Online Learning for Group Lasso

Haiqin Yang HQYANG@CSE.CUHK.EDU.HK
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

Zenglin Xu ZLXU@MPI-INF.MPG.DE
Cluster of Excellence MMCI, Saarland University and Max Planck Institute for Informatics, 66123 Saarbrucken, Germany

Irwin King, Michael R. Lyu {KING,LYU}@CSE.CUHK.EDU.HK
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

Abstract

We develop a novel online learning algorithm for the group lasso in order to efficiently find the important explanatory factors in a grouped manner. Different from traditional batch-mode group lasso algorithms, which suffer from the inefficiency and poor scalability, our proposed algorithm performs in an online mode and scales well: at each iteration one can update the weight vector according to a closed-form solution based on the average of previous subgradients. Therefore, the proposed online algorithm can be very efficient and scalable. This is guaranteed by its low worst-case time complexity and memory cost both in the order of $\mathcal{O}(d)$, where d is the number of dimensions. Moreover, in order to achieve more sparsity in both the group level and the individual feature level, we successively extend our online system to efficiently solve a number of variants of sparse group lasso models. We also show that the online system is applicable to other group lasso models, such as the group lasso with overlap and graph lasso. Finally, we demonstrate the merits of our algorithm by experimenting with both synthetic and real-world datasets.

1. Introduction

Group lasso (Yuan & Lin, 2006), a technique of selecting key explanatory factors in a grouped manner, is an important extension of lasso (Tibshirani, 1996). It has been successfully employed in a number of applications, such as birthweight prediction and gene finding (Yuan & Lin, 2006; Meier et al., 2008). In these applications, data may either be dominated by k-th order polynomial expansions of some inputs or contain categorical features which are usually represented as groups of dummy variables (Meier et al., 2008; Roth & Fischer, 2008; Jacob et al., 2009). Due to its advantages, group lasso has been intensively studied in statistics and machine learning (Yuan & Lin, 2006; Bach, 2008). Extensions include the group lasso for logistic regression (Meier et al., 2008), the group lasso for generalized linear models (Roth & Fischer, 2008), the group lasso with overlap between groups (Jacob et al., 2009), etc.

Despite its success in the above applications, the original group lasso model and most of its extensions have several limitations which need to be addressed: (i) the models are learned by a batch-mode training. In the training process, data are given in advance, and then they are fed into a convex optimization problem which minimizes the empirical loss with a regularization that introduces the group sparsity. However, in real-world applications, the training data may appear sequentially. (ii) Existing group lasso algorithms can only handle data up to several thousands of instances or features (Yuan & Lin, 2006; Meier et al., 2008; Roth & Fischer, 2008). While in real-world applications, data can be in large volume, over millions in both of the sample size and the feature space. Previous group lasso algorithms will fail in this situation due to their inefficiency or poor scalability. (iii) The original group lasso can only yield solutions with sparsity in the group level. It usually lacks the ability in further finding the key factors in an important group. This is a non-trivial drawback for some real-world applications, where data may be explained by the key features within the important groups. Only seeking sparsity in the group level may lose some useful information that is important to accurately interpret the data.

To address the above problems caused by the batch-mode training and poor data scalability, we develop a novel and very efficient online learning algorithm for the group lasso,

Appearing in *Proceedings of the 27^{th} International Conference on Machine Learning*, Haifa, Israel, 2010. Copyright 2010 by the author(s)/owner(s).

which updates the learning weight vector at each iteration by a closed-form solution based on the average of the previous subgradients. To the best of our knowledge, our algorithm is the first online algorithm for the group lasso. Our algorithm enjoys several good properties in terms of efficiency and effectiveness. First, the efficiency of the algorithm can be guaranteed by its low complexity in both memory space and time cost: at each iteration, the proposed algorithm only needs $\mathcal{O}(d)$ memory to store the required information and the updating process has a worst-case time complexity of $\mathcal{O}(d)$ in computation, where d is the number of features. Hence, our proposed algorithm has the potential to solve large-scale problems. Second, as the accuracy guarantee, we provide the convergence rate for both the regret bound and the bound of the learning weight vector for the proposed algorithm.

In order to seek the group lasso with more sparsity in both the group level and the individual feature level, we successfully extend the algorithm to solve the *sparse group lasso* problem (Friedman et al., 2010) and propose the *enhanced sparse group lasso* model. We further derive closed-form solutions to update the weight vectors in both models. Our algorithm framework can also be easily extended to solve the group lasso with overlap and the graph lasso problems (Jacob et al., 2009). Therefore, this suggests the good applicability of our proposed algorithm in that it can be employed to solve a large family of group lasso algorithms. Finally, experiments on both synthetic and real-world datasets demonstrate the advantages of the proposed online algorithm.

2. Related Work

In the following, we mainly review the related work on online learning algorithms.

Online learning has been extensively studied in machine learning area in recent years (Zinkevich, 2003; Bottou & LeCun, 2003; Shalev-Shwartz & Singer, 2006; Fink et al., 2006; Amit et al., 2006; Crammer et al., 2006; Bottou & Bousquet, 2007; Dredze et al., 2008; Hu et al., 2009; Zhao et al., 2009). These methods can be cast into different categories. One family of online learning algorithms is based on the criterion of maximum margin (Shalev-Shwartz & Singer, 2006; Dredze et al., 2008), which repeatedly chooses the hyperplane that correctly classifies the training samples with the maximum margin or updates the decision boundary when a new sample is misclassified or when its classification score does not exceed some predefined margin. Another family of online learning algorithms is solved by the stochastic gradient method (Zinkevich, 2003; Bottou & LeCun, 2003), where the weight vector is updated based on the subgradient of the coming sample and projected back to the constraint space if needed. An attractiveness of stochastic gradient decent methods is that their runtime may not depend at all on the number of examples (Bottou & Bousquet, 2007; Shalev-Shwartz & Srebro, 2008). Although various online learning algorithms have been proposed, there is no online learning algorithm developed for the group lasso yet.

More recently, online learning algorithms on minimizing the summation of data fitting and L_1-regularization have been proposed to yield sparse solutions (Balakrishnan & Madigan, 2008; Langford et al., 2009; Duchi & Singer, 2009; Xiao, 2009a). These algorithms are very promising in real-world applications, especially for training large-scale datasets. In (Langford et al., 2009), a truncated gradient method is proposed to truncate the elements of the learning weight vector to 0 when they cross 0 after the stochastic gradient step. Experiments on data with over 10^7 samples and 10^9 features using about 10^{11} bytes are evaluated for that method (Langford et al., 2009). In (Duchi & Singer, 2009), a forward-backward splitting method (FOBOS) is studied for solving the regularized convex optimization problem, especially the lasso problem. The algorithm of FOBOS consists of two steps: performing an unconstrained gradient descent step first and then minimizing a regularization term while keeping the solution close to the result of the first phase. In (Xiao, 2009a), the regularized dual averaging method is proposed to solve the lasso problem, where the learning weight is updated based on the average of all calculated subgradients of the loss functions. The efficiency of the above methods motivates us to propose an online learning algorithm for the group lasso.

3. Group Lasso

Given a training dataset consisting of N independent and identically distributed observations, $\{\mathbf{z}_i = (\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional vector and $y_i \in \{-1, 1\}$ for the binary classification problem or $y_i \in \mathbb{R}$ for the regression problem. Suppose that these d features are divided into G groups with d_g, the number in g-th group. Hence, we can rewrite $\mathbf{x}_i = (\mathbf{x}_i^{1\top}, \ldots, \mathbf{x}_i^{G\top})^\top$ with the group of variables $\mathbf{x}_i^g \in \mathbb{R}^{d_g}$, $g = 1, \ldots, G$. When $d_g = 1$ for all groups, the data do not form a group in the feature space.

The *lasso* algorithm (Tibshirani, 1996) is a linear regression model that selects the variables individually and it cannot find the key factors in the grouped mode. Correspondingly, the *group lasso* algorithm (Yuan & Lin, 2006) is proposed to select a subset of important factors for producing accurate prediction. Concretely, it is to seek the weight $\mathbf{w}$ and the bias b in $f(\mathbf{x}) = \sum_{g=1}^G \mathbf{w}^{g\top}\mathbf{x}^g + b$, by solving the following optimization problem

$$\min_{\mathbf{w}} \quad \sum_{i=1}^N l(\mathbf{w},\ \mathbf{z}_i) + \Omega_\lambda(\mathbf{w}), \tag{1}$$

where $\mathbf{w} = (\mathbf{w}^{1\top}, \ldots, \mathbf{w}^{G\top})^\top$. In (1), since the bias usually can be absorbed by the weight without penalty, we only consider the optimization on the weight vector $\mathbf{w}$.

Various loss functions, e.g., the squared loss, the logit loss, have been adopted for $l(\cdot)$ (Yuan & Lin, 2006; Meier et al., 2008). The loss functions are usually assumed convex.

In (1), $\Omega_\lambda(\cdot)$ defines the regularization on the weight. In the group lasso, the "groupwise" L_2-norm is adopted as the regularizer, i.e.,

$$\Omega_\lambda(\mathbf{w}) = \lambda \sum_{g=1}^{G} \sqrt{d_g} \|\mathbf{w}^g\|_2, \tag{2}$$

where the trade-off constant $\lambda \geq 0$ is to balance between the loss and the regularization term. The value $\sqrt{d_g}$ accounts for the varying group sizes and $\|\cdot\|_2$ is the Euclidean norm.

Remark 1. The regularizer in (2) makes the model act as the lasso at the group level: a large λ may make a whole group of predictors drop out of the model. As $d_g = 1$ for all the groups, the group lasso is equivalent to the lasso.

Remark 2. To introduce group sparsity, it is also possible to impose other joint regularization on the weight, e.g., the $L_{1,\infty}$-norm (Quattoni et al., 2009).

Remark 3. The group lasso regularizer has also been extended to use in Multiple Kernel Learning (MKL) (Bach, 2008). The consistency analysis on the connection between the group lasso and MKL can be referred to the paper (Bach, 2008).

Here, we can further introduce *sparse group lasso* as that in (Friedman et al., 2010)

$$\Omega_{\lambda,\mathbf{r}}(\mathbf{w}) = \lambda \sum_{g=1}^{G} \left(\sqrt{d_g} \|\mathbf{w}^g\|_2 + r_g \|\mathbf{w}^g\|_1 \right), \tag{3}$$

where $r_g > 0$, for $g = 1, \ldots, G$, is a constant balancing the L_2-norm against the L_1-norm in each group. By imposing L_1-norm in each group, the sparse group lasso can further yield sparse solutions in the selected group.

To solve the optimization with the group lasso, various methods, e.g., Group LARs (Yuan & Lin, 2006), block co-ordinate descent (Meier et al., 2008), active set algorithm (Roth & Fischer, 2008), have been proposed. Some batch-mode training methods for group lasso penalties also have been proposed, e.g., (Liu et al., 2009; Kowalski et al., 2009). Interested readers can read the above papers and references therein.

4. Online Learning for Group Lasso

Algorithm 1 Online learning algorithm for group lasso

```
Input:
  • $\mathbf{w}_0 \in \mathbb{R}^d$, and a strongly convex function $h(\mathbf{w})$ with
    modulus 1 such that
        $\mathbf{w}_0 = \arg\min_{\mathbf{w}} h(\mathbf{w}) \in \arg\min_{\mathbf{w}} \Omega(\mathbf{w})$ .   (4)
  • Given const $\lambda > 0$ for the regularizer.
  • Given const $\gamma > 0$ for the function $h(\mathbf{w})$.
Initialization: $\mathbf{w}_1 = \mathbf{w}_0$, $\bar{\mathbf{u}}_0 = \mathbf{0}$.
for $t = 1, 2, 3, \ldots$ do
  1. Given the function $l_t$, compute the subgradient on
     $\mathbf{w}_t$, $\mathbf{u}_t \in \partial l_t$.
  2. Update the average subgradient $\bar{\mathbf{u}}_t$:
        $\bar{\mathbf{u}}_t \; = \; \frac{t-1}{t}\bar{\mathbf{u}}_{t-1} + \frac{1}{t}\mathbf{u}_t$.
  3. Calculate the next iteration $\mathbf{w}_{t+1}$:
        $\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \Upsilon(\mathbf{w})$,   (5)
     where $\Upsilon(\mathbf{w}) = \left\{ \bar{\mathbf{u}}_t^\top \mathbf{w} + \Omega_\lambda(\mathbf{w}) + \frac{\gamma}{\sqrt{t}} h(\mathbf{w}) \right\}$.
end for
```

Inspired by recently developed first-order methods for optimizing composite functions (Nesterov, 2009) and the efficiency of the dual averaging method for minimizing the L_1-regularization in (Xiao, 2009a), we propose an online learning algorithm by adopting the dual averaging method to solve the group lasso, namely **DA-GL**. The algorithm is outlined in Algorithm 1. In this case, data come in sequence. At each time, we have to make the decision of $\mathbf{w}_T$ based on the coming data. By defining the objective up to the T-th step as

$$S_T(\mathbf{w}_T) := \frac{1}{T} \sum_{t=1}^{T} (\Omega_\lambda(\mathbf{w}_T) + l_t(\mathbf{w}_T)), \tag{6}$$

the objective of online learning for group lasso is to find $\mathbf{w}_T$ in the T-th step such that the objective up to the T-th step, $S_T(\mathbf{w}_T)$, is not much larger than $\min_{\mathbf{w},b} S_T(\mathbf{w})$, the smallest objective of any fixed decision $\mathbf{w}$ from hindsight. Note that in (6), we have used $l_t(\cdot)$ to simplify the expression of the loss induced by the t-th coming instance.

The difference between the objective value up to the T-th step and the smallest objective value from hindsight is the *regret* of the online algorithm for group lasso. We can define the *average regret* as

$$\bar{R}_T(\mathbf{w}) := \frac{1}{T} \sum_{t=1}^{T} (\Omega_\lambda(\mathbf{w}_t) + l_t(\mathbf{w}_t)) - S_T(\mathbf{w}). \tag{7}$$

Analysis of the regret bound and convergence rate is a key problem to guarantee the online learning algorithms. We will delay the analysis until Section 5.

Remark 4. The above proposed online learning for the group lasso is derived from the regularized dual averaging method in (Xiao, 2009a). We can also use the FOBOS (Duchi & Singer, 2009) to solve the online learning for the group lasso. In this case, at each iteration, the FOBOS method is to solve the following minimization problem:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \left\{ \frac{1}{2} \|\mathbf{w} - (\mathbf{w}_t - \eta_t \mathbf{u}_t)\|^2 + \eta_t \Omega(\mathbf{w}) \right\}, \quad (8)$$

where $\Omega(\mathbf{w})$ is defined as (2) for the group lasso or as (3) for the sparse group lasso. η_t is a constant term can be set to $O(1/\sqrt{t})$. It is easy to see the difference between FOBOS for the group lasso and our DA-GL algorithm: the FOBOS method scale the regularization term by a diminishing stepsize η_t while our method keep it the same.

Remark 5. In the standard group lasso, the features are assumed belonging to one and only one group, i.e., the groups are non-overlapped. If data contain overlapped groups, we can simply replicate the overlapped features as that in (Jacob et al., 2009) to obtain an enlarged dataset, then feed the data into Algorithm 1 to get the solution of the group lasso with overlap. The procedure can be performed similarly as that for the graph lasso.

The key to make Algorithm 1 efficiently solve the group lasso is that the update of the weight in (5) should be simple. Here, we first consider the calculation of the bias. In the batch-mode learning for the group lasso, data are given in advance. One can center the data to make the bias vanish. However, for online learning algorithms, the data are not preprocessed. Hence we have to calculate the bias. Here, since the bias is not regularized, it can be calculated by

$$b_{t+1} = \arg\min_{b} \left\{ \bar{b}_t b + \frac{\gamma}{2\sqrt{t}} b^2 \right\} = -\frac{\sqrt{t}}{\gamma} \bar{b}_t \,. \quad (9)$$

Next, let $[v]_+$ denote $\max\{0, v\}$. We can calculate the optimal solution of $\mathbf{w}_{t+1}$ in (5) in a closed-form for the following three group lasso models:

Theorem 1. *Given $\bar{\mathbf{u}}_t$ at each iteration, the optimal solution of (5) is updated correspondingly as follows:*

a) ***Group lasso****: $\Omega_\lambda(\mathbf{w})$ is defined in (2) for some $\lambda > 0$, and $h(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$. Then, for $g = 1, \ldots, G$, we have*

$$\mathbf{w}_{t+1}^g = -\frac{\sqrt{t}}{\gamma} \left[1 - \frac{\lambda\sqrt{d_g}}{\|\bar{\mathbf{u}}_t^g\|_2} \right]_+ \cdot \bar{\mathbf{u}}_t^g. \quad (10)$$

b) ***Sparse group lasso****: $\Omega_{\lambda,\mathbf{r}}(\mathbf{w})$ is defined in (3) for some $\lambda > 0$ and $\mathbf{r} \geq \mathbf{0}$, and $h(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$. Then we have*

$$\mathbf{w}_{t+1}^g = -\frac{\sqrt{t}}{\gamma} \left[1 - \frac{\lambda\sqrt{d_g}}{\|\mathbf{c}_t^g\|_2} \right]_+ \cdot \mathbf{c}_t^g, \quad (11)$$

where the j-th element of $\mathbf{c}_t^g$ is calculating by

$$c_t^{g,j} = \left[|\bar{u}_t^{g,j}| - \lambda r_g \right]_+ \cdot \text{sign}\,(\bar{u}_t^{g,j}),\ j = 1, \ldots, d_g. \quad (12)$$

c) ***Enhanced sparse group lasso****: $\Omega_{\lambda,\mathbf{r}}(\mathbf{w})$ is defined in (3) for some $\lambda > 0$ and $\mathbf{r} \geq \mathbf{0}$, and $h(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + \rho\|\mathbf{w}\|_1$ with $\rho \geq 0$ being a* sparsity-enhancing *parameter. Then*

$$\mathbf{w}_{t+1}^g = -\frac{\sqrt{t}}{\gamma} \left[1 - \frac{\lambda\sqrt{d_g}}{\|\tilde{\mathbf{c}}_t^g\|_2} \right]_+ \cdot \tilde{\mathbf{c}}_t^g, \quad (13)$$

where the j-th element of $\tilde{\mathbf{c}}_t^g$ is calculating by

$$\tilde{c}_t^{g,j} = \left[|\bar{u}_t^{g,j}| - \lambda r_g - \frac{\gamma\rho}{\sqrt{t}} \right]_+ \cdot \text{sign}\,(\bar{u}_t^{g,j}),\ j = 1, \ldots, d_g. \quad (14)$$

Remark 6. Equation (10) indicates that the solution for the group lasso achieves sparsity in the group level. Equation (11) implies that the solution for the sparse group lasso achieves sparsity in both the group level and the individual feature level. Since $\|\mathbf{c}_t^g\|_2 \leq \|\bar{\mathbf{u}}_t^g\|_2$, the solution for the sparse group lasso can achieve more sparsity in the group level than that in the group lasso. Similarly, the solution of the enhanced sparse group lasso achieves more sparsity in both the group and the individual feature level due to the introduced sparsity-enhancing parameter.

Remark 7. Theorem 1 indicates the simplicity of updating the weight in (5). It is noted that the algorithm only needs $\mathcal{O}(d)$ space to store the average subgradient, the weight at each iteration. In addition, it is possible to adopt the efficient implementation of lazy update in (Duchi & Singer, 2009) to avoid updating the whole weight at each time for high dimensional data.

5. Convergence and Regret Analysis

We have the following theorem providing the bound of the average regret and the weight:

Theorem 2. *Suppose there exists an optimal solution $\mathbf{w}_T^\star$ for the problem of (1) which satisfies $h(\mathbf{w}^\star) \leq D^2$ for some $D > 0$, and there exists a constant L such that $\|\bar{\mathbf{u}}_T\|_*^2 \leq L^2$ for all $T \geq 1$. Then we have the following properties for Algorithm 1:*

a) For each $T \geq 1$, the average regret is bounded by

$$\bar{R}_T \leq \left(\gamma\sqrt{T}D^2 + \frac{L^2}{2\gamma} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \right) / T \quad (15)$$

b) The sequence of primal variables are bounded by

$$\frac{1}{2}\|\mathbf{w}_{T+1} - \mathbf{w}^\star\|^2 \leq D^2 + \frac{L^2}{\gamma^2} - \frac{\sqrt{T}}{\gamma}\bar{R}_T \quad (16)$$

Table 1. Evaluation on the synthetic dataset when varying the number of training data. The best results are highlighted (achieved by paired t-test with 95% confidence level).

N_{tr}	Accuracy (%)					Average F1 (%)				
	Lasso	GL	L_1-RDA	DA-GL	DA-SGL	Lasso	GL	L_1-RDA	DA-GL	DA-SGL
25	54.2 ± 14.1	54.2± 11.4	56.6± 9.9	57.0± 11.6	**57.6**± 11.0	23.6± 8.5	37.3± 13.6	35.6± 6.3	37.2± 3.0	**37.9**± 4.5
50	58.2 ± 7.7	60.0± 6.3	59.5± 6.9	**60.9**± 6.2	**60.9**± 6.0	35.0± 9.3	**49.8**± 6.0	39.7± 6.5	49.7± 3.0	**49.8**± 4.9
100	62.7 ± 5.5	64.0± 5.1	61.7± 4.8	64.5± 4.1	**64.6**± 4.5	47.0± 7.2	**57.4**± 2.4	46.5± 9.7	57.1± 2.7	**57.4**± 5.9
500	75.6 ± 2.4	75.7± 2.3	66.2± 3.0	74.8± 2.3	**75.9**± 2.2	65.0± 2.5	65.5± 2.1	63.6± 9.7	65.2± 6.8	**81.9**± 5.3
1000	77.7 ± 1.5	77.8± 1.5	65.9± 2.0	76.3± 1.4	**77.9**± 1.6	70.1± 2.4	67.2± 2.1	64.9± 8.7	67.2± 4.7	**87.3**± 4.3
5000	**79.4** ± 0.4	**79.4**± 0.3	67.8± 1.5	78.2± 0.6	**79.4**± 0.8	88.2± 2.4	68.2± 2.0	66.8± 8.0	68.3± 2.9	**93.7**± 2.5
10^4	**80.0** ± 0.2	**80.0**± 0.1	68.0± 1.3	79.8± 0.3	**80.0**± 0.1	94.1± 2.3	69.1± 1.8	67.4± 5.5	68.4± 2.5	**94.2**± 2.1
10^5	**80.1** ± 0.1	**80.1**± 0.1	69.7± 1.2	79.9± 0.1	**80.1**± 0.1	**97.3**± 2.2	69.5± 1.7	68.1± 5.1	68.7± 2.3	**97.3**± 2.1

The proof of Theorem 2 can follow the framework developed in (Nesterov, 2009). A detailed proof can be found in (Xiao, 2009b). The bound in (15) can further be simplified as

$$\text{Bound of } (15) \le \frac{\gamma\sqrt{T}D^2 + \frac{L^2}{2\gamma}2\sqrt{T}}{T} = \frac{\gamma D^2 + \frac{L^2}{\gamma}}{\sqrt{T}}.$$

This also indicates that the best γ for the above bound is attained when $\gamma^\star = L/D$ and this leads to the *average regret* bound as $\bar{R}_T \le 2LD/\sqrt{T}$.

Hence, Algorithm 1 can achieve the optimal converge rate $O(1/\sqrt{T})$. It would be interesting to investigate that by introducing additional assumption, whether the average regret bound can be improve to $O(\log(T)/T)$ as that in (Hazan et al., 2007).

The second result of Theorem 2 gives a bound for the difference between the learned weight and the optimal weight. If $\bar{R}_T > 0$, then the bound can be tighter. However, the term of $\bar{R}_T$ in the bound cannot be simply discarded since the average regret $\bar{R}_T$ may be negative although this is unlikely for practical situation.

6. Experiments

In the following, we present experimental results to demonstrate the advantages of the online learning algorithms for the group lasso models on both synthetic and real-world datasets.

We compare the following five algorithms: the batch-mode learning algorithms for the lasso and the group lasso (GL); the online learning algorithm by the dual averaging method on the L_1 regularization (L_1-RDA) in (Xiao, 2009a); the online learning algorithm by the dual averaging method for the group lasso (DA-GL) in (10) and for the sparse group lasso (DA-SGL) in (11). We use the implementation of the R-package, `grplasso` (Meier et al., 2008) for the batch-mode learning algorithms. The online learning algorithms are implemented in Matlab. All algorithms run on a PC with 2.13 GHz dual-core CPU.

6.1. Synthetic data

We test the algorithms on various synthetic data similar to those generated in (Yuan & Lin, 2006; Meier et al., 2008; Friedman et al., 2010), including data with sparsity in the group level and the individual feature level. Our proposed online learning algorithms for the group lasso models consistently reveal merits. Due to space limitations, we only report the results on the data with sparsity both in group level and individual feature level. The goal of this experiment is to test the efficiency and effectiveness of the proposed algorithms.

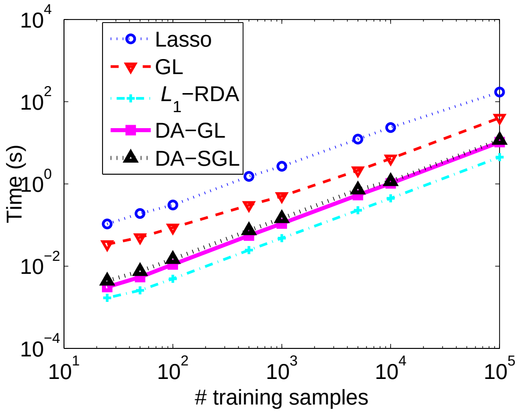

Figure 1. Log-log plot of computation time on training the synthetic dataset. The batch-model algorithms suffer from much time cost in loading large-scale datasets.

Before generating the data, we first generate a true model. A weight vector is in 100 dimension consisting of ten blocks of ten, i.e., $\mathbf{w} \in \mathbb{R}^{100}$, and $d_g = 10$, for $g = 1, \ldots, 10$. The numbers of non-zero weights in the first six blocks of 10 are 10, 8, 6, 4, 2, 1, respectively, with $w_i = \pm 1$, the sign chosen at random. The weights for the rest forty features are all zero. The bias is set to 0.

We then generate N_{tr} data points by letting $\mathbf{x}_i = L\mathbf{v}_i$, $i = 1, \ldots, N_{tr}$, where $\mathbf{v}_i \sim \mathcal{N}(0, \mathbf{I}_d)$ and L is the Cholesky decomposition of the correlation matrix, Σ. The (i, j)-th entry in the g-th group of the correlation matrix is $\Sigma^g_{i,j} = 0.2^{|i-j|}$ and zero for entries within different groups. The target value is set by $y_i = \text{sign}\left(\mathbf{w}^\top \mathbf{x}_i + \epsilon\right)$, where ϵ is a Gaussian noise with standard deviation 4.0. We randomly generate data with the size in $\{25, 50, 100, 500, 1000, 5000, 10^4, 10^5\}$. The model is evaluated on an additional test set of size N_{tr}. We repeat the experiments 50 times and average the results.

For the group lasso models, the regularization parameter λ is tested from $\lambda_{\max} * \{0.5, 0.2, 0.1, 0.05\}$, where $\lambda_{\max}$ is the maximum λ that make the weight in the group lasso vanish. For the online learning algorithms, since we know the true model, we can obtain the corresponding L and D as defined in Theorem 2 and set $\gamma = L/D$. For the DA-SGL model, r_g is set to 1 for all groups.

Table 1 reports the average results on the synthetic data in term of accuracy and the average F1 score on the true weight. The average F1 score is to verify whether the learned weight has the same sign of the true model. We calculate the F1 scores of the weight on the tasks of $+1$ vs. $\{-1, 0\}$, -1 vs. $\{+1, 0\}$, and 0 vs. $\{-1, +1\}$ and average these three F1 scores. The larger the F1 score, the more accurate in predicting the sign of the weight.

Several observations can be drawn from the results. First, the accuracy values of all algorithms increase with the number of training instances. Among them, the DA-SGL achieves the best accuracy, especially when the number is small. The DA-GL achieves slightly worse results than the DA-SGL and slightly worse results than the GL when the number is large. The two batch-mode algorithms achieve nearly the same accuracy when the number of training instances is large. Second, results about the average F1 score clearly show that the DA-SGL outperforms all the other four algorithms. With respect to F1-scores, the DA-SGL behaves similarly as the GL when the number of training instances is small and as the lasso when the number is large. The DA-SGL combines both the advantages of the lasso and the GL and is more accurate in predicting the sign of the weight. The average F1 scores on the GL and on the DA-GL are similar. Both models cannot achieve sparsity in the individual feature level and therefore, the scores are lower than the DA-SGL.

To see the efficiency of the online learning algorithms, we show the running time in Figure 1. Since the online learning algorithms and the batch-mode algorithms run in different programming platforms, the time comparison is not fair and a little bias to the R-package. However, the time cost by the online learning algorithms is clearly less than that cost by the batch-mode algorithms. These three online algorithms cost nearly the same time and the L_1-RDA costs less since the DA-GL and the DA-SGL need some calculation within each group. The batch-train algorithms cost much time in loading the data into memory when the size of the data is large.

6.2. Splice Site Detection

In order to evaluate performance of the online learning algorithms on real-world applications, we apply our algorithms in the task of splice site detection, which plays an important role in gene finding. Splice sites are the regions between coding (exons) and non-coding (introns) DNA segments. The 5' splice site (5'ss) end of an intron is called a donor splice site and the 3'ss end an acceptor splice site.

We adopt the MEMset Donar dataset for the evaluation, which is available from `http://genes.mit.edu/burgelab/maxent/ssdata/`. This dataset is widely used to demonstrate the advantages of the group lasso models (Meier et al., 2008; Roth & Fischer, 2008). It contains a training set of 8,415 true and 179,438 false human donor sites. An additional test set consists of 4,208 true and 89,717 false donor site. A sequence of a real splice site is modeled within a window over positions $[-3, 5]$ that consists of the last three bases of the exon and the first six bases of the intron. False splice sites are sequences on the DNA which match the consensus sequence at positions 0 and 1. Removing the consensus "GT" results in a sequence length of 7 with 4 level $\{A, C, G, T\}$; see (Yeo & Burge, 2004) for detailed description.

We follow the experimental setup in (Meier et al., 2008) and measure the performance by the maximum correlation coefficient (Yeo & Burge, 2004). The original training dataset is used to construct a balanced training dataset with 5,610 true and 5,610 false donar sites and an unbalanced validation set with 2,805 true and 59,804 false donor sites, which exhibits the same true/false ratio as the test set. All sites are chosen randomly without replacement such that the two sets are disjoint. The test set remains unchanged to evaluate the performance. The group lasso on the data with up to 2nd order interactions and up to 4 order interactions has been analyzed in (Meier et al., 2008) and (Roth & Fischer, 2008), respectively. As reported in (Roth & Fischer, 2008), there is no much improvement using higher order interaction. Hence we construct a model consisting of all three-way and lower order interactions, which involves 64 terms or $d = 2604$-dimensional feature space.

In the algorithms, the parameter λ is varied from $[0.01, 10]$ to produce different levels of sparsity. The parameter γ for the online learning algorithms is tuned on the validation set. The element level sparsity parameter of the DA-SGL

Table 2. Maximum correlation coefficients vs. sparsity on the MEMset Donar dataset.

% Non-zero	L1-RDA	DA-GL	DA-SGL
10	0.5632	**0.5656**	**0.5656**
40	0.6056	0.6071	**0.6082**
60	0.6481	0.6496	**0.6501**
80	0.6494	**0.6520**	**0.6520**

is set to $\sqrt{d_g}$ for simplicity. Table 2 shows the results of the online learning algorithms in terms of correlation coefficient vs. the sparsity. We can see that the online learning algorithms attain satisfactory results and they are competitive with the results in (Yeo & Burge, 2004; Meier et al., 2008). It is noted that the DA-SGL can achieve better performance in all given levels of structural sparsity. In terms of computation time, the online learning algorithms cost about 10^3 seconds for each epoch and the DA-GL costs the lest time, while the batch-train group lasso algorithm costs about quadruple of the online learning algorithms.

7. Conclusion

In this paper, we propose a novel online learning algorithm framework for the group lasso. We apply this framework for different group lasso extensions, including the sparse group lasso and our proposed enhanced sparse group lasso. We provide closed-form solutions for all the group lasso models and give the convergence rate of the average regret. We also conduct empirical evaluation on the proposed algorithms in comparison to a recently proposed online learning algorithm for L_1-regularization minimization and the batch-mode learning algorithms for the lasso and the group lasso. The results clearly demonstrate the advantages of the proposed algorithms in both efficiency and effectiveness.

There are still some remaining work: 1) to further evaluate on the FOBOS method for the group lasso in (8); 2) to further study the lazy update scheme in the FOBOS method for handling high-dimensional data; 3) to derive a faster convergence rate for the online learning algorithm by including additional assumptions, e.g., the strongly convexity assumption; and 4) to extend the online learning algorithm to solve other problems in the group lasso style.

Acknowledgments

We thank the anonymous reviewers for the valuable comments and Prof. Stephen Boyd for helpful comments on motivating the online learning algorithms when Haiqin Yang visited his research group at Stanford University. The work described in this paper was substantially supported by two grants from the Research Grants Council of the Hong Kong SAR, China (Project No. CUHK4128/08E and Project No. CUHK4158/08E).

Appendix. Proof of Theorem 1

Proof. Since the objective of (5) is component-wise, we can focus on the solution in one group, say g. In the following, we first sketch the proof of a) in Theorem 1.

The optimal $\mathbf{w}_{t+1}^g$ in (5) should be $\mathbf{w}_{t+1}^g = \kappa_g \bar{\mathbf{u}}_t^g$ with $\kappa_g \le 0$. Otherwise, we can assume for the sake of contradiction that $\mathbf{w}_{t+1}^g = \kappa_g \bar{\mathbf{u}}_t^g + \mathbf{v}^g$, where $\kappa_g \in \mathbb{R}$ and $\mathbf{v}^g$ is in the null space of $\bar{\mathbf{u}}_t^g$. It is easy to verify that $\mathbf{v}^g$ should be a zero vector.

Next, $\kappa_g > 0$ is not the optimal solution. If $\kappa_g > 0$, it can be easily verified that by setting $\kappa_g = -\kappa_g$ we can obtain a lower objective function value. Hence, the objective of (5) becomes

$$\min_{\kappa_g \le 0} \ \kappa_g \|\bar{\mathbf{u}}_t^g\|_2^2 - \lambda\sqrt{d_g}\kappa_g\|\bar{\mathbf{u}}_t^g\|_2 + \frac{\gamma}{2\sqrt{t}}\kappa_g^2\|\bar{\mathbf{u}}_t^g\|_2^2 \qquad (17)$$

By constructing the Lagrangian, $\mathcal{L}(\kappa_g, \nu)$, of the above optimization problem, we have $\nu \ge 0$ and

$$\mathcal{L} = \kappa_g \|\bar{\mathbf{u}}_t^g\|_2^2 - \lambda\sqrt{d_g}\kappa_g\|\bar{\mathbf{u}}_t^g\|_2 + \frac{\gamma}{2\sqrt{t}}\kappa_g^2\|\bar{\mathbf{u}}_t^g\|_2^2 + \nu\kappa_g.$$

The Karush-Kuhn-Tucker (KKT) condition indicates the optimal solution must satisfy

$$\frac{\partial \mathcal{L}}{\partial \kappa_g} = \|\bar{\mathbf{u}}_t^g\|_2^2 - \lambda\sqrt{d_g}\|\bar{\mathbf{u}}_t^g\|_2 + \frac{\gamma}{\sqrt{t}}\kappa_g\|\bar{\mathbf{u}}_t^g\|_2^2 + \nu = 0,$$
$$\nu\kappa_g = 0.$$

Hence, the value of $\kappa_g < 0$ iff $\lambda\sqrt{d_g} < \|\bar{\mathbf{u}}_t^g\|_2$. If $\lambda\sqrt{d_g} > \|\bar{\mathbf{u}}_t^g\|_2$, then ν must be positive and κ_g should be zero. The above analysis concludes the closed form of $\mathbf{w}_{t+1}^g$ in (10).

The sparse group lasso and the enhanced sparse group lasso have an additional L_1-norm on the weight only with different coefficients. Hence, the proof of b) and c) is similar.

Here, we just sketch the proof of b). Since the objective of (5) for the sparse group lasso is also element-wise, we can consider one entry, say j, in the g-th group. The objective of (5) on $w_{t+1}^{g,j}$ is

$$\Upsilon(w_{t+1}^{g,j}) = \bar{u}_t^{g,j} w_{t+1}^{g,j} + \lambda r_g |w_{t+1}^{g,j}| + \xi((w_{t+1}^{g,j})^2), \qquad (18)$$

where $\xi((w_{t+1}^{g,j})^2)$ is a non-negative function on $(w_{t+1}^{g,j})^2$ and $\xi(w_{t+1}^{g,j})^2) = 0$ iff $w_{t+1}^{g,j} = 0$ for all $j \in [1, d_g]$.

If $\bar{u}_t^{g,j} = 0$, obviously, the optimal solution for (18) is $w_{t+1}^{g,j} = 0$. When $\bar{u}_t^{g,j} \ne 0$, to simplify the analysis, we first assume $\bar{u}_t^{g,j} > 0$, then $w_{t+1}^{g,j}$ should be non-positive. Otherwise, if $w_{t+1}^{g,j} > 0$, we have $\Upsilon(-w_{t+1}^{g,j}) < \Upsilon(w_{t+1}^{g,j})$. It means that we can set $w_{t+1}^{g,j}$ to its negative and obtain a lower objective function value.

Next, if $\bar{u}_{t+1}^{g,j} \le \lambda r_g$, then $w_{t+1}^{g,j} = 0$ is the optimal solution. Otherwise, we have $w_{t+1}^{g,j} < 0$ and $\Upsilon(w_{t+1}^{g,j}) =$

$(\bar{u}_t^{g,j} - \lambda r_g) w_{t+1}^{g,j} + \xi((w_{t+1}^{g,j})^2) > \Upsilon(0)$. This implies that by setting $w_{t+1}^{g,j} = 0$ we can obtain a lower objective function value.

Third, $\bar{u}^{g,j} > \lambda r_g$ for all $j \in [1, d_g]$. The objective of (5) for the g-th group, $\Upsilon(\mathbf{w}_{t+1}^g)$, becomes

$$(\bar{\mathbf{u}}_t^g - \lambda r_g \mathbf{1}_{d_g})^\top \mathbf{w}_{t+1}^g + \lambda\sqrt{d_g}\|\mathbf{w}_{t+1}^g\|_2 + \frac{\gamma}{2\sqrt{t}}\|\mathbf{w}_{t+1}^g\|_2^2 \tag{19}$$

This objective function has the same structure to that of the group lasso in (5). The only difference is a slight change in the vector $\bar{\mathbf{u}}_t$. Hence, following the result of a), we can define $c^{g,j}$ as that in (12) and obtain a closed form solution in (11) for (19).

The analysis for $\bar{u}_t^{g,j} < 0$ is similar. Hence, we conclude the proof of b). □

References

Amit, Y., Shalev-Shwartz, S., and Singer, Y. Online classification for complex problems using simultaneous projections. In *NIPS*, pp. 17–24, 2006.

Bach, F. Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9: 1179–1225, 2008.

Balakrishnan, S. and Madigan, D. Algorithms for sparse linear classifiers in the massive data setting. *Journal of Machine Learning Research*, 9:313–337, 2008.

Bottou, L. and Bousquet, O. The tradeoffs of large scale learning. In *NIPS*, pp. 161–168, 2007.

Bottou, L. and LeCun, Y. Large scale online learning. In *NIPS*, 2003.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

Dredze, M., Crammer, K., and Pereira, F. Confidence-weighted linear classification. In *ICML*, pp. 264–271, 2008.

Duchi, J. and Singer, Y. Efficient learning using forward-backward splitting. In *NIPS*, pp. 495–503, 2009.

Fink, M., Shalev-Shwartz, S., Singer, Y., and Ullman, S. Online multiclass learning by interclass hypothesis sharing. In *ICML*, pp. 313–320, 2006.

Friedman, J., Hastie, T., and Tibshirani, R. A note on the group lasso and a sparse group lasso, 2010.

Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Hu, C., Kwok, J., and Pan, W. Accelerated gradient methods for stochastic optimization and online learning. In *NIPS*, pp. 781–789, 2009.

Jacob, L., Obozinski, G., and Vert, J. Group lasso with overlap and graph lasso. In *ICML*, pp. 55, 2009.

Kowalski, M., Szafranski, M., and Ralaivola, L. Multiple indefinite kernel learning with mixed norm regularization. In *ICML*, 2009.

Langford, J., Li, L., and Zhang, T. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10:777–801, 2009.

Liu, J., Ji, S., and Ye, J. Multi-task feature learning via efficient $l_{2,1}$ norm minimization. In *UAI*, 2009.

Meier, L., van de Geer, S., and Bühlmann, P. The group lasso for logistic regression. *Journal of the Royal Statistical Society, Series B*, 70(1):53–71, 2008.

Nesterov, Y. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, 2009.

Quattoni, A., Carreras, X., Collins, M., and Darrell, T. An efficient projection for $l_{1,\infty}$ regularization. In *ICML*, pp. 108, 2009.

Roth, V. and Fischer, B. The group-lasso for generalized linear models: uniqueness of solutions and efficient algorithms. In *ICML*, pp. 848–855, 2008.

Shalev-Shwartz, S. and Singer, Y. Online learning meets optimization in the dual. In *COLT*, pp. 423–437, 2006.

Shalev-Shwartz, S. and Srebro, N. SVM optimization: inverse dependence on training set size. In *ICML*, pp. 928–935, 2008.

Tibshirani, R. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996.

Xiao, L. Dual averaging method for regularized stochastic learning and online optimization. In *NIPS*, pp. 2116–2124, 2009a.

Xiao, L. Dual averaging method for regularized stochastic learning and online optimization. Technical Report MSR-TR-2009-100, Microsoft Research, 2009b.

Yeo, G. W. and Burge, C. B. Maximum entropy modeling of short sequence motifs with applications to rna splicing signals. *Journal of Computational Biology*, 11(2/3):377–394, 2004.

Yuan, M. and Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–67, 2006.

Zhao, P., Hoi, S. C., and Jin, R. Duol: A double updating approach for online learning. In *NIPS*, pp. 2259–2267, 2009.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pp. 928–936, 2003.