Multi-Task Learning with Second-Order Gradient Information

Xu Sun
MOE Key Laboratory of Computational Linguistics, Peking University

Abstract

Multi-task learning on large-scale data is costly and slow. For efficiency, our goal is accelerated multi-task learning with fast training speed. In addition, revealing task-relationship is important for real-world applications of multi-task learning. To solve this problem, we propose the *2nd-order multi-task learning (2MTL)* method, a general framework which can effectively combine 2nd-order information with multi-task learning. Moreover, our 2MTL method can effectively capture task-relationships. We test 2MTL on a wide variety of models and tasks, including large-scale sequence labeling tasks using conditional random fields and classification tasks using logistic regression. Results show that 2MTL is very close to the empirical optimum in a few passes, and it significantly outperforms existing methods in terms of both accuracy and training time.

1 Introduction

A real world classification task can often consist of multiple correlated subtasks, and multi-task learning (MTL) can be helpful in such scenarios (Caruana, 1997; Bakker and Heskes, 2003). For multi-task learning, the goal is to jointly learn the related tasks so as to improve generalization across most or all tasks. In this paper, out target is adaptive multi-task learning methods, which not only optimize model weights, but also adaptively learn task relationships (similarities among tasks) from data. Adaptive multi-task learning is especially meaningful for real-world datasets which lack prior knowledge on task relationships.

Unfortunately, our preliminary experiments on adaptive multi-task learning demonstrate that the inter-task interactions can be quite expensive and the large-scale adaptive multi-task learning can be slow. For example, in human activity recognition where each person forms a task, there can be massive number of tasks (persons) and adaptive multi-task learning is costly.

For efficiency, we consider accelerated adaptive multi-task learning with extremely fast convergence speed, so that the training can be accomplished (be close to empirical optimum) within a few number of passes. For this purpose, we consider exploiting 2nd-order gradient information in an efficiently way to accelerate multi-task learning. Typically, the usage of 2nd-order gradient information requires the computation of the inverse of the Hessian matrix, which is intractable for large-scale datasets with high dimensional features. To estimate 2nd-order information efficiently, our method approximates the inverse of Hessian in an incremental fashion. This is done via fast estimation of Jacobian matrix of the learning function.

To solve the adaptive learning problem and the efficiency concern, we propose the *2nd-order multi-task learning (2MTL)* method, a consistent framework which can effectively combine 2nd-order information with multi-task learning. The 2MTL is a general framework, which can flexibly combine with structured prediction models or simple classification methods. We will test 2MTL on a wide variety of models and tasks, including large-scale sequence labeling tasks using conditional random fields and classification tasks using logistic regression. We will perform experiments to show that the 2MTL is very close to empirical optimum in one pass, and it significantly outperforms existing methods.

To our knowledge, this is the first study on 2nd-order adaptive multi-task learning. In the next section, we briefly describe related work and background. Thereafter, we present the proposed method and we will make theoretical analysis on convergence. Finally, we perform experiments to verify the proposed method.

2 Related Work

First, we introduce related work on multi-task learning. Then, we briefly introduce backgrounds of conditional random fields and online learning, which will be used in this work.

2.1 Multi-Task Learning Multi-task learning has been the focus of much interest in machine learning societies over the last decade. Traditional multi-task learning methods include: sharing hidden nodes in neural networks (Baxter, 2011; Caruana, 1997); feature augmentation among interactive tasks (Daumé III, 2007); producing a common prior in hierarchical Bayesian models (Yu et al., 2005; Zhang et al., 2005); sharing parameters or common structures on the learning or predictor space (Lawrence and Platt, 2004; Ando and Zhang, 2005); multi-task feature selection (Yang et al., 2010); and matrix regularization based methods (Argyriou et al., 2007; Xue et al., 2007), among others.

Recent development of multi-task learning is online multi-task learning, started from (Dekel et al., 2006). (Dekel et al., 2006) assumes the tasks are related by a global loss function and the goal is to reduce the overall loss via online algorithm. With a similar but somewhat different motivation, (Abernethy et al., 2007) and (Agarwal et al., 2008) studied alternate formulations of online multi-task learning under traditional expert advice models. This is a formulation to exploit low dimensional common representations (Evgeniou et al., 2005; Rai and III, 2010). Online multi-task learning is also considered via reducing mistake bounds (Cavallanti et al., 2008), and via perceptron-based online multi-task learning (Saha et al., 2011).

One of our target is adaptive online multi-task learning. Our adaptive (online) multi-task learning methods not only learn model weights, but also learn task relationships simultaneously from data. More importantly, the proposed method can effectively estimate 2nd-order information, so that we can achieve very fast convergence of the multi-task learning. Finally, our proposal is a general framework which allows *non-structured* and *structured* classification.

2.2 Conditional Random Fields Conditional random fields (CRFs) are popular models for structured classification (Lafferty et al., 2001). Assuming a feature function that maps a pair of observation sequence $\boldsymbol{x}$ and label sequence $\boldsymbol{y}$ to a feature vector $\boldsymbol{f}$, the probability function is defined as follows (Lafferty et al., 2001; Sha and Pereira, 2003):

$$P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w}) = \frac{\exp\left[\boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{y},\boldsymbol{x})\right]}{\sum_{\forall \boldsymbol{y}'} \exp\left[\boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{y}',\boldsymbol{x})\right]}, \tag{2.1}$$

where $\boldsymbol{w}$ is a parameter vector.

Given a training set consisting of n labeled sequences, $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, for $i = 1 \ldots n$, parameter estimation is performed by maximizing the objective function,

$$\mathcal{L}(\boldsymbol{w}) = \sum_{i=1}^{n} \log P(\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{w}) - R(\boldsymbol{w}). \tag{2.2}$$

The second term is a regularizer, typically an L_2 norm. In what follows, we denote the conditional log-likelihood of each sample, $\log P(\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{w})$, as $\ell(i,\boldsymbol{w})$.

2.3 Online Training A representative online training method is the stochastic gradient descent (SGD) (Bottou, 1998; Bottou and Bousquet, 2008; Spall, 2005). Suppose $\hat{\mathcal{S}}$ is a randomly drawn subset of the full training set $\mathcal{S}$, the stochastic objective function is given by

$$\mathcal{L}_{stoch}(\boldsymbol{w},\hat{\mathcal{S}}) = \sum_{i\in\mathcal{S}} \ell(i,\boldsymbol{w}) - \frac{|\hat{\mathcal{S}}|}{|\mathcal{S}|}\frac{||\boldsymbol{w}||^2}{2\sigma^2}.$$

The extreme case is a batch size of 1, and it gives the maximum frequency of updates, which we adopt in this work. In this case, $|\hat{\mathcal{S}}| = 1$ and $|\mathcal{S}| = n$ (given n samples). In this case, we have

$$\mathcal{L}_{stoch}(\boldsymbol{w},\hat{\mathcal{S}}) = \ell(i,\boldsymbol{w}) - \frac{1}{n}\frac{||\boldsymbol{w}||^2}{2\sigma^2}, \tag{2.3}$$

where $\hat{\mathcal{S}} = \{i\}$. The model parameters are updated in such a way:

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} + \gamma_k \nabla_{\boldsymbol{w}^{(k)}} \mathcal{L}_{stoch}(\boldsymbol{w},\hat{\mathcal{S}}), \tag{2.4}$$

where k is the update counter, γ_k is the learning rate.

3 Proposal

In this section, we introduce the 2MTL framework. For every positive integer q, we define $\mathcal{N}_q$ =

$\{1, \ldots, q\}$. Let T be the number of tasks which we want to simultaneously learn. For each task $t \in \mathcal{N}_T$, there are n data examples $\{(\boldsymbol{x}_{t,i}, \boldsymbol{y}_{t,i}) : i \in \mathcal{N}_n\}$ available. In practice, the number of examples per task may vary but we have kept it constant for simplicity of notation. We use $\boldsymbol{D}$ to denote the $n \times T$ matrix whose t-th column is given by the vector $\boldsymbol{d}_t$ of data examples.

3.1 Model Our goal is to learn the weight vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_T$ from the data $\boldsymbol{D}$. For denotational simplicity, we assume that each of the weight vectors is of the same size f (feature dimension), and corresponds to the same ordering of features. We use $\boldsymbol{W}$ to denote the $f \times T$ matrix whose t-th column is given by the vector $\boldsymbol{w}_t$. We learn $\boldsymbol{W}$ by maximizing the objective function,

$$\mathrm{Obj}(\boldsymbol{W}, \boldsymbol{D}) \triangleq \mathrm{Likelihood}(\boldsymbol{W}, \boldsymbol{D}) - R(\boldsymbol{W}), \quad (3.5)$$

where $\mathrm{Likelihood}(\boldsymbol{W}, \boldsymbol{D})$ is the accumulative likelihood over all tasks, namely,

$$\mathrm{Likelihood}(\boldsymbol{W}, \boldsymbol{D}) = \sum_{t \in \mathcal{N}_T} \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{D}), \quad (3.6)$$

and $\mathcal{L}(\boldsymbol{w}_t, \boldsymbol{D})$ is defined as follows:

$$\mathcal{L}(\boldsymbol{w}_t, \boldsymbol{D}) \triangleq \sum_{t' \in \mathcal{N}_T} \left[\alpha_{t,t'} \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'})\right]. \quad (3.7)$$

$\alpha_{t,t'}$ is a real-valued *task-similarity*, with $\alpha_{t,t'} = \alpha_{t',t}$ (symmetric). Intuitively, a task-similarity $\alpha_{t,t'}$ measures the *similarity of patterns* between the t-th task and the t'-th task. $\mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'})$ is defined as follows:

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'}) &\triangleq \sum_{i \in \mathcal{N}_n} \log P(\boldsymbol{y}_{t',i} | \boldsymbol{x}_{t',i}, \boldsymbol{w}_t) \\ &= \sum_{i \in \mathcal{N}_n} \ell_{t'}(i, \boldsymbol{w}_t), \end{aligned} \quad (3.8)$$

where $P(\cdot)$ is a prescribed probability function. We can flexibly use any prescribed probability function. This makes our 2MTL method a flexible and general framework for no matter structured or non-structured classification tasks. In this paper, we will test on the CRF probability function (Eq. (2.1)) for structured classification tasks, and the well-known logistic regression function for non-structured classification tasks.

Finally, $R(\boldsymbol{W})$ is a regularization term for dealing with overfitting. In this paper, we simply use L_2 regularization:

$$R(\boldsymbol{W}) = \sum_{t \in \mathcal{N}_T} \frac{||\boldsymbol{w}_t||^2}{2\sigma_t^2}. \quad (3.9)$$

To summarize, our multi-task learning objective function is as follows:

$$\mathrm{Obj}(\boldsymbol{W}, \boldsymbol{D}) = \sum_{t,t' \in \mathcal{N}_T} \Big[\alpha_{t,t'} \sum_{i \in \mathcal{N}_n} \ell_{t'}(i, \boldsymbol{w}_t)\Big] - \sum_{t \in \mathcal{N}_T} \frac{||\boldsymbol{w}_t||^2}{2\sigma_t^2}.$$

To simplify denotations, we introduce a $T \times T$ matrix $\boldsymbol{A}$, such that $\boldsymbol{A}_{t,t'} \triangleq \alpha_{t,t'}$. We also introduce a $T \times T$ functional matrix $\boldsymbol{\Phi}$, such that $\boldsymbol{\Phi}_{t,t'} \triangleq \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'})$. Then, the objective function can be compactly expressed as follows:

$$\mathrm{Obj}(\boldsymbol{W}, \boldsymbol{D}) = \mathrm{tr}(\boldsymbol{A}\boldsymbol{\Phi}^\top) - \sum_{t \in \mathcal{N}_T} \frac{||\boldsymbol{w}_t||^2}{2\sigma_t^2}, \quad (3.10)$$

In the following content, we will first discussion a simple case that the task-similarity matrix $\boldsymbol{A}$ is fixed. After that, we will focus on the case that $\boldsymbol{A}$ is unknown.

3.2 2MTL with Fixed Task-Similarities The case of fixed task-similarities is important for two reasons. First, in practice the task-similarities may be derived from prior knowledge. Second, even if the task-similarities are unknown, we will present a learning algorithm that iteratively reduce the problem to a case of fixed task-similarities. With fixed task-similarities, the optimization problem is as follows:

$$\boldsymbol{W}^* = \operatorname*{argmax}_{\boldsymbol{W}} \left[\mathrm{tr}(\boldsymbol{A}^*\boldsymbol{\Phi}^\top) - \sum_{t \in \mathcal{N}_T} \frac{||\boldsymbol{w}_t||^2}{2\sigma_t^2}\right]. \quad (3.11)$$

It is clear to see that we can independently optimize $\boldsymbol{w}_t$ and $\boldsymbol{w}_{t'}$ ($t \neq t'$) given fixed task-similarities. In other words, we can independently optimize each column of $\boldsymbol{W}$ and derive $\boldsymbol{W}^*$:

$$\boldsymbol{w}_t^* = \operatorname*{argmax}_{\boldsymbol{w}_t} \psi(\boldsymbol{w}_t, \boldsymbol{D}), \quad (3.12)$$

where $\psi(\boldsymbol{w}_t, \boldsymbol{D})$ has the form as follows:

$$\psi(\boldsymbol{w}_t, \boldsymbol{D}) = \sum_{t' \in \mathcal{N}_T} \left[\alpha_{t,t'}^* \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'})\right] - \frac{||\boldsymbol{w}_t||^2}{2\sigma_t^2}. \quad (3.13)$$

3.3 2nd-Order Gradient Information The key issue of 2MTL-F is how to effectively and efficiently approximate the Hessian matrix. We present a simple yet effective method to approximate the eigenvalues of the Jacobian matrix of a fixed point iterative mapping. In 2MTL-F, assume our update formulation is as follows:

$$\boldsymbol{w}_t^{(k+1)} = \boldsymbol{w}_t^{(k)} + \boldsymbol{\eta}_t \cdot \boldsymbol{g}_t, \tag{3.14}$$

The update term $\boldsymbol{g}_t$ is derived by weighted sampling over different tasks. The weighted sampling is based on fixed task-similarities, $\boldsymbol{A}^*$. $\boldsymbol{g}_t$ has a form as follows:

$$\boldsymbol{g}_t = \sum_{t' \in \mathcal{N}_T} \left[\alpha^*_{t,t'} \nabla_{\boldsymbol{w}_t} \ell_{t'}(i_{t'}, \boldsymbol{w}_t) \right] - \frac{1}{n} \nabla_{\boldsymbol{w}_t} \frac{||\boldsymbol{w}_t||^2}{2\sigma_t^2}, \tag{3.15}$$

where $\alpha^*_{t,t'} = \boldsymbol{A}^*_{t,t'}$ and $i_{t'}$ indexes a random sample selected from $\boldsymbol{d}_{t'}$. Then, the expectation (over distribution of data) of the update term is as follows:

$$\begin{aligned}
\mathrm{E}(\boldsymbol{g}_t) &= \sum_{t' \in \mathcal{N}_T} \left\{ \alpha^*_{t,t'} \left[\frac{1}{n} \nabla_{\boldsymbol{w}_t} \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'}) \right] \right\} - \frac{1}{n} \nabla_{\boldsymbol{w}_t} \frac{||\boldsymbol{w}_t||^2}{2\sigma_t^2} \\
&= \frac{1}{n} \left\{ \sum_{t' \in \mathcal{N}_T} \left[\alpha^*_{t,t'} \nabla_{\boldsymbol{w}_t} \mathcal{L}(\boldsymbol{w}_t, \boldsymbol{d}_{t'}) \right] - \nabla_{\boldsymbol{w}_t} \frac{||\boldsymbol{w}_t||^2}{2\sigma_t^2} \right\} \\
&= \frac{1}{n} \nabla_{\boldsymbol{w}_t} \psi(\boldsymbol{w}_t, \boldsymbol{D}).
\end{aligned}$$

In addition, $\boldsymbol{\eta}_t \in \mathbb{R}^f_+$ is a positive vector-valued step size and $\cdot$ denotes component-wise (Hadamard) product of two vectors. The optimal step size is the one that asymptotically approaches to $\boldsymbol{H}_t^{-1}$, the inverse Hessian matrix of $\psi(\boldsymbol{w}_t, \boldsymbol{D})$ in our setting. To avoid actually evaluating $\boldsymbol{H}_t^{-1}$, we can approximate $\boldsymbol{H}_t^{-1}$ with its eigenvalues. We consider an update iterate as a fixed-point iterative mapping (though a stochastic one) $\mathcal{M}$. Taking partial derivative of $\mathcal{M}$ with respect to $\boldsymbol{w}_t$, we have

$$\boldsymbol{J}_t = \mathcal{M}' = \boldsymbol{I} - diag(\boldsymbol{\eta}_t) \boldsymbol{H}_t. \tag{3.16}$$

By exploiting this linear relation between Jacobian and Hessian, we can obtain approximate eigenvalues of inverse Hessian using eigenvalues of Jacobian:

$$eigen_i(\boldsymbol{H}_t^{-1}) = \frac{\boldsymbol{\eta}_t(i)}{1 - eigen_i(\boldsymbol{J}_t)}. \tag{3.17}$$

In addition, $eigen_i(\boldsymbol{J}_t)$ can be asymptotically approximated as follows

$$eigen_i(\boldsymbol{J}_t) \approx \lambda_i \triangleq \frac{\boldsymbol{w}_t^{(k+1)}(i) - \boldsymbol{w}_t^{(k)}(i)}{\boldsymbol{w}_t^{(k)}(i) - \boldsymbol{w}_t^{(k-1)}(i)}. \tag{3.18}$$

When k is sufficiently large, λ_i will be sufficiently close to $eigen_i(\boldsymbol{J}_t)$.

Hence, we can asymptotically estimate the inverse of Hessian via efficient estimation of the Jacobian matrix of fixed-point mapping. In multi-task setting, this optimization problem is a cost-sensitive optimization problem. We present the adaptive multi-task learning algorithm with 2nd-order information, called *2MTL with fixed task-similarities (2MTL-F)*, to summarize our discussion in this subsection. The 2MTL-F algorithm is shown in Figure 1 (upper). The derivation of $\frac{1}{n}$ before the regularization term was explained in Eq. (2.3). Lower-bounding $\boldsymbol{v}_i$ with β is for stability consideration during stochastic learning.

3.4 2MTL with Unknown Task-Similarities For many real-world applications, the task-similarities are hidden variables that are unknown. To solve this problem, we present an algorithm to learn the task-similarities and model weights in an alternating optimization manner. Our alternating learning algorithm with unknown task-similarities, called *2MTL*, is presented in Figure 1 (bottom). In the 2MTL learning, the 2MTL-F algorithm is employed as a subroutine. In the beginning of the 2MTL, model weights $\boldsymbol{W}$ and task-similarities $\boldsymbol{A}$ are initialized. $\boldsymbol{W}$ is then optimized to $\hat{\boldsymbol{W}}$ by using the 2MTL-F algorithm, based on the fixed $\boldsymbol{A}$. Then, in an alternative way, $\boldsymbol{A}$ is updated based on the optimized weights $\hat{\boldsymbol{W}}$. After that, $\boldsymbol{W}$ are optimized based on updated (and fixed) task-similarities. This iterative process continues until empirical convergence of $\boldsymbol{A}$ and $\boldsymbol{W}$.

In updating task-similarities $\boldsymbol{A}$ based on $\boldsymbol{W}$, a natural idea is to estimate a task-similarity $\alpha_{t,t'}$ based on the similarity between weight vectors, $\boldsymbol{w}_t$ and $\boldsymbol{w}_{t'}$. Based on prior work on kernel similarities, we study three natural similarity measures in 2MTL setting, and will compare them in experiments.

Gaussian RBF kernel (RBF): We can define Gaussian RBF kernel to estimate similarities:

$$\alpha_{t,t'} \triangleq \frac{1}{C} \exp\left(-\frac{||\boldsymbol{w}_t - \boldsymbol{w}_{t'}||^2}{2\sigma^2}\right), \tag{3.19}$$

where C is a real-valued constant for tuning the magnitude of task-similarities. Intuitively, a big C will result in "weak multi-tasking" and a small C will make "strong multi-tasking". σ is used to control the variance of a Gaussian RBF function.

Polynomial kernel (Poly): Alternatively, we can use (normalized) polynomial kernel to estimate similarities:

$$\alpha_{t,t'} \triangleq \frac{1}{C} \frac{\langle \boldsymbol{w}_t, \boldsymbol{w}_{t'} \rangle^d}{||\boldsymbol{w}_t||^d \cdot ||\boldsymbol{w}_{t'}||^d}, \tag{3.20}$$

where $\langle \boldsymbol{w}_t, \boldsymbol{w}_{t'} \rangle$ means inner product between the two vectors; d is the degree of the polynomial kernel; $||\boldsymbol{w}_t||^d \cdot ||\boldsymbol{w}_{t'}||^d$ is the normalizer. For example, when $d = 1$, the normalized kernel has exactly the form $\frac{1}{C}\cos\theta$, where θ is the the angle between $\boldsymbol{w}_t$ and $\boldsymbol{w}_{t'}$ in the Euclidean space.

Correlation (Cor): Since the *covariance* of task weight vectors is a natural way to estimate inter-task interactions, we consider using covariance information for estimating task-similarities. However, we find directly using a covariance matrix (to estimate task-similarities) faces the problem of stability in our online setting. Hence, we use the correlation matrix via normalizing the covariance matrix.

$$\alpha_{t,t'} \triangleq \frac{1}{C} cor(\boldsymbol{w}_t, \boldsymbol{w}_{t'}) = \frac{1}{C} \frac{cov(\boldsymbol{w}_t, \boldsymbol{w}_{t'})}{std(\boldsymbol{w}_t) std(\boldsymbol{w}_{t'})}, \tag{3.21}$$

where $cov(\boldsymbol{w}_t, \boldsymbol{w}_{t'})$ is the covariance between $\boldsymbol{w}_t$ and $\boldsymbol{w}_{t'}$). $std(\cdot)$ is *standard deviation*.

3.5 Convergence Analysis Theoretical analysis in adaptive online multi-task learning is a remaining problem. To deal with this problem, we make convergence analysis of the proposed 2MTL method. Extending the work of (Murata and Amari, 1999; Hsu et al., 2009) we will show that the proposed method has reasonable convergence properties.

When we have the least possible step size $\boldsymbol{\eta}_t^{k+1} = \beta \boldsymbol{\eta}_t^k$, the expectation of $\boldsymbol{w}_t$ obtained by 2MTL can be shown to be:

$$E(\boldsymbol{w}_t^{(k)}) = \boldsymbol{w}_t^* + \prod_{m=1}^{k} (\boldsymbol{I} - \boldsymbol{\eta}_t^{(0)} \beta^m \boldsymbol{H}_t(\boldsymbol{w}_t^*, \boldsymbol{D}))(\boldsymbol{w}_t^{(0)} - \boldsymbol{w}_t^*).$$

The rate of convergence is governed by the largest eigenvalue of $\boldsymbol{S}^{(k)} = \prod_{m=1}^{k} (\boldsymbol{I} - \boldsymbol{\eta}_t^{(0)} \beta^m \boldsymbol{H}_t(\boldsymbol{w}_t^*, \boldsymbol{D}))$. With those preparations, we can derive a bound of rate of convergence.

THEOREM 3.1. *Assume λ is the minimum eigenvalue of $\boldsymbol{H}_t(\boldsymbol{w}_t^*, \boldsymbol{D})$. For each weight optimization step 2MTL-F in 2MTL, the asymptotic rate of convergence is bounded by*

$$eigen(\boldsymbol{S}^{(k)}) \leq \exp\left\{ \frac{-\boldsymbol{\eta}_t^{(0)} \lambda \beta}{1 - \beta} \right\}.$$

Proof: We can show that

$$\begin{aligned} eigen(\boldsymbol{S}^{(k)}) &= \prod_{m=1}^{t} (1 - \boldsymbol{\eta}_t^{(0)} \beta^m \lambda) \\ &\leq \exp\left\{ -\boldsymbol{\eta}_t^{(0)} \lambda \sum_{m=1}^{t} \beta^m \right\}. \end{aligned}$$

Since $\sum_{m=1}^{t} \beta^m \to \frac{\beta}{1-\beta}$ when $t \to \infty$, we have

$$\begin{aligned} eigen(\boldsymbol{S}^{(k)}) &\leq \exp\left\{ -\boldsymbol{\eta}_t^{(0)} \lambda \sum_{m=1}^{t} \beta^m \right\} \\ &\to \exp\left\{ \frac{-\boldsymbol{\eta}_t^{(0)} \lambda \beta}{1 - \beta} \right\}. \end{aligned}$$

3.6 Accelerated 2MTL Learning The 2MTL learning algorithm can be further accelerated. The naive 2MTL learning algorithm waits for the convergence of the model weights $\boldsymbol{W}$ (in the 2MTL-F step) before updating the task-similarities $\boldsymbol{A}$. In practice, we can update task-similarities $\boldsymbol{A}$ before the convergence of the model weights $\boldsymbol{W}$. For example, we can update task-similarities $\boldsymbol{A}$ after running the 2MTL-F step over a small number of training passes. We will adopt this accelerated version of the 2MTL learning for experiments. In the experiment section, we will compare the (accelerated) 2MTL method with a variety of strong baseline methods.

References

Jacob Abernethy, Peter Bartlett, and Alexander Rakhlin. 2007. Multitask learning with expert advice. In *COLT'07*. Springer, volume 4539 of *Lecture Notes in Computer Science*, pages 484–498.

Alekh Agarwal, Alexander Rakhlin, and Peter Bartlett. 2008. Matrix regularization techniques for online

multitask learning. Technical Report UCB/EECS-2008-138, EECS Department, University of California, Berkeley.

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

Andreas Argyriou, Charles A. Micchelli, Massimiliano Pontil, and Yiming Ying. 2007. A spectral regularization framework for multi-task structure learning. In *Proceedings of NIPS'07*. MIT Press.

Bart Bakker and Tom Heskes. 2003. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research* 4:83–99.

Jonathan Baxter. 2011. A model of inductive bias learning. *CoRR* abs/1106.0245. Informal publication.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic, pages 440–447.

Léon Bottou. 1998. Online algorithms and stochastic approximations. *Online Learning and Neural Networks. Saad, David. Cambridge University Press* .

Léon Bottou and Olivier Bousquet. 2008. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, pages 161–168.

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

Giovanni Cavallanti, Nicolo Cesa-Bianchi, and Claudio Gentile. 2008. Linear algorithms for online multitask classification. In *COLT'08*. Omnipress, pages 251–262.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, Prague, Czech Republic, pages 256–263.

Ofer Dekel, Philip M. Long, and Yoram Singer. 2006. Online multitask learning. In *COLT'06*. Springer, volume 4005 of *Lecture Notes in Computer Science*, pages 453–467.

Theodoros Evgeniou, Charles A. Micchelli, and Massimiliano Pontil. 2005. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research* 6:615–637.

Jianfeng Gao, Xiaolong Li, Daniel Micol, Chris Quirk, and Xu Sun. 2010. A large scale ranker-based system for search query spelling correction. In *Proceedings of the 23rd International Conference on Computational Linguistics*. COLING '10, pages 358–366.

Yuichi Hattori, Masaki Takemori, Sozo Inoue, Go Hirakawa, and Osamu Sudo. 2010. Operation and baseline assessment of large scale activity gathering system by mobile device. In *Proceedings of DICOMO'10*.

Chun-Nan Hsu, Han-Shen Huang, Yu-Ming Chang, and Yuh-Jye Lee. 2009. Periodic step-size adaptation in second-order gradient descent for single-pass on-line structured learning. *Machine Learning* 77(2-3):195–224.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*. pages 282–289.

Neil D. Lawrence and John C. Platt. 2004. Learning to learn with the informative vector machine. In *ICML'04*. ACM, volume 69.

Noboru Murata and Shun-ichi Amari. 1999. Statistical analysis of learning dynamics. *Signal Processing* 74(1):3–28.

Piyush Rai and Hal Daume III. 2010. Infinite predictor subspace models for multitask learning. *Journal of Machine Learning Research - Proceedings Track* 9:613–620.

Rajat Raina, Andrew Y. Ng, and Daphne Koller. 2006. Constructing informative priors using transfer learning. In *ICML*. ACM, pages 713–720.

Avishek Saha, Piyush Rai, Hal Daumé III, and Suresh Venkatasubramanian. 2011. Online learning of multiple tasks and their relationships. In *AISTATS'10*. Ft. Lauderdale, Florida.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. pages 134–141.

James C. Spall. 2005. Introduction to stochastic search and optimization. *Wiley-IEEE* .

Xu Sun, Hisashi Kashima, Takuya Matsuzaki, and Naonori Ueda. 2010. Averaged stochastic gradient descent with feedback: An accurate, robust, and fast training method. In *Proceedings of the 10th International Conference on Data Mining (ICDM'10)*. pages 1067–1072.

Xu Sun, Hisashi Kashima, and Naonori Ueda. 2013. Large-scale personalized human activity recognition using online multitask learning. *IEEE Trans. Knowl. Data Eng.* 25(11):2551–2563.

Xu Sun, Louis-Philippe Morency, Daisuke Okanohara, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2008. Modeling latent-dynamic in shallow parsing: A latent conditional model with imrpoved inference. In *COLING 2008, 22nd International Conference on Computational Linguistics, Proceedings of the Conference, 18-22 August 2008, Manchester, UK*. pages 841–848.

Xu Sun, Houfeng Wang, and Wenjie Li. 2012. Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *Annual Meeting of the Association for Computational Linguistics (ACL)*. pages 253–262.

Ya Xue, David Dunson, and Lawrence Carin. 2007. The matrix stick-breaking process for flexible multi-task learning. In *ICML'07*. ACM, Corvalis, Oregon, pages 1063–1070.

Haiqin Yang, Irwin King, and Michael R. Lyu. 2010. Online learning for multi-task feature selection. In *Proceedings of CIKM'10*. ACM, pages 1693–1696.

Kai Yu, Volker Tresp, and Anton Schwaighofer. 2005. Learning gaussian processes from multiple tasks. In *ICML'05*. ACM, volume 119, pages 1012–1019.

Jian Zhang, Zoubin Ghahramani, and Yiming Yang. 2005. Learning multiple related tasks using latent independent component analysis. In *NIPS'05*.

2MTL with *fixed* task-similarities (**2MTL-F**)

Input: Initialize $\boldsymbol{W}^{(0)}$ with small random values that are close to 0; given $\boldsymbol{D}, \boldsymbol{A}^*, \beta$; $k \leftarrow 0$
for $t \leftarrow 1$ to T
. Initialize $\boldsymbol{\eta}^{(0)}$
. **Repeat** until convergence
. . $\boldsymbol{g}_t \leftarrow -\frac{1}{n}\nabla_{\boldsymbol{w}_t} \frac{||\boldsymbol{w}_t||^2}{2\sigma_t^2}$
. . **for** $t' \leftarrow 1$ to T
. . . Draw $i \in \mathcal{N}_n$ at random
. . . $\boldsymbol{g}_t \leftarrow \boldsymbol{g}_t + \boldsymbol{A}^*_{t,t'} \nabla_{\boldsymbol{w}_t} \ell_{t'}(i, \boldsymbol{w}_t)$
. . $\boldsymbol{w}_t^{(k+1)} \leftarrow \boldsymbol{w}_t^{(k)} + \boldsymbol{\eta}^{(k)} \cdot \boldsymbol{g}_t$
. . **if** $k+1 \bmod 2 = 0$
. . . $\boldsymbol{v}_i \leftarrow \frac{\boldsymbol{w}_t^{(k+1)}(i) - \boldsymbol{w}_t^{(k)}(i)}{\boldsymbol{w}_t^{(k)}(i) - \boldsymbol{w}_t^{(k-1)}(i)}$
. . . Lower-bounds $\boldsymbol{v}_i$ with β
. . . $\boldsymbol{\eta}^{(k+1)} \leftarrow \boldsymbol{v} \cdot \boldsymbol{\eta}^{(k)}$
. . **else**
. . . $\boldsymbol{\eta}^{(k+1)} \leftarrow \boldsymbol{\eta}^{(k)}$
. . $k \leftarrow k+1$
Output: $\forall t$, $\boldsymbol{w}_t^{(k)}$ converges to $\boldsymbol{w}_t^*$; i.e., $\boldsymbol{W}^{(k)}$ converges to $\boldsymbol{W}^*$.

2MTL with *unknown* task-similarities (**2MTL**)

Input: Initialize $\boldsymbol{W}^{(0)}, \boldsymbol{A}^{(0)}$; given $\boldsymbol{D}$; $k \leftarrow 0$
Repeat until convergence
. $\boldsymbol{W}^{(k+1)} \leftarrow$ 2MTL-F$(\boldsymbol{W}^{(k)}, \boldsymbol{A}^{(k)}, \boldsymbol{D})$
. **for** $t \leftarrow 1$ to T
. . **for** $t' \leftarrow 1$ to T
. . . Update $\boldsymbol{A}^{(k+1)}_{t,t'}$ with Eq.3.19/3.20/3.21
. $k \leftarrow k+1$
Output: $\boldsymbol{A}^{(k)}$ empirically converges to $\hat{\boldsymbol{A}}$. $\boldsymbol{W}^{(k)}$ converges to $\hat{\boldsymbol{W}}$.

Figure 1: 2MTL algorithms (using batch size of 1).