Accelerated Gradient Method for Multi-Task Sparse Learning Problem

Xi Chen* Weike Pan† James T. Kwok† Jaime G. Carbonell*
*School of Computer Science, Carnegie Mellon University Pittsburgh, U.S.A
{xichen, jgc}@cs.cmu.edu
†Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong
{weikep, jamesk}@cse.ust.hk

***Abstract*—Many real world learning problems can be recast as multi-task learning problems which utilize correlations among different tasks to obtain better generalization performance than learning each task individually. The feature selection problem in multi-task setting has many applications in fields of computer vision, text classification and bio-informatics. Generally, it can be realized by solving a L-1-infinity regularized optimization problem. And the solution automatically yields the joint sparsity among different tasks. However, due to the nonsmooth nature of the L-1-infinity norm, there lacks an efficient training algorithm for solving such problem with general convex loss functions. In this paper, we propose an accelerated gradient method based on an "optimal" first order black-box method named after Nesterov and provide the convergence rate for smooth convex loss functions. For nonsmooth convex loss functions, such as hinge loss, our method still has fast convergence rate empirically. Moreover, by exploiting the structure of the L-1-infinity ball, we solve the black-box oracle in Nesterov's method by a simple sorting scheme. Our method is suitable for large-scale multi-task learning problem since it only utilizes the first order information and is very easy to implement. Experimental results show that our method significantly outperforms the most state-of-the-art methods in both convergence speed and learning accuracy.**

***Keywords*-multi-task learning; L-1-infinity regularization; optimal method; gradient descend**

I. Introduction

The traditional learning problem is to estimate a function $f : \mathcal{X} \mapsto \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is either a continuous space for regression or a discrete space for classification. In many practical situations, a learning task can often be divided into several related subtasks. Since the related subtasks always share some common latent factors, learning them together is more advantageous than learning each one independently. Consequently, this leads to the popularity of *multi-task learning* (MTL) in recent years [1]–[4]. More formally, given M related tasks, the objective of MTL is to estimate M functions $f^{(k)} : \mathcal{X}^{(k)} \mapsto \mathcal{Y}^{(k)}$ jointly. Moreover, it is often the case that different tasks share the same input space but with different output spaces.

Feature selection for MTL has received increasing attention in machine learning community due to its applications in many high-dimensional sparse learning problems. For single task, feature selection is often performed by introducing the ℓ_1 regularization term [5], [6]. A well-known property of ℓ_1 regularization is its ability to recover sparse solutions. For feature selection task in MTL, the use of mixed norms, such as the $\ell_{1,2}$ [7]–[9] and the $\ell_{1,\infty}$ [10], [11], has been shown to yield joint sparsity on both the feature level and task level. In particular, the $\ell_{1,\infty}$ is sometimes more advantageous than the $\ell_{1,2}$ as it can often lead to an even more sparse solution.

In this paper, we mainly consider multi-task learning problem with the $\ell_{1,\infty}$ regularizer. Recently, there has been a lot of interest in this problem. However, there still lacks an efficient training algorithm for large-scale application. Turlach *et al.* [11] develop an interior point method which requires computation of Hessian matrix of the objective function. This thus limits its application due to the potentially huge memory requirement. In contrast, gradient methods only need the first order information (gradient for smooth optimization and subgradient for nonsmooth optimization), thus making them suitable for large-scale learning problems. Most recently, Quattoni *et al.* [12] propose a projected subgradient method. The convergence rate of this algorithm is only $O(1/\sqrt{t})$, where t is the number of iterations. Han *et al.* [13] propose a simple blockwise coordinate descent algorithm for multi-task Lasso. However, their algorithm lacks theoretical analysis of the convergence rate and can only handle square loss. Duchi *et al.* [14] provide another algorithm, forward looking subgradients method, for this problem. However, its convergence rate is still only $O(1/\sqrt{t})$. Recently, Ji *et al.* [15] take advantage of the composite gradient mapping [16] and propose an accelerated gradient method for trace norm minimization with a convergence rate $O(1/t^2)$. However, their goal is to solve the convex relaxation of matrix rank minimization problem instead of joint sparsity for multi-task learning.

The main difficulty for solving the $\ell_{1,\infty}$ regularized formulation of multi-task learning problem lies in the nonsmooth property of the $\ell_{1,\infty}$ regularizer. In general, projected subgradient based methods, as in [12], [14], can only achieve very slow convergence rate of $O(1/\sqrt{t})$. In this paper, we present an accelerated gradient descent algorithm with the convergence rate $O(1/t^2)$ by a variation of Nesterov's method [17]. We particularly note that Nesterov's algorithm calls a black-box oracle in the projection step at each iteration. By exploiting the structure of the $\ell_{1,\infty}$ ball, we show that the projection step can be efficiently solved by a simple

sorting procedure. In sum, our accelerated gradient method can solve the $\ell_{1,\infty}$-norm regularized problem with smooth convex loss function in $O(d(N + M \log M)/\sqrt{\epsilon})$ time, where N, M, d, ϵ denote the number of training examples, the number of tasks, the dimensionality of the feature vector, and the desired accuracy, respectively. Although we mainly consider the $\ell_{1,\infty}$ norm, the $\ell_{1,2}$ penalized learning problem can also be readily solved in our framework.

The rest of the paper is structured as follows. Section II gives some background and presents the formulation of our problem. Section III then proposes the accelerated gradient method and shows how to solve the gradient mapping update efficiently. We also briefly discuss the efficient gradient mapping update scheme for other regularizer, such as the $\ell_{1,2}$. Subsections III-A and III-B present the convergence rate and time complexity respectively. Section IV reports experiments on multi-task classification and regression. Experimental results show that the proposed method significantly outperforms the most recent state-of-the-art algorithms proposed in 2009, [12], [14]. Finally, we conclude our work and point out some potential future work.

II. Background and Notations

Assume the dataset contains N tuples, $z_i = (\mathbf{x}_i, y_i, k_i)$ for $i = \{1 \dots N\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector and $k_i \in \{1 \dots M\}$ is the indicator specifying which task the example $(\mathbf{x}_i, y_i)$ corresponds to. y_i is either a real number in regression case or $y_i \in \{-1, +1\}$ for binary classification. Our goal is to learn M linear classifiers of the form $\mathbf{w}_k^T \cdot \mathbf{x}$. In this work, we mainly consider three different types of loss:

1) square loss: $\ell_s(z, W) = (y - \mathbf{w}_k^T \cdot \mathbf{x})^2$;
2) logistic loss: $\ell_l(z, W) = \ln(1 + \exp(-y\mathbf{w}_k^T \cdot \mathbf{x}))$;
3) hinge loss: $\ell_h(z, W) = \max(0, 1 - y\mathbf{w}_k^T \cdot \mathbf{x})$.

where $z = (\mathbf{x}, y, k)$.

Let $W = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M] \in \mathbb{R}^{d \times M}$ and W^j be the jth row of W. In sparse multi-task learning, we enforce the joint sparsity across different tasks by adding the $l_{1,\infty}$ norm of the matrix W to the loss function, which leads to only a few non-zero rows of W. In sum, we formulate our problem as:

$$\min_W F(W) = f(W) + \psi(W) = \frac{1}{N} \sum_{i=1}^{N} \ell(z_i, W) + \lambda \|W\|_{1,\infty}, \tag{1}$$

where

$$\|W\|_{1,\infty} = \sum_{j=1}^{d} \|W^j\|_\infty = \sum_{j=1}^{d} \max_{1 \le k \le M} |W_{jk}|. \tag{2}$$

A natural way to solve (1) is subgradient method. Namely,

$$W_{t+1} = W_t - h_t F'(W_t), \tag{3}$$

where W_t is the solution at t's step and h_t is the step size. The most common strategy is to set $h_t = \frac{h}{\sqrt{t+1}}$. $F'(W) \in \partial F(W)$ is the subgradient of $F(W)$ at W and $\partial F(W)$ denotes the subdifferential of $F(W)$ at W [18]. According to [19], the subdifferential of sup-norms can be characterized as following:

Proposition 1: The subdifferential of $\|\cdot\|_\infty$ is:

$$\partial \|\cdot\|_\infty |_{\mathbf{x}} = \begin{cases} \{\mathbf{y} : \|\mathbf{y}\|_1 \le 1\} & \mathbf{x} = \mathbf{0}, \\ \text{conv}\{\text{sign}(x_i) e_i : |x_i| = \|\mathbf{x}\|_\infty\} & \mathbf{x} \ne \mathbf{0}. \end{cases} \tag{4}$$

where conv denotes the convex hull and e_i is the vector with one at ith entry and zeros at all other entries. Due to the additivity property of subdifferential, we can easily obtain the subgradient of $\|W\|_{1,\infty}$ and then plug into the subgradient descent procedure. However, as shown in [20], the convergence rate of subgradient method is only $O(1/\sqrt{t})$, *i.e.*

$$F(W_t) - F(W^*) \le \frac{\tau}{\sqrt{t}}, \tag{5}$$

where τ is some constant and W^* is the optimal solution.

III. Accelerated Gradient Method

For *smooth* convex functions, Nesterov [20] introduces a so-called "optimal" first order (gradient) method in the sense of complexity with the convergence rate $O(1/t^2)$. However, in our formulation (1), the objective function is *non-smooth* due to the $\ell_{1,\infty}$ regularizer. The recent unpublished manuscript by Nesterov [16] considers the minimization problem with the objective function composed of a smooth convex part and a "simple" nonsmooth convex part. Here "simple" means that we have the closed form minimizer of the sum of the nonsmooth part with a quadratic auxiliary function. The algorithm in [16] still achieves $O(1/t^2)$ convergence rate. Independently, Beck et al. [21] propose the "ISTA" algorithm for solving linear inverse problem with the same convergence rate. [22] further extends this method for the convex-concave optimization and obtains $O(1/t)$ convergence rate.

We adopt framework in [22] to provide a fast convergence rate algorithm for solving (1). Moreover, by exploiting the structure of the $\ell_{1,\infty}$ ball, we show that the generalized gradient update step in each iteration can be easily solved by a simple sorting procedure.

Firstly, we define the generalized gradient update step as following:

$$\begin{aligned} Q_L(W, W_t) =& f(W_t) + \langle W - W_t, \nabla f(W_t) \rangle \\ &+ \frac{L}{2} \|W - W_t\|_F^2 + \lambda \|W\|_{1,\infty} \\ q_L(W_t) =& \text{argmin}_W Q_L(W, W_t), \end{aligned} \tag{6}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\langle A, B \rangle = \text{Tr}(A^T B)$ denotes the matrix inner product.

The accelerated gradient method is presented in algorithm 1.

Algorithm 1 Accelerated Gradient Algorithm

Initialization: $L_0 > 0$, $\eta > 1$, $W_0 \in \mathbb{R}^{d\times M}$, $V_0 = W_0$ and $a_0 = 1$.

Iterate for $t = 0, 1, 2, \ldots$ until convergence of W_t:

1) Set $L = L_t$
2) While $F(q_L(V_t)) > Q_L(q_L(V_t), V_t)$
$$L = \eta L$$
3) Set $L_{t+1} = L$ and compute
$$W_{t+1} = \text{argmin}_W Q_{L_{t+1}}(W, V_t)$$
$$a_{t+1} = \frac{2}{t+3}$$
$$\delta_{t+1} = W_{t+1} - W_t$$
$$V_{t+1} = W_{t+1} + \frac{1-a_t}{a_t} a_{t+1}\delta_{t+1}$$

In addition, we suggest a look-ahead stopping criterion for algorithm 1. Firstly, we fix a step size h and in each iteration t, we calculate the following ratio:

$$\kappa = \frac{\max\limits_{t\le i\le t+h} F(W_i) - \min\limits_{t\le i\le t+h} F(W_i)}{\max\limits_{t\le i\le t+h} F(W_i)}. \tag{7}$$

And we stop the procedure when $\kappa \le \tau$ where τ is a prefixed constant.

Now, we focus on how to solve the generalized gradient update efficiently. Rewrite (6), we obtain that

$$\begin{aligned} q_L(V_t) = \text{argmin}_W \big(&\frac{1}{2}\|W - (W_t - \frac{1}{L}\nabla f(W_t))\|_F^2 \\ &+ \frac{\lambda}{L}\|W\|_{1,\infty}\big). \end{aligned} \tag{8}$$

For the sake of simplicity, we denote $(W_t - \frac{1}{L}\nabla f(W_t))$ as V and $\frac{\lambda}{L}$ as $\widetilde{\lambda}$. (8) then takes the following form:

$$\begin{aligned} q_L(V_t) =& \text{argmin}_W \left(\frac{1}{2}\|W - V\|_F^2 + \widetilde{\lambda}\|W\|_{1,\infty}\right) \\ =& \underset{W^1\ldots W^d}{\text{argmin}} \sum_{i=1}^{d}\left(\frac{1}{2}\|W^i - V^i\|_2^2 + \widetilde{\lambda}\|W^i\|_\infty\right), \end{aligned} \tag{9}$$

where W^i, V^i denotes the ith row of the matrix W, V respectively. Therefore, (8) can be decomposed into d separate subproblems of dimension M.

For each subproblem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w} - \mathbf{v}\|_2^2 + \widetilde{\lambda}\|\mathbf{w}\|_\infty, \tag{10}$$

since the conjugate of a quadratic function is still a quadratic function and the conjugate of the l_∞ norm is the l_1 barrier function, the dual of (10) takes the following form:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\boldsymbol{\alpha} - \mathbf{v}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{\alpha}\|_1 \le \widetilde{\lambda}. \tag{11}$$

And the vector of dual variables $\boldsymbol{\alpha}$ satisfies the relation $\boldsymbol{\alpha} = \mathbf{v} - \mathbf{w}$. (11) can be efficiently solved by a efficient projection onto the l_1 ball according to [23]. With the primal dual relationship, we present algorithm 2 for solving (10).

Algorithm 2 Algorithm for projection onto the ℓ_∞ ball

Input: A vector $\mathbf{v} \in \mathbb{R}^M$ and a scalar $\widetilde{\lambda} > 0$

1) If $\|\mathbf{v}\|_1 \le \widetilde{\lambda}$, set $\mathbf{w} = \mathbf{0}$. Return.
2) Let u_i be the absolute value of v_i, *i.e.* $u_i = |v_i|$. Sort vector $\mathbf{u}$ in the decreasing order: $u_1 \ge u_2 \ge \ldots \ge u_M$
3) Find $\hat{j} = \max\left\{j : \widetilde{\lambda} - \sum_{r=1}^{j}(u_r - u_j) > 0\right\}$

Output: $w_i = \text{sign}(v_i)\min\left(|v_i|, (\sum_{r=1}^{\hat{j}} u_r - \widetilde{\lambda})/\hat{j}\right)$, $i = 1\ldots M$

In the multi-task learning setting, the step 1 of algorithm 2 is the key step to enforce the coefficients of a feature to achieve zeros simultaneously among different tasks.

At last, we briefly describe how to solve the $\ell_{1,2}$ penalized multi-task learning problem and thus demonstrate the universality of the algorithm.

Recall that the $\ell_{1,2}$ norm of a matrix W is defined as:

$$\|W\| = \sum_j \|W^j\|_2. \tag{12}$$

Note that the key step in algorithm 1 is to efficiently compute the gradient mapping update. For the $\ell_{1,2}$ norm, the simple update rule can be derived. Similarly, we decompose the gradient mapping update into d subproblems as in (9). Each subproblem takes the following form:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w} - \mathbf{v}\|_2^2 + \widetilde{\lambda}\|\mathbf{w}\|_2. \tag{13}$$

It is easy to show that the optimal solution $\mathbf{w}^*$ must lie on the same direction of $\mathbf{v}$ and takes the form: $\mathbf{w}^* = \gamma\mathbf{v}$ with $\gamma \ge 0$. Otherwise, we can always remove the non-parallel part with respect to $\mathbf{v}$ from the vector $\mathbf{w}^*$ and achieve a lower objective value. By forming the Lagrangian dual form, the analytical solution of (13) can be easily obtained:

$$\mathbf{w}^* = \begin{cases} \left(1 - \frac{\widetilde{\lambda}}{\|\mathbf{v}\|_2}\right)\mathbf{v} & \|\mathbf{v}\|_2 > \widetilde{\lambda} \\ \mathbf{0} & \|\mathbf{v}\|_2 \le \widetilde{\lambda}. \end{cases} \tag{14}$$

A similar algorithm for $\ell_{1,2}$ regularized multi-task learning problem has also been proposed very recently [24].

A. Convergence Rate Analysis

Following the same strategy as in [21] and [22], we present the following theorem:

Theorem 1: Consider the general composite optimization problem:

$$\min_W F(W) = f(W) + \psi(W), \tag{15}$$

where f is a smooth convex function of the type $C^{1,1}_{L(f)}$, *i.e.* f is continuously differentiable and its gradient is Lipschitz continuous with the constant $L(f)$:

$$\|\nabla f(W) - \nabla f(V)\|_F \le L(f)\|W - V\|_F \quad \forall \quad W, V.$$

And $\psi(W)$ is a continuous function which is possibly nonsmooth. Furthermore, we assume the set of optimal solution is nonempty.

Let W_0 be the randomly chosen starting point, W_t, V_t be the sequences generated by algorithm 1 and W^* be any optimal solution. We assume that:

$$F(W^*) \leq F(W_t) \quad \forall\, t. \tag{16}$$

Then for any $t \geq 1$, we have

$$F(W_t) - F(W^*) \leq \frac{2\eta L(f)\|W_0 - W^*\|_F^2}{(t+1)^2}. \tag{17}$$

According to theorem 1, the number of iterations to achieve ϵ optimal solution, *i.e.*

$$F(W_t) - F(W^*) \leq \epsilon,$$

is at most $\lceil \sqrt{\frac{2\eta L(f)\|W_0 - W^*\|_F^2}{\epsilon}} - 1 \rceil$, *i.e.* $O(1/\sqrt{\epsilon})$. In other words, the convergence rate of algorithm 1 is $O(1/t^2)$.

Finally, we should point out that the hinge loss is nonsmooth which contradicts our assumption in theorem 1. Therefore, we cannot guarantee $O(1/t^2)$ convergence rate for hinge loss. It is a very challenging work to derive an algorithm with fast convergence rate for the combination of nonsmooth loss function and nonsmooth regularizer. However, we find out that, simply replacing the gradient by the subgradient of hinge loss in (6), the experiment still has impressive performance.

B. Time Complexity Analysis

For each iteration, the main computational cost is to calculate the gradient of the loss function and solve the minimization problem (6). The computation of the gradient for the above three types of loss functions lies on the calculation of vector inner product. Thus, for each data point, the time complexity for calculating the gradient is $O(d)$ and, in sum, $O(dN)$. The time complexity of algorithm 2 is $O(M \log M)$ due to the sorting procedure. We need to call d times algorithm 2 to solve (6). In sum, the total time complexity for each iteration is $O(d(N + M \log M))$. Combining the result in section III-A, the time for achieving ϵ accuracy is $O(d(N + M \log M)/\sqrt{\epsilon})$.

[23] proposes a randomized algorithm which has the expected linear time complexity to project onto the ℓ_1 ball. The similar tricks can also be applied here. Interested readers are referred to [23].

Similarly, for the $\ell_{1,2}$ norm regularizer, the total time complexity is $O(d(N + M)/\sqrt{\epsilon})$.

IV. Experiments

In this section, we perform experiments on sparse multi-task learning with $\ell_{1,\infty}$ regularization. We will compare the proposed accelerated gradient method (denoted MTL-AGM in the sequel) with two state-of-the-art algorithms, namely, the projected gradient method (denoted MTL-PGM) in [12] and the FOLOS method (denoted MTL-FOLOS) in [14].

Note that both our MTL-AGM and the MTL-FOLOS solve the following regularization problem:

$$\min_W \quad \frac{1}{N}\sum_{i=1}^{N} \ell(z_i, W) + \lambda\|W\|_{1,\infty}, \tag{18}$$

where the amount of regularization is controlled by λ. However, MTL-PGM puts the regularizer in the constraint, as:

$$\begin{aligned} \min_W \quad & \frac{1}{N}\sum_{i=1}^{N} \ell(z_i, W) \\ \text{s.t.} \quad & \|W\|_{1,\infty} \leq C, \end{aligned} \tag{19}$$

where the amount of regularization is controlled by C. It is well known that, due to the Lagrangian duality, the formulations (18) and (19) are equivalent, *i.e.* there is a one-to-one correspondence between λ and C [25]. However, it is hard to find the closed-form function to characterize this one-to-one mapping. For a relatively fair comparison, we choose (λ, C) that gives comparable level of sparsity.

A. Multi-Task Classification

In this section, we perform multi-task classification experiments on the Letter data set, which is a handwritten words data set with 45,679 examples collected from more than 180 different writers. There are 8 binary classification tasks for the handwritten letters: a vs o, a vs o, c vs e, g vs y, m vs n, f vs t, i vs j, and h vs n. Each letter is represented as an 8×16 binary pixel image. This data set has been studied in the context of multi-task learning by Obozinski *et al.* [8].

We randomly split the data into training and testing sets such that each of them contains roughly half of the entire data set. We run the algorithms for three different types of loss functions: (a) square loss; (b) logistic loss and (c) hinge loss, and then report the values of the (a) optimization objective, (b) training error, (c) testing error and (d) sparsity level. Here, sparsity level means the number of relevant features (non-zero rows) in the coefficient matrix W.

In the first experiment, we only enforce a small amount of regularization by using a small λ ($\lambda = 0.01$) and a large C ($C = 100$). This leads to the non-sparse results as shown in Figure 1. As can be seen, obviously, MTL-AGM converges much faster than MTL-FOLOS and MTL-PGM. The objective values for MTL-AGM decrease rapidly at the first few iterations and become stable after about 30 iterations for the square loss and hinge loss, and 70 iterations for the logistic loss. As for the other metrics, MTL-AGM also performs much faster than the other multi-task learning algorithms.

To achieve larger sparsity level, we increase λ to 0.05, and decrease C to 50. The corresponding experimental results are reported in Figure 2. Again, we can see that MTL-AGM

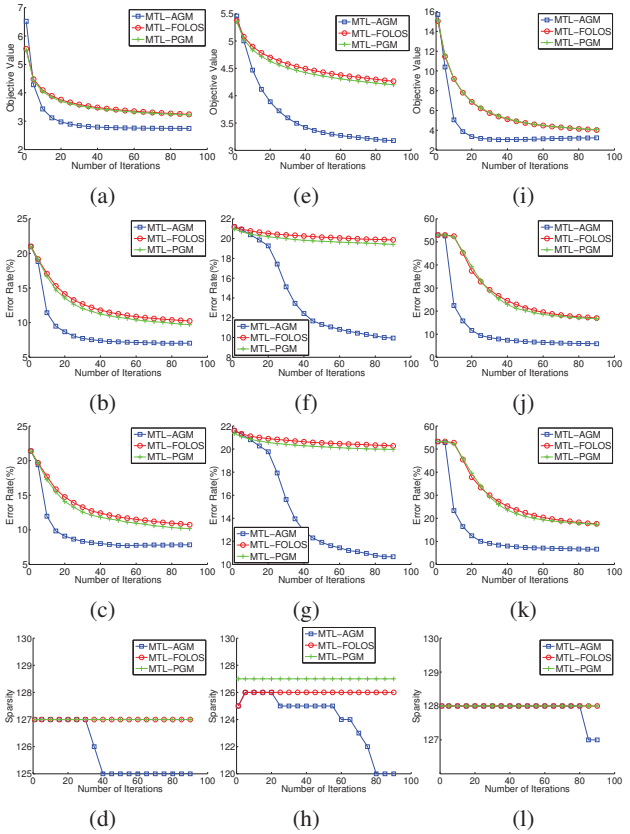


Figure 1: Performance of MTL methods on the Letter data set (with weak sparsity). 1st row: objective value; 2nd row: training error rate; 3rd row: testing error rate; 4th row: sparsity level. (a)-(d): square loss; (e)-(h): logistic loss; (i)-(l): hinge loss.

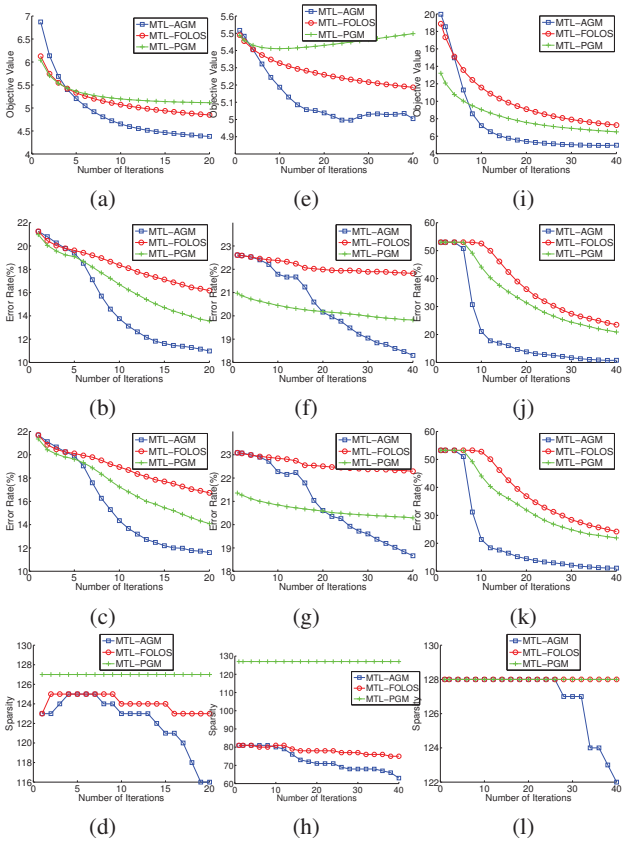


Figure 2: Performance of MTL methods on the Letter data set (with strong sparsity). 1st row: objective value; 2nd row: training error rate; 3rd row: testing error rate; 4th row: sparsity level. (a)-(d): square loss; (e)-(h): logistic loss; (i)-(l): hinge loss.

achieves significantly better performance over MTL-FOLOS and MTL-PGM on all performance metrics.

B. Multi-Task Regression

We further demonstrate the efficiency and effectiveness of MTL-AGM on a multi-task regression problem. We experiment on the commonly used School data set [8], which contains 139 regression tasks with $15,362$ instances. Again, we randomly take half of each task's data for training, and the rest for testing.

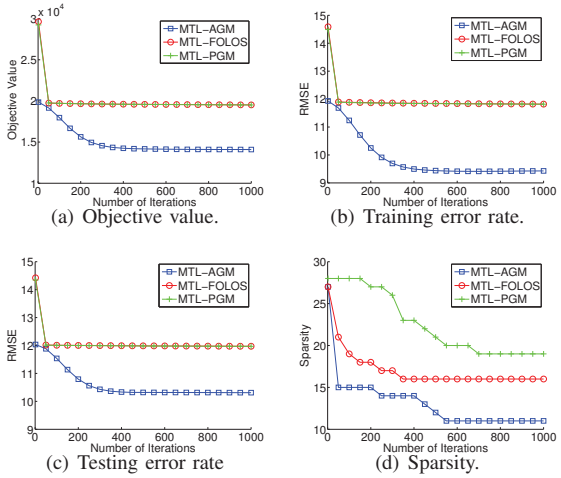
As it is a regression task, we use the square loss and report the objective value, root mean squared error (RMSE), and the sparsity level. We set $\lambda = 1$ and $C = 100$. Experimental results are shown in Figure 3. As can be seen, MTL-AGM again significantly outperforms MTL-FOLOS and MTL-PGM on all performance metrics.

In both the classification and regression experiments, the empirically much faster convergence speed strongly echoes with the theoretical guarantee of the convergence rate of the proposed algorithm.

V. Conclusion and Discussion

In this paper, we study the multi-task sparse learning problem. We mainly consider the formulation based on the $\ell_{1,\infty}$ norm regularization with the "grouping" effect such that the coefficient among different tasks can achieve zeros simultaneously. We present a very efficient gradient method by composite gradient mapping and show that the generalized gradient update in each iteration can be solved analytically by a simple sorting procedure. We also present the convergence rate analysis of the algorithm. Experimental results show that our method significantly outperforms the most state-of-the-art algorithms in both the convergence speed and learning accuracy. Moreover, our method only needs first order information, making it suitable for large-scale learning problems.

In order to further improve the practical performance of our algorithms for very large-scale setting, as in text classification, a natural idea is to design the online version of our algorithm. Since it is convex optimization method, we can easily adopt online convex optimization framework proposed in [26]. Moreover, we might take the advantage of



(a) Objective value. (b) Training error rate. (c) Testing error rate (d) Sparsity.

Figure 3: Performance of MTL methods on the School data set.

stochastic programming to further improve the convergence rate for the online version of our algorithm based on the method proposed in [27].

Another future work is to design an algorithm with the theoretically superior convergence rate for the combination of general nonsmooth convex loss, such as hinge loss, and nonsmooth regularization term. Can we design a similar algorithm and theoretically prove the fast convergence rate for nonsmooth convex loss? It is a good question for the further investigation.

REFERENCES

[1] S. Thrum and L. Pratt, *Learning to Learn*. Kluwer Academic Publishers, 1998.

[2] T. Evgeniou and M. Pontil, "Regularized multi-task learning," in *KDD '04: Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 109–117.

[3] J. Zhang, "A probabilistic framework for multi-task learning," Ph.D. dissertation, Carnegie Mellon Univerisity.

[4] R. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," RC23462, IBM T.J. Watson Research Center, Tech. Rep., 2004.

[5] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B, Methodological*, vol. 58, pp. 267–288, 1996.

[6] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific and Statistical Computing*, vol. 20, pp. 33–61, 1998.

[7] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine Learning*, vol. 73, pp. 243–272, 2008.

[8] G. Obozinski, B. Taskar, and M. Jordan, "Multi-task feature selection," Statistics Department, UC Berkeley, Tech. Rep., 2006.

[9] G. Obozinski, M. Wainwright, and M. Jordan, "High dimensional union support recovery in multivariate regression," in *Advances in Neural Information Processing Systems*, 2008.

[10] J. Tropp, A. Gilbert, and M. Strauss, "Algorithms for simultaneous sparse approximation. part ii: Convex relaxation," *Signal Processing*, vol. 86, pp. 572–588, 2006.

[11] B. Turlach, W. Venables, and S. Wright, "Simultaneous variable selection," *Technometrics*, vol. 27, pp. 349–363, 2005.

[12] A. Quattoni, X. Carreras, M. Collins, and T. Darrell, "An efficient projection for $l_{1,\infty}$ regularization," in *Proceedings of the International Conference on Machine Learning*, 2009.

[13] H. Liu, M. Palatucci, and J. Zhang, "Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery," in *Proceedings of the International Conference on Machine Learning*, 2009.

[14] J. Duchi and Y. Singer, "Online and batch learning using forward looking subgradients," 2008.

[15] S. Ji and J. Ye, "An accelerated gradient method for trace norm minimization," in *Proceedings of the International Conference on Machine Learning*, 2009.

[16] Y. Nesterov, "Gradient methods for minimizing composite objective function," *CORE Discussion Paper 2007/76*, September 2007.

[17] ——, "Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 103, no. 1, pp. 127–152, 2005.

[18] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[19] R. Rockafellar and R. Wets, *Variational analysis*. Springer-Verlag Inc., 1998.

[20] Y. Nesterov, *Introductory lectures on convex optimization: a basic course*. Kluwer Academic Pub, 2003.

[21] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci*, vol. 2, pp. 183–202, 2009.

[22] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization." submitted to SIAM Journal on Optimization.

[23] J. Duchi, S. Shalev-Shwartzand, Y. Singer, and T. Chandra, "Efficient projections onto the l1-ball for learning in high dimensions," in *Proceedings of the 25th international conference on Machine learning*, 2008.

[24] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient l2,1-norm minimization," in *Conference on Uncertainty in Artificial Intelligence*, 2009.

[25] M. Osborne, B. Presnell, and B. Turlach, "On the lasso and its dual," *Journal of Computational and Graphical Statistics*, vol. 9, pp. 319–337, 1999.

[26] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proceedings of the International Conference on Machine Learning*, 2003.

[27] G. Lan, "Efficient methods for stochastic composite optimization," School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205 USA, Tech. Rep., June, 2008.