian multitask learning with latent hierarchies. In *UAI*, pages 135–142, 2009.

[12] C. Ding, T. Li, and M. Jordan. Convex and semi-nonnegative matrix factorizations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):45–55, 2010.

[13] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM, 2004.

[14] P. Gong, J. Ye, and C. Zhang. Robust multi-task feature learning. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 895–903, 2012.

[15] S Gupta, D. Phung, and S. Venkatesh. Factorial multi-task learning: A Bayesian nonparametric approach. In *International Conference on Machine Learning*, 2013.

[16] S Gupta, T Tran, W Luo, D Phung, R L Kennedy, A Broad, D Campbell, D Kipp, M Singh, M Khasraw, and S Venkatesh. Machine-learning prediction of cancer survival: a retrospective study using electronic administrative records and a cancer registry. *BMJ open*, 4(3), 2014.

[17] J. Huang and C. X Ling. Using auc and accuracy in evaluating learning algorithms. *TKDE*, 17(3):299–310, 2005.

[18] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *International Conference on Machine Learning*, pages 521–528, 2011.

[19] A. Kumar and H. Daumé III. Learning task grouping and overlap in multi-task learning. In *International Conference on Machine Learning (ICML)*, 2012.

[20] D. D Lee and H S. Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[21] S. Lee, V. Chatalbashev, D. Vickrey, and D. Koller. Learning a meta-level prior for feature relevance from multiple related tasks. In *International conference on Machine learning*, pages 489–496. ACM, 2007.

[22] P. J Lenk, W. S DeSarbo, P. E Green, and M. R Young. Hierarchical bayes conjoint analysis: recovery of partworth heterogeneity from reduced experimental designs. *Marketing Science*, 15(2):173–191, 1996.

[23] P. Rai and H. Daume. Infinite predictor subspace models for multitask learning. In *International Conference on Artificial Intelligence and Statistics*, pages 613–620, 2010.

[24] A. Tsanas, M. A Little, P. E McSharry, and L. O Ramig. Enhanced classical dysphonia measures and sparse regression for telemonitoring of parkinson's disease progression. In *ICASSP*, pages 594–597. IEEE, 2010.

[25] X. Wang, C. Zhang, and Z. Zhang. Boosted multi-task learning for face verification with applications to web image and video search. In *IEEE Conference on Computer Vision and Pattern Recognition.*, pages 142–149. IEEE, 2009.

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

[27] S. Yang, L. Yuan, Y-C. Lai, X. Shen, P. Wonka, and J. Ye. Feature grouping and selection over an undirected graph. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 922–930. ACM, 2012.

[28] K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In *International conference on Machine learning*, pages 1012–1019. ACM, 2005.

[29] Y. Zhang and D-Y. Yeung. A convex formulation for learning task relationships in multi-task learning. In *Uncertainty in Artificial Intelligence*, pages 733–442, 2010.

[30] L. W. Zhong and J. T Kwok. Efficient sparse modeling with automatic feature grouping. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(9):1436–1447, 2012.

[31] J. Zhou, J. Liu, V. A Narayan, and J. Ye. Modeling disease progression via multi-task learning. *NeuroImage*, 78:233–248, 2013.