# Generalized Dictionary for Multitask Learning with Boosting

**Boyu Wang** and **Joelle Pineau**
School of Computer Science
McGill University, Montreal, Canada
boyu.wang@mail.mcgill.ca, jpineau@cs.mcgill.ca

## Abstract

While multitask learning has been extensively studied, most existing methods rely on linear models (e.g. linear regression, logistic regression), which may fail in dealing with more general (nonlinear) problems. In this paper, we present a new approach that combines *dictionary learning* with *gradient boosting* to achieve multitask learning with general (nonlinear) basis functions. Specifically, for each task we learn a sparse representation in a nonlinear dictionary that is shared across the set of tasks. Each atom of the dictionary is a nonlinear feature mapping of the original input space, learned in function space by gradient boosting. The resulting model is a hierarchical ensemble where the top layer of the hierarchy is the task-specific sparse coefficients and the bottom layer is the boosted models common to all tasks. The proposed method takes the advantages of both dictionary learning and boosting for multitask learning: knowledge across tasks can be shared via the dictionary, and flexibility and generalization performance are guaranteed by boosting. More important, this general framework can be used to adapt any learning algorithm to (nonlinear) multitask learning. Experimental results on both synthetic and benchmark real-world datasets confirm the effectiveness of the proposed approach for multitask learning.

## 1 Introduction

Multitask learning [Caruana, 1997] is a learning paradigm that aims to improve learning performance across many tasks by leveraging information and knowledge that is shared across tasks. It has been demonstrated both theoretically [Ben-David and Schuller, 2003; Ando and Zhang, 2005; Maurer *et al.*, 2013] and empirically [Argyriou *et al.*, 2007; Liu *et al.*, 2009; Kumar and Daumé III, 2012; Hernández-Lobato *et al.*, 2015] that generalization performance can be improved by learning multiple tasks jointly, in contrast to learning each task individually, especially when training samples for each task are limited and the number of tasks is large.

One key assumption of multitask learning is that the tasks are related to each other and therefore there is some underlying relatedness structure that can be exploited and shared across tasks. Examples of such structure include the model parameters lying close to each other [Evgeniou and Pontil, 2004] in a low dimensional subspace [Ando and Zhang, 2005], manifold [Agarwal *et al.*, 2010], or sharing similar sparsity patterns [Liu *et al.*, 2009; Obozinski *et al.*, 2010].

One drawback of most multitask approaches is that they assume that *all* the tasks are related to each other. This is restrictive in real world applications where the tasks may share knowledge in a more complicated way. To address this issue, algorithms have been proposed to model more sophisticated task relatedness structures. For example, some methods assume that the tasks can be clustered into groups, and that tasks within each group are similar to each other [Xue *et al.*, 2007; Jacob *et al.*, 2008; Kang *et al.*, 2011]. Other models consider the existence of outlier tasks [Chen *et al.*, 2011; Gong *et al.*, 2012] or hierarchical structure of model parameters [Zweig and Weinshall, 2013]. Task relatedness has also been modeled by correlations [Zhang and Yeung, 2010; Zhang and Schneider, 2010] or tree structures [Kim and Xing, 2010]. Finally, the dictionary learning approach [Kumar and Daumé III, 2012; Maurer *et al.*, 2013] offers another method for multitask learning and can model various relatedness structures such as disjoint grouping, partial overlap, and outlier tasks. Its generalization performance has been analyzed in [Maurer *et al.*, 2013; 2014].

However, most existing methods are limited to learning a linear model of tasks, which restricts their potential for addressing more complex nonlinear problems. Although some kernel methods have been proposed for this issue [Yu *et al.*, 2005; Evgeniou *et al.*, 2005], they usually require well-defined kernel functions which can be difficult to specify. In addition, the computational complexity of kernel algorithms grows cubically with the number of training samples, which limits their applications on large datasets. There have also been boosting-based multitask learning algorithms proposed in [Chapelle *et al.*, 2010; Becker *et al.*, 2013], but both of these approaches implicitly assume that all the tasks are related to each other, and fail to capture more sophisticated task relatedness such as grouped and/or outlier tasks.

In this paper, we propose a generalized dictionary learning algorithm for multitask learning. The starting point of our method is similar to the dictionary multitask learning (DMTL) approach [Kumar and Daumé III, 2012], assuming

that the model parameters of the tasks lie in a low dimensional subspace spanned by a linear dictionary. We extend this by constructing a nonlinear mapping defined by a generalized dictionary, which allows us to handle datasets that are difficult to model by linear algorithms. More specifically, instead of learning a dictionary of basis vectors as in DMTL, we learn a more generalized dictionary that contains a set of basis functions in function space. We optimize the set of basis functions using *gradient boosting*, and call the approach *generalized dictionary multitask learning with boosting* (GDMTLB). There are several advantages to the GDMTLB: 1. Compared with DMTL, GDMTLB produces more expressive nonlinear models to tackle complex problems arising from real world applications. 2. As a meta-learning algorithm, GDMTLB offers out-of-the-box usability and allows arbitrary learning algorithm to be used for multitask learning. 3. Compared with other nonlinear multitask learning approaches (e.g., [Chapelle *et al.*, 2010]), GDMTLB can capture sophisticated task relatedness structures by using dictionary learning and sparse coding. 4. It offers theoretical guarantee of generalization bound, which gives the insight into the nature of the algorithm.

## 2 Method

We being this section by formulating the problem and describing the generalized dictionary learning framework for multitask learning. We then derive learning algorithms for specific loss functions and problems, based on the idea of *functional gradient descent* [Friedman, 2001; Mason *et al.*, 2000], leading to our *boosted dictionary learning* algorithm.

### 2.1 Problem Formulation

Let $\{\mathcal{S}_1, \ldots, \mathcal{S}_T\}$ be $T$ related tasks, where $\mathcal{S}_t = \{(x_1^t, y_1^t), \ldots (x_{N_t}^t, y_{N_t}^t)\}$ are the $d$-dimensional training samples for the $t$-th task. In the DMTL approach [Kumar and Daumé III, 2012; Maurer *et al.*, 2013], the objective is to learn a linear model parameter $w_t \in \mathbb{R}^d$ for each task, which is sparse coded by $w_t = D\gamma_t$, where $D \in \mathbb{R}^{d \times M}$ is the dictionary shared across the tasks, $\gamma_t \in \mathbb{R}^M$ is the sparse coefficient vector for the $t$-th task, $M$ is the size of dictionary. Formally, the goal is to minimize the following objective function:

$$\min_{D,\{\gamma_t\}} \mathcal{L}(D, \{\gamma_t\}) \tag{1}$$
$$= \min_{D,\{\gamma_t\}} \sum_{t=1}^{T} \sum_{i=1}^{N_t} \ell(\langle D\gamma_t, x_i^t \rangle, y_i^t) + \mu \sum_{t=1}^{T} ||\gamma_t||_1 + \lambda \mathcal{R}(D),$$

where $\langle \cdot, \cdot \rangle$ is an inner product, $\ell(\cdot, \cdot)$ is a loss function, $|| \cdot ||_1$ is the $\ell_1$ norm used to encourage sparsity of the coefficients $\{\gamma_t\}$, $\mathcal{R}(\cdot)$ is the regularization term imposed on dictionary $D$ to avoid overfitting, and $\mu$ and $\lambda$ are the regularization parameters. It has been proven that given $M \ll d$ and $M < T$, the DMTL algorithm can have a lower generalization error bound than learning $T$ tasks separately [Maurer *et al.*, 2013].

The main drawback of this approach (as well as others in the literature) is that it only considers linear hypotheses, which cannot properly deal with nonlinear problems. The principal contribution in our paper is to propose a more flexible learning framework that can accommodate any existing algorithm to model nonlinearity in multitask learning scenarios. Specifically, instead of learning a $d \times M$ matrix $D$, we consider a generalized dictionary $F(\cdot) = [f_1(\cdot), \ldots, f_M(\cdot)]$, where $f_m(\cdot), m = 1, \ldots, M$ can be any hypothesis:

$$\min_{F,\{\gamma_t\}} \mathcal{L}(F, \{\gamma_t\}) \tag{2}$$
$$= \min_{F,\{\gamma_t\}} \sum_{t=1}^{T} \sum_{i=1}^{N_t} \ell\left(\langle F(x_i^t), \gamma_t \rangle, y_i^t\right) + \mu \sum_{t=1}^{T} ||\gamma_t||_1.$$

Note that we have omitted the regularization term $\mathcal{R}(F)$, since we later will use the trick of gradient approximation to avoid overfitting, as detailed in [Friedman, 2001].

The dictionary $D$ in Eq. 1 can be regarded as a linear mapping from $x \in \mathbb{R}^d$ to $z = D^\top x \in \mathbb{R}^M$. The DMTL algorithm can be retrieved as a special case of GDMTLB by setting $F(x) = D^\top x$. In this paper, we focus on the more general case where the atoms of dictionary $F$ are the set of nonlinear mappings.

Eq. 2 can be optimized by the alternating optimization approach [Bezdek and Hathaway, 2003], as detailed in Algorithm 1. More precisely, we alternate between the following two optimization steps:

**Sparse Coding (Line 4 of Algorithm 1)**
Given a fixed hypothesis set, Eq. 2 can be decomposed into $T$ individual $\ell_1$-regularized optimization problems:

$$\gamma_t = \arg\min_{\gamma_t} \sum_{i=1}^{N_t} \ell\left(\langle F(x_i^t), \gamma_t \rangle, y_i^t\right) + \mu ||\gamma_t||_1, \tag{3}$$

for $t = 1, \ldots, T$, which can be solved efficiently by many algorithms (e.g., two-metric projection, coordinate descent, accelerated gradient method).

**Generalized Dictionary Learning (Line 6 of Algorithm 1)**
The second objective is to learn a dictionary over any hypothesis class, rather than a matrix of linear mapping or some specific model, which motivates us to perform gradient descent of $F$ in function space. In particular, we treat $F$ as a set of parameters, and solve Eq. 2 as a sum of component dictionaries:

$$F = \sum_{k=1}^{K} \rho_k H_k,$$

where $K$ is the number of weak learners/dictionaries. We select $H_k$ such that the Frobenius distance between $H_k$ and the negative gradient of $\mathcal{L}$ at $F = F_{k-1}$ is minimized:

$$H_k = \arg\min_{H} \left|\left| -\left[\frac{\partial \mathcal{L}(F, \{\gamma_t\})}{\partial F}\right]_{F=F_{k-1}} - H \right|\right|_{\mathsf{F}}, \tag{4}$$

and $\rho_k$ is the step size chosen by line search:

$$\rho_k = \arg\min_{\rho} \mathcal{L}\left(F_{k-1} + \rho H_k, \{\gamma_t\}\right). \tag{5}$$

Let $\alpha_t(x) \triangleq \langle F(x), \gamma_t \rangle$ and using chain rule, the gradient of the loss function $\ell$ with respect to $F(x)$ is given by

$$\frac{\partial \ell(\langle F(x), \gamma_t \rangle, y)}{\partial F(x)} = \frac{\partial \ell(\alpha_t(x), y)}{\partial \alpha_t} \cdot \gamma_t. \tag{6}$$

**Algorithm 1** Generalized Dictionary for Multitask Learning

**Input:** $\{\mathcal{S}_1, \ldots, \mathcal{S}_T\}$, $maxIter$, the number of iterations $K$, the number of basis hypotheses $M$, regularization parameter $\mu$

Initialize $F$
**while** $n < maxIter$ **do**
  **for** $t = 1, \ldots, T$ **do**
    Solve Eq. 3.
  **end for**
  Learn a generalized dictionary given $\{\gamma_t\}$. (Detailed in Algorithm 2 and Algorithm 3.)
  $n = n + 1$
  **if** converge **then**
    break
  **end if**
**end while**

**Output:** Generalized dictionary $F$, sparse coefficients $\{\gamma_t\}$.

By choosing different loss functions $\ell$ we can obtain different learning algorithms, suitable for different types of problems. More important, by using a gradient boosting approach, the basis functions $\{f_m\}$ of $F$ are decoupled and therefore can be learned individually and efficiently, as detailed below.

## 2.2 Exponential Loss for Classification

We first consider an AdaBoost-type [Freund and Schapire, 1997] algorithm, which minimizes the **exponential loss** $\ell(\langle F(x), \gamma\rangle, y) = \exp(-y\langle F(x), \gamma\rangle)$, where $y \in \{-1, +1\}$. Given fixed $\{\gamma_t\}$, the gradient for exponential loss over $\{x_i^t, y_i^t\}$ at $F = F_{k-1}$ is given by

$$\left[\frac{\partial \ell(F(x_i^t), \gamma_t)}{\partial F(x_i^t)}\right]_{F=F_{k-1}} = -y_i^t \gamma_t \exp\left(-y_i^t \langle F(x_i^t), \gamma_t\rangle\right). \tag{7}$$

Plugging Eq. 7 into Eq. 4 gives

$$h_{k,m} = \arg\min_h \sum_{t=1}^{T}\sum_{i=1}^{N_t} \left(h(x_i^t) - y_i^t w_i^t \gamma_{t,m}\right)^2 \tag{8}$$

for $m = 1, \ldots, M$, where $w_i^t \triangleq \exp\left(-y_i^t \langle F_{k-1}(x_i^t), \gamma_t\rangle\right)$, $h_{k,m}$ is the $m$-th basis function of $H_k$, $\gamma_{t,m}$ is the $m$-th entry of $\gamma_t$. As we focus on classification problems, we have $h(x) \in \{1, +1\}$. Therefore, Eq. 8 is equivalent to

$$h_{k,m} = \arg\min_h \sum_{t=1}^{T}\sum_{i=1}^{N_t} \gamma_{t,m} w_i^t \mathbb{1}(y_i^t \neq h(x_i^t)). \tag{9}$$

Eq. 9 reveals that the solution of Eq. 4 can be decomposed into $M$ individual learning problems, and for each we learn a hypothesis that minimizes the weighted error rate in predicting the label $y$. In addition, at the $k$-th iteration, for the $m$-th basis function, the weight for a sample $\{x_i^t, y_i^t\}$ is determined by $v_{i,k,m}^t \triangleq \gamma_{t,m} w_i^t$. As $\{\gamma_t\}$ are sparse vectors, each basis hypothesis is only trained on the tasks with non-zero coefficients $\{\gamma_{t,m}\}$. This is reasonable, since $\gamma_{t,m} = 0$ means that the $m$-th basis function is not involved in predicting the $t$-th task, and therefore samples from the $t$-th task should not contribute to the training of the $m$-th base learner. This is not only computationally efficient but also introduces grouping and/or partial overlap effects that enable the algorithm to selectively share information across tasks, as in [Kumar and Daumé III, 2012].

**Algorithm 2** AdaBoosted Dictionary Learning

**Input:** $\{\mathcal{S}_1, \ldots, \mathcal{S}_T\}$, $\{\gamma_t\}$, the number of iterations $K$, number of basis hypothesises $M$,

Initialize $w_i^t = \frac{1}{N}$ for $t \in \{1, \ldots, T\}$, $i \in \{1, \ldots, N_t\}$, where $N = \sum_{t=1}^{T} N_t$.
**for** $k = 1, \ldots, K$ **do**
  **for** $m = 1, \ldots, M$ **do**
    **for** $t = 1, \ldots, T$ **do**
      $v_{i,k,m}^t = \gamma_{t,m} w_i^t$ for $i \in \{1 \ldots N_t\}$
    **end for**
    Normalize $v_{i,k,m}^t$
    $h_{k,m} = \arg\min_h \sum_{t=1}^{T}\sum_{i=1}^{N_t} v_{k,m}^t \mathbb{1}(y_i^t \neq h(x_i^t))$
  **end for**
  Compute error: $\epsilon_k = \sum_{t=1}^{T}\sum_{y_i^t \neq \text{sign}\langle H_k(x_i^t), \gamma_t\rangle} w_i^t$
  Compute $\rho_k = \frac{1}{2}\ln\frac{1-\epsilon_k}{\epsilon_k}$
  Set $w_i^t \leftarrow w_i^t \cdot \exp(\beta_k \mathbb{1}(y_i^t \neq \text{sign}\langle H_k(x_i^t), \gamma_t\rangle))$, followed by a normalization step.
**end for**

**Output:** $F = [f_1, \ldots, f_M]$, where $f_m(x) = \sum_{k=1}^{K} \rho_k h_{k,m}(x)$

To obtain the step size $\rho_k$, we differentiate Eq. 5 with respect to $\rho_k$ and set it equal to zero. Using some simple calculation, we determine $\rho_k$ analytically:

$$\rho_k = \frac{1}{2}\ln\frac{1-\epsilon_k}{\epsilon_k},$$

where $\epsilon_k = \sum_{t=1}^{T}\sum_{y_i^t \neq \text{sign}\langle H_k, \gamma_t\rangle} w_i^t$. The pseudo-code for dictionary learning for classification is shown in Algorithm 2.

## 2.3 Squared Loss for Regression

Alternately, we consider a regression problem, applying the proposed framework with **squared loss** $\ell(\langle F(x), \gamma\rangle, y) = \frac{1}{2}(\langle F(x), \gamma\rangle - y)^2$, where $y \in \mathbb{R}$. This yields an L2boosting-type [Bühlmann and Yu, 2003] dictionary learning algorithm.

Given training sample $\{x_i^t, y_i^t\}$, the loss function with respect to the $m$-th basis function $f_m$ can be reformulated as

$$\ell(\langle F(x_i^t), \gamma_t\rangle, y_i^t) = \gamma_{t,m}^2 \ell(f_m(x_i^t), z_i^t), \tag{10}$$

where $z_i^t = \frac{y_i^t - \langle F(x_i^t), \gamma_t\rangle}{\gamma_{t,m}} + f_m(x_i^t)$. Therefore, the original least square fitting problem can be reformulated as a weighted least square fitting problem for $f_m$, where the weight is given by $\gamma_{t,m}^2$. Differentiating Eq. 10 with respect to $f_m(x_i^t)$ gives

$$\frac{\partial \ell(F(x_i^t), \gamma_t)}{\partial f_m(x_i^t)} = \gamma_{t,m}^2 \frac{\partial \ell(f_m(x_i^t), z_i^t)}{\partial f_m(x_i^t)} = \gamma_{t,m}^2 (f_m(x_i^t) - z_i^t). \tag{11}$$

Plugging Eq. 11 into Eq. 4 gives

$$h_{k,m} = \arg\min_h \sum_{t=1}^{T}\sum_{i=1}^{N_t} \gamma_{t,m}^2 (h(x_i^t) - r_{i,m}^t)^2, \tag{12}$$

**Algorithm 3** L2Boosted Dictionary Learning

**Input:** $\{\mathcal{S}_1, \ldots, \mathcal{S}_T\}$, $\{\gamma_t\}$, the number of iterations $K$, number of basis hypothesises $M$,

1: Initialize residual: $r_{i,m}^t = \frac{y_i^t}{\gamma_{t,m}}$
2: **for** $k = 1, \ldots, K$ **do**
3: **for** $m = 1, \ldots, M$ **do**
4: $h_{k,m} = \arg\min_h \sum_{t=1}^{T} \sum_{i=1}^{N_t} \gamma_{t,m}^2 (h(x_i^t) - r_i^t)^2$
5: **end for**
6: Compute $\rho_k$ using Eq. 14.
7: Update residual: $r_i^t \leftarrow r_i^t - \frac{\langle \text{diag}(\rho_k) H_k(x_i^t), \gamma_t \rangle}{\gamma_{t,m}}$
8: **end for**

**Output:** $F = [f_1, \ldots, f_M]$, where $f_m(x) = \sum_{k=1}^{K} \rho_{k,m} h_{k,m}(x)$

where

$$r_{i,m}^t = z_i^t - f_m(x_i^t)|_{F=F_{k-1}} = \frac{y_i^t - \langle F_{k-1}(x_i^t), \gamma_t \rangle}{\gamma_{t,m}},$$

for $m = 1, \ldots, M$. Again, each basis function of $F$ can be learned separately by repeated weighted least square fitting of current residuals, where the weights of samples of the $t$-th task for the $m$-th basis function are given by $\gamma_{t,m}^2$. For L2Boosting, the step size of $H_k$ is not strictly necessary [Bühlmann and Hothorn, 2007], but it can be beneficial if we assign different step sizes to each basis function $h_{k,m}$ (i.e., $\rho_k$ is a vector of step sizes). Differentiating $\mathcal{L}$ with respect to $\rho_k$ and setting equal to zero, we have

$$\begin{aligned} &\sum_{t=1}^{T} \sum_{i=1}^{N_t} \text{diag}(\gamma_t) H_k(x_i^t) H_k^\top(x_i^t) \text{diag}(\gamma_t) \rho_k \qquad (13) \\ &\quad = \sum_{t=1}^{T} \sum_{i=1}^{N_t} \text{diag}(\gamma_t) H_k(x_i^t) \left( y_i^t - \langle F_{k-1}(x_i^t), \gamma_t \rangle \right), \end{aligned}$$

which gives

$$\begin{aligned} \rho_k = &\left( \sum_{t=1}^{T} \sum_{i=1}^{N_t} \text{diag}(\gamma_t) H_k(x_i^t) H_k^\top(x_i^t) \text{diag}(\gamma_t) \right)^{-1} \cdot \\ &\sum_{t=1}^{T} \sum_{i=1}^{N_t} \text{diag}(\gamma_t) H_k(x_i^t) \left( y_i^t - \langle F_{k-1}(x_i^t), \gamma_t \rangle \right), \qquad (14) \end{aligned}$$

where $\text{diag}(\gamma_t)$ is a diagonal matrix with the elements of vector $\gamma_t$ on the main diagonal. The boosted dictionary learning algorithm with squared loss is summarized in Algorithm 3.

## 2.4 Dictionary Initialization

The dictionary $F$ can be initialized in several ways. For example, we can first learn a linear dictionary by DMTL and use it as a warm start, or randomly select $T'$ tasks to train each basis function. In this work, we consider both approaches and the better empirical results between the two are reported in the experimental section.

## 2.5 Computational Complexity

The computational complexity of GDMTLB depends on the choice of base learner, as well as the optimization algorithms used for sparse coding. We assume that the complexity of training a baser learner is $\mathcal{O}(\xi(N_{tr}^m, d))$, where $N_{tr}^m$ is the number of training samples for the $m$-th atom of dictionary. In general $N_{tr}^m \leq N$, since the coefficients $\{\gamma_t\}$ are sparse. Then, the overall complexity of each dictionary learning step will be $\mathcal{O}(KM\xi(N, d))$. Note that we have omitted the complexity of testing and weight update step of boosting since it is usually much smaller than that of training cost. The sparse coding step requires solving a $\ell_1$ regularized minimization problem (i.e., Eq. 3). If we use accelerated gradient descent [Nesterov, 2004], for each sparse coefficient $\gamma_t$, it takes $\mathcal{O}(dN_t)$ to evaluate the function value and its gradient, and $\mathcal{O}(d)$ to project the point back onto $\ell_1$ ball. As the convergence rate of this method is quadratic, the computational complexity of sparse coding step is $\mathcal{O}(\frac{1}{\sqrt{\varepsilon}} dN)$, where $\varepsilon$ is the error tolerance. Therefore, the overall complexity of each alternating optimization iteration is $\mathcal{O}(KM\xi(N, d) + \frac{1}{\sqrt{\varepsilon}} dN)$, which scales linearly with $K$ and $M$. Empirically, the entire GDMTLB algorithm usually stops within ten iterations.

## 2.6 Theoretical Analysis

The following theorem provides a generalization error bound for GDMTLB with exponential loss (Algorithm 2) and using linear functions as base learners.[1]

**Theorem 1.** *Let $G = (G^1, \ldots, G^T) : \mathbb{R}^d \to \mathbb{R}^T$, with $G^t(x) = \langle F(x), \gamma_t \rangle$, be the multitask classifier returned by GDMTLB with exponential loss, and $\mathcal{G}$ be the function class of $G$. Let $\{\mathcal{S}_1, \ldots, \mathcal{S}_T\}$ be $T$ related tasks, where $\mathcal{S}_t = \{(x_1^t, y_1^t), \ldots (x_{N_t}^t, y_{N_t}^t)\}$ are the $d$-dimensional training samples for the $t$-th task. For simplicity we assume that $N_t = N, \forall t \in \{1, \ldots, T\}$. Given $G$ and a sample $\{x^t, y^t\}$ of the $t$-th task, define loss function $\ell : \mathbb{R}^T \times \mathbb{R} \to \{0, 1\}$ as $\ell(G(x^t), y^t) = \mathbb{1}_{y^t G^t(x) \leq 0}$, where $\mathbb{1}_\omega$ is the indicator function of event $\omega$. If the base learners of GDMTLB are linear functions, and $\epsilon_k < \frac{1}{2}, \forall k \in \{1, \ldots, K\}$, then for any $\delta > 0$ and fixed $\tau > 0$, with probability at least $1 - \delta$, for all $G \in \mathcal{G}$, its generalization error $\mathbb{E}[\ell(G)]$ is bounded by*

$$\mathbb{E}[\ell(G)] \leq \overbrace{2^K \prod_{k=1}^{K} \sqrt{\epsilon_k^{1-\tau}(1-\epsilon_k)^{1+\tau}}}^{\mathcal{A}} + 3\sqrt{\frac{\ln(2/\delta)}{2NT}}$$

$$+ \overbrace{\frac{2}{\tau NT} \sqrt{M \sum_{t=1}^{T} \sum_{i=1}^{N} ||x_i^t||_2^2}}^{\mathcal{B}} + \overbrace{\frac{8}{\tau} \sqrt{\frac{\ln(2M) \sum_{t=1}^{T} \lambda_{\max}\left(\hat{\Sigma}(X_t)\right)}{NT}}}^{\mathcal{C}},$$

*where* $\lambda_{\max}\left(\hat{\Sigma}(X_t)\right) = \sup_{||d|| \leq 1} \sum_{i=1}^{N} \langle d, x_i^t \rangle$.

We have several remarks concerning Theorem 1.

1. From the learning bound, it can be observed that GDMTLB inherits the benefits from both AdaBoost and linear dictionary for multitask learning. $\mathcal{A}$ is the upper bound of the margin loss for AdaBoost [Mohri *et al.*, 2012], while $\mathcal{B}$ and $\mathcal{C}$ are the upper bound of

[1] The detailed proof can be found in our online supplementary materials https://sites.google.com/site/borriewang/.

the Rademacher complexity of a linear dictionary-based multitask learning algorithm [Maurer *et al.*, 2014].

2. If the margin loss is small for a relative large $\tau$, small generalization error is guaranteed. In addition, it can be proved that, under certain conditions, the upper bound $\mathcal{A}$ is reduced exponentially as a function of number of iterations $K$ [Mohri *et al.*, 2012], which justifies the advantage of our boosting approach for multitask learning. Finally, given a fixed function class, GDMTLB will succeed if we can design an effective algorithm that performs with error across all tasks through all iterations (i.e., $\epsilon_k$ are small), since it leads to low margin loss.
3. $\mathcal{B}$ and $\mathcal{C}$ indicate the benefits of performing multitask learning using a dictionary learning approach. Note that $\mathcal{C}$ can be dominated by $\mathcal{B}$ for $N \ll d$, and compared with the individual task learning approach, $\mathcal{B}$ is lower by a factor of $\sqrt{M/T}$, which demonstrates the advantage of multitask dictionary learning in high dimensional spaces by choosing $M < T$ [Maurer *et al.*, 2014].

## 3 Experiments

We now evaluate GDMTLB algorithm against several state-of-the-art algorithms on both synthetic and real-world datasets. Competitive methods include $\ell_{2,1}$-regularized multitask feature learning (MTFL) [Liu *et al.*, 2009], trace-norm regularized multitask learning (Trace) [Argyriou *et al.*, 2007], dictionary multitask learning (DMTL) [Kumar and Daumé III, 2012], as well as a nonlinear boosted multitask learning algorithm (MultiBoost) [Chapelle *et al.*, 2010]. In addition, single task learning (STL) is also used as the baseline algorithm, where the tasks are learned individually.

In all experiments, the hyper-parameters (e.g., $M, \mu$, different dictionary initializations) are selected by cross-validation. Regression tree is used as the weak learner of GDMTLB for regression, and logistic regression is used as the weak learner for classification. Each dataset is evaluated by using 10 randomly generated 50/50 splits of the data between training and test set, and the average results are reported.

### 3.1 Synthetic Data

The synthetic dataset consists of 2-dimensional vectors, two groups of tasks, and 20 tasks per group. For the $j$-th task of the $i$-th group, the samples are generated by $y_j^i \sim c_j \cdot (x_j^{i\top} w_i + x_j^{i\top} P_i x_j^i) + \epsilon$, where $x \sim \mathcal{N}(\mathbf{0}, I)$, $c_j \sim \mathcal{U}(0, 2)$, $w_i \sim \mathcal{N}(\mathbf{0}, 3I)$, $\epsilon \sim \mathcal{N}(0, 1)$, $P_i = Q_i^\top Q_i$ (each entry of $Q_i$ is sampled from a normal distribution), where $\mathcal{N}$ denotes the Gaussian distribution, $\mathcal{U}$ denotes the uniform distribution. Therefore, the parameters of the tasks within each group are identical up to a scaling factor. For each task, there are 30 training samples and 30 test samples.

Figures 2(a)-2(c) show the samples in the original feature space, where we observe that the data cannot be properly fitted by linear regression due to the nonlinearity of the data. Figures 2(d)-2(f) demonstrate the samples projected into a new feature space by nonlinear dictionary $F$, from which it can be observed that the samples of the first group (blue samples) exhibit linearity in the first dimension (Figure 2(e)) while the samples of the second group (red samples) exhibit linearity in the second dimension (Figure 2(f)), which means the data can be well fitted by sparse linear regression in the new feature space. In other words, the nonlinear structure of tasks can be well captured by the dictionary $F$, where each basis function of $F$ corresponds to one group of tasks. Each task within the group can be fitted by the corresponding basis function up to a scaling factor, which is the slope of linear fitting in the new feature space. This can be further illustrated by Figure 1, where it can be observed that after projection the outputs of each group of tasks are highly correlated with only one dimension (basis function) of the new feature space. The results of different algorithms, measured by root mean squared error (RMSE), are shown in the first column of Table 1, where we see that GDMTLB outperforms other multitask learning algorithms in this simple case. This is not surprising, since the linear multitask learning algorithms cannot fit nonlinear functions, while MultiBoost cannot capture the group structure of the tasks.

Table 1: Learning performances (mean $\pm$ std dev.), RMSE for synthetic and school datasets, AUC for landmine dataset. The best results for each dataset are bolded.

| | Synthetic | School | Landmine |
|---|---|---|---|
| STL | $5.05 \pm 0.24$ | $10.91 \pm 0.08$ | $0.7767 \pm 0.009$ |
| MTFL | $4.97 \pm 0.21$ | $10.68 \pm 0.06$ | $0.7805 \pm 0.011$ |
| Trace | $5.01 \pm 0.35$ | $10.65 \pm 0.06$ | $0.7847 \pm 0.008$ |
| DMTL | $4.92 \pm 0.19$ | $10.44 \pm 0.07$ | $0.7809 \pm 0.010$ |
| MultiBoost | $4.42 \pm 0.28$ | $10.59 \pm 0.08$ | $0.7789 \pm 0.013$ |
| GDMTLB | $\mathbf{3.31 \pm 0.42}$ | $\mathbf{10.11 \pm 0.07}$ | $\mathbf{0.7936 \pm 0.008}$ |

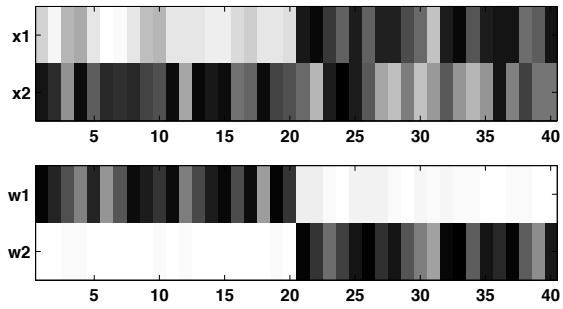


Figure 1: Correlation coefficients between features and outputs. *Top*: Original feature space, *Bottom*: Projected feature space.

### 3.2 Real Data

Next, we evaluate multi-task methods on three real-world datasets, one for regression: *London school* data [Argyriou *et al.*, 2007]; and two for classification: *landmine* data [Xue *et al.*, 2007], and *BCI Competition* data[2]. We omit the description of the first two datasets as they are frequently used benchmarks for multitask learning. The BCI dataset consists of EEG signals from 9 subjects who are instructed with visual

[2]http://www.bbci.de/competition/iv/.

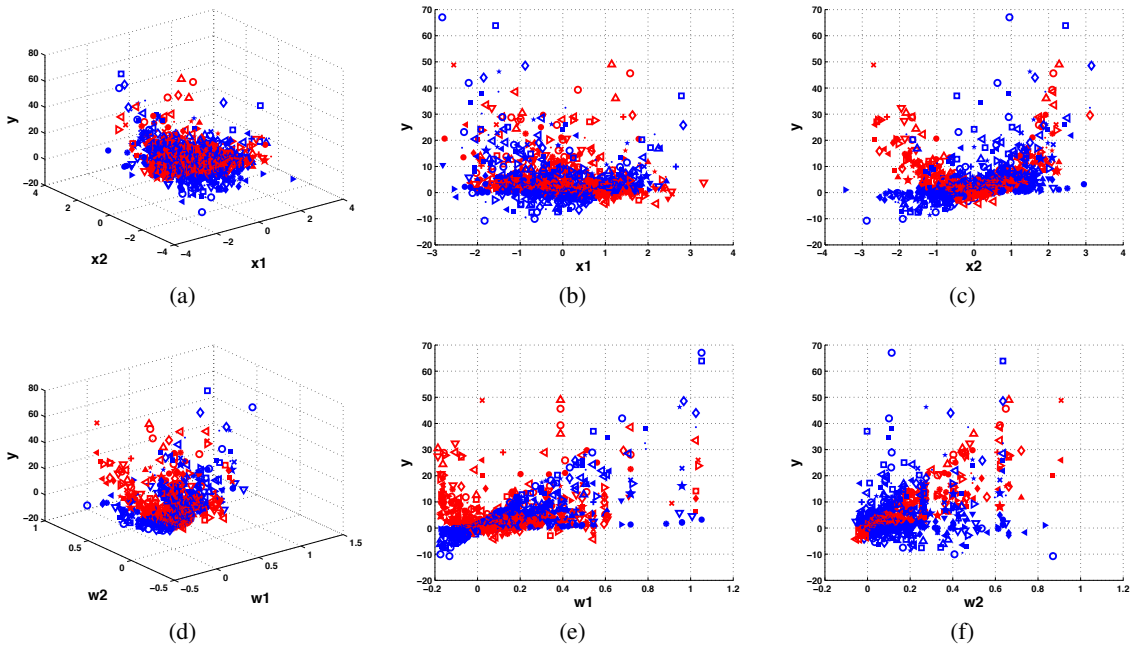


Figure 2: A synthetic example with two groups of tasks marked in different colors. Samples of different tasks within each group are marked in different symbols. *Top*: the original samples, *Bottom*: the samples projected by nonlinear dictionary.

Table 2: Classification accuracy (%) of different algorithms for nine different subjects. The best results are bolded.

| | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | **Mean** |
|---|---|---|---|---|---|---|---|---|---|---|
| STL | 86.81 | 51.39 | 90.28 | 64.58 | 51.39 | 61.11 | 81.25 | **92.36** | 87.50 | 74.07 |
| MTFL | 82.64 | 52.78 | 92.36 | 65.97 | 50.69 | 61.11 | **82.64** | **92.36** | 88.89 | 74.38 |
| Trace | 84.03 | 50.69 | 93.75 | **68.06** | **54.86** | 61.11 | 81.94 | 90.97 | 87.50 | 74.77 |
| DMTL | 84.03 | 54.86 | 91.67 | 65.97 | 52.08 | 63.19 | 80.56 | **92.36** | **89.58** | 74.92 |
| MultiBoost | 85.42 | 53.47 | 91.67 | 65.28 | 53.47 | 61.81 | 79.86 | 90.97 | **89.58** | 74.61 |
| GDMTLB | **90.97** | **55.56** | **95.83** | 66.67 | 52.78 | **65.28** | 81.25 | 90.28 | 88.19 | **76.31** |

cues to perform left hand or right hand motor imagery. Each subject corresponds to a distinct task. For each subject, the EEG signals consist of a training set and a test set, each containing 72 trials. The main challenge of this problem is that the underlying task (i.e. patient) relatedness is unknown and the EEG data structure can be complex [Müller *et al.*, 2003].

For the London school regression problem, RMSE is used for performance evaluation. Performance on the classification problems is measured using area under ROC curve (AUC) for the landmine data since the dataset is imbalanced, and classification accuracy for the EEG dataset. The results on the London school and landmine datasets are summarized in the second and third columns of Table 1, which again shows that GDMTLB improves the predictive performances over single task learning as well as other multitask learning algorithms. Table 2 presents the results on the EEG dataset. GDMTLB achieves the highest classification accuracy on four subjects, yielding an average improvement of 2.24% over all subjects, which is significant compared with other multitask learning approaches. Across all the experiments, the improvements of GDMTLB over STL is at least twice as much as other algorithms, which validates the effectiveness of our algorithm.

## 4 Conclusion

This paper presents a novel GDMTLB algorithm for multitask learning with nonlinear structure. The core idea is to apply gradient boosting to learn the dictionary in function space, which substantially enriches the expressiveness of the model. The proposed model can be applied to a variety of loss functions and can readily accommodate many choices of nonlinear base algorithms for multitask learning. We validate the effectiveness of allowing nonlinear model and dictionary learning through theoretical and empirical analysis. Perhaps one of the most promising future directions is to investigate use of deep neural network for the base learners [Bengio, 2012]; our approach could provide an appealing framework for learning multitask constraints over several such learners.

## Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) through the Discovery Grants Program and the NSERC Canadian Field Robotics Network (NCFRN), as well as by the Fonds de Recherche du Quebec Nature et Technologies (FQRNT).

## References


[Agarwal *et al.*, 2010] Arvind Agarwal, Samuel Gerber, and Hal Daumé III. Learning multiple tasks using manifold regularization. In *NIPS*, pages 46–54, 2010.

[Ando and Zhang, 2005] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. of Machine Learning Research*, 6:1817–1853, 2005.

[Argyriou *et al.*, 2007] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *NIPS*, pages 41–48, 2007.

[Becker *et al.*, 2013] Carlos J Becker, C. Mario Christoudias, and Pascal Fua. Non-linear domain adaptation with boosting. In *NIPS*, pages 485–493, 2013.

[Ben-David and Schuller, 2003] Shai Ben-David and Reba Schuller. Exploiting task relatedness for multiple task learning. In *COLT*, pages 567–580, 2003.

[Bengio, 2012] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. *J. of Machine Learning Research*, 27:17–37, 2012.

[Bezdek and Hathaway, 2003] James C Bezdek and Richard J Hathaway. Convergence of alternating optimization. *Neural, Parallel and Scientific Computations*, 11:351–368, December 2003.

[Bühlmann and Hothorn, 2007] Peter Bühlmann and Torsten Hothorn. Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, pages 477–505, 2007.

[Bühlmann and Yu, 2003] Peter Bühlmann and Bin Yu. Boosting with the L2 loss: Regression and classification. *J of the American Statistical Association*, 98(462):324–339, 2003.

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

[Chapelle *et al.*, 2010] Olivier Chapelle, Pannagadatta Shivaswamy, Srinivas Vadrevu, Kilian Weinberger, Ya Zhang, and Belle Tseng. Multi-task learning for boosting with application to web search ranking. In *SIGKDD*, pages 1189–1198, 2010.

[Chen *et al.*, 2011] Jianhui Chen, Jiayu Zhou, and Jieping Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *SIGKDD*, pages 42–50, 2011.

[Evgeniou and Pontil, 2004] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi-task learning. In *SIGKDD*, pages 109–117, 2004.

[Evgeniou *et al.*, 2005] Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. In *J. of Machine Learning Research*, pages 615–637, 2005.

[Freund and Schapire, 1997] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J of Computer and System Sciences*, 55(1):119–139, 1997.

[Friedman, 2001] Jerome H Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[Gong *et al.*, 2012] Pinghua Gong, Jieping Ye, and Changshui Zhang. Robust multi-task feature learning. In *SIGKDD*, pages 895–903, 2012.

[Hernández-Lobato *et al.*, 2015] Daniel Hernández-Lobato, José Miguel Hernández-Lobato, and Zoubin Ghahramani. A probabilistic model for dirty multi-task feature selection. In *ICML*, pages 1073–1082, 2015.

[Jacob *et al.*, 2008] Laurent Jacob, Jean-philippe Vert, and Francis R Bach. Clustered multi-task learning: A convex formulation. In *NIPS*, pages 745–752, 2008.

[Kang *et al.*, 2011] Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with whom to share in multi-task feature learning. In *ICML*, pages 521–528, 2011.

[Kim and Xing, 2010] Seyoung Kim and Eric P Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *ICML*, pages 543–550, 2010.

[Kumar and Daumé III, 2012] Abhishek Kumar and Hal Daumé III. Learning task grouping and overlap in multi-task learning. In *ICML*, pages 1383–1390, 2012.

[Liu *et al.*, 2009] Jun Liu, Shuiwang Ji, and Jieping Ye. Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In *UAI*, pages 339–348, 2009.

[Mason *et al.*, 2000] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Boosting algorithms as gradient descent in function space. In *NIPS*, pages 512–518, 2000.

[Maurer *et al.*, 2013] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. Sparse coding for multitask and transfer learning. In *ICML*, pages 343–351, 2013.

[Maurer *et al.*, 2014] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. An inequality with applications to structured sparsity and multitask dictionary learning. In *COLT*, pages 440–460, 2014.

[Mohri *et al.*, 2012] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, 2012.

[Müller *et al.*, 2003] Klaus-Robert Müller, Charles W Anderson, and Gary E Birch. Linear and nonlinear methods for brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):165–169, 2003.

[Nesterov, 2004] Yurii Nesterov. *Introductory Lectures on Convex Optimization*. Springer Science & Business Media, 2004.

[Obozinski *et al.*, 2010] Guillaume Obozinski, Ben Taskar, and Michael I Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, 2010.

[Xue *et al.*, 2007] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *J. of Machine Learning Research*, 8:35–63, 2007.

[Yu *et al.*, 2005] Kai Yu, Volker Tresp, and Anton Schwaighofer. Learning gaussian processes from multiple tasks. In *ICML*, pages 1012–1019, 2005.

[Zhang and Schneider, 2010] Yi Zhang and Jeff G Schneider. Learning multiple tasks with a sparse matrix-normal penalty. In *NIPS*, pages 2550–2558, 2010.

[Zhang and Yeung, 2010] Yu Zhang and Dit-Yan Yeung. A convex formulation for learning task relationships in multi-task learning. In *UAI*, pages 733–742, 2010.

[Zweig and Weinshall, 2013] Alon Zweig and Daphna Weinshall. Hierarchical regularization cascade for joint learning. In *ICML*, pages 37–45, 2013.