Pattern Recognition Letters 101 (2018) 37–43

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Multinomial classification with class-conditional overlapping sparse feature groups

Xiangrui Li, Dongxiao Zhu*, Ming Dong

Department of Computer Science, Wayne State University, Detroit 48202, USA

ARTICLE INFO

Article history:
Received 9 April 2017
Available online 10 November 2017

Keywords:
Multinomial classification
Class-conditional feature group
Feature selection

ABSTRACT

Regularized multinomial logistic model is widely used in multi-class classification problems. For high dimension data, various regularization methods achieving sparsity have been developed and applied successfully to many real-world applications such as bioinformatics, health informatics and text mining. In many cases there exist intrinsic group structures among the features. Incorporating the group information in the model can enhance model performance. In multi-class classification, different classes may relate to different feature groups. With these considerations, we propose a class-conditional regularization of the multinomial logistic model (CCSOGL) to enable the discovery of class-specific feature groups. To solve the model, we developed an efficient cyclic block coordinate descent based algorithm. We also apply our method to analyze real-world datasets to demonstrate its superior performance.

1. Introduction

Multinomial logistic regression is one of the most popular discriminative methods for multi-class classification problems. It directly models the probabilities of a sample belonging to each class. The typical approach for model training is to maximize likelihood function, which usually requires more samples than features. Otherwise the models may be overfitted and of high variances. In many modern applications such as multi-type cancer and document classification, this condition is often not satisfied as there have more features than samples in the data. A lot more parameters need to be learned as one feature corresponds to multiple parameters across multiple classes, resulting in a more complex model training and possible overfitting. To overcome this situation, feature selection, which in many applications is of great value itself and makes the model more interpretable, has attracted much interest from the research community. Various sparsity-inducing regularization methods have been developed and achieved great success in analyzing high dimensional data.

In applications such as cancer and text classification, prior knowledge is often available that there exists some intrinsic group structures among the features. As the structure of feature groups is known, incorporating this information in building a sparsity-inducing model could potentially not only lead to better models but also achieve sparsity on a larger scale. While it might be too restrictive to assume that all classes share the same structure of features, it is more reasonable to allow the class-specific structures of features and feature groups vary across classes. Moreover, as we are motivated by many real-world problems, within-group sparsity is also desired.

To further motivate our work, we briefly discuss two exemplar applications in cancer and text classification. In cancer classification, genes are grouped into overlapping gene sets (pathways). Different cancers are regulated by different pathways, and within each pathway are regulated by a subset of genes [8,11]. For another example, in document classification, key words are grouped into different topics. Different document classes are related to different topics, and each topic is represented by a set of different keywords [1]. Successful identification of the relevant feature groups and features within each group is crucial for this classification task.

In this paper, we propose a regularized multinomial logistic model, called class-conditional sparse overlapping group lasso (CCSOGL), to specifically incorporate the considerations in motivation. Our CCSOGL formulation has several contributions to the field:

- We integrate class-conditional feature group structures from prior knowledge in CCSOGL and allow different class has its own group structures. This flexibility makes CCSOGL capable of achieving class-specific sparsity pattern at the group level and further selecting relevant features within the group, which fits many applications well and enhances model interpretability.
- We present a novel algorithm that combines “’majorize-minimization” scheme and block coordinate descent algorithm

[*] Corresponding author.
E-mail address: dzhu@wayne.edu (D. Zhu).

https://doi.org/10.1016/j.patrec.2017.11.002
0167-8655/© 2017 Elsevier B.V. All rights reserved.

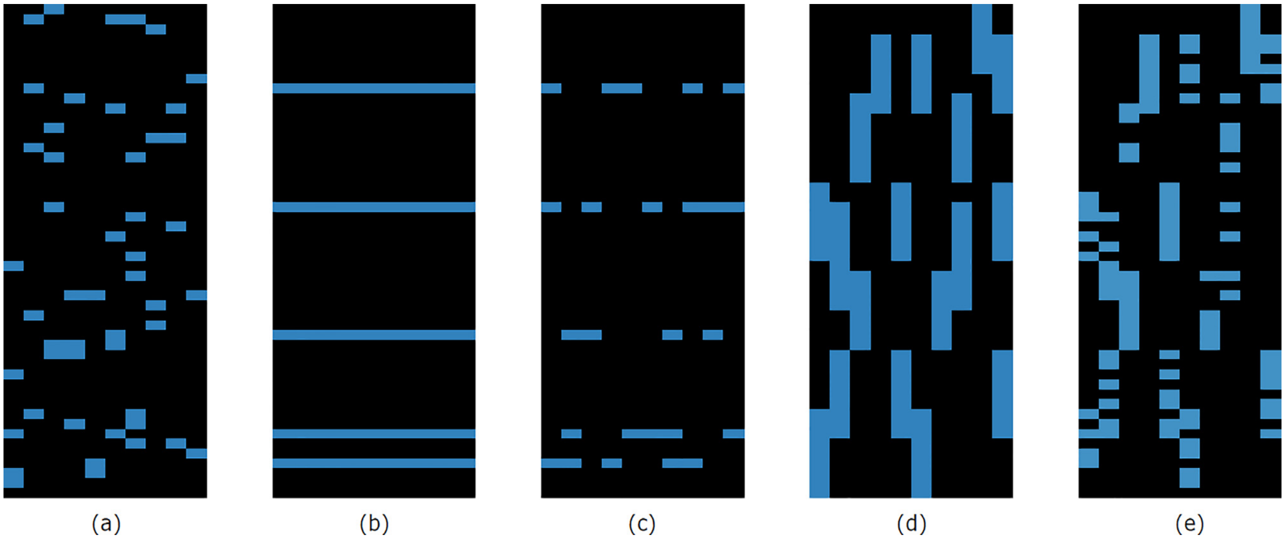

Fig. 1. An illustrative example: sparsity pattern of coefficient matrix induced by different methods: (a) Lasso [4]. (b) group lasso (GLasso): coefficients corresponding to the same feature being grouped [15]. (c) sparse group lasso (SGL), extending (b) by introducing within-group sparsity [22]. (d) CCOGL and (e) CCSOGL represent our main contributions in this paper.

in the unique setting described above so that the latter can be efficiently minimized.

- We evaluate the performance of CCSOGL with benchmark datasets and compare CCSOGL with other state-of-art sparsity-inducing methods for multinomial classification.

Related Work Lasso [19] and its variants [4,7] are among the penalized methods that induce sparsity. In cases that features are grouped according to prior knowledge, group Lasso (GLasso) [9,24] extends Lasso by introducing the group structure and be capable of achieving sparsity at the group level. Sparse group Lasso (SGL) develops group Lasso and further introduces within-group sparsity: it first selects feature groups; then within the selected groups, it selects features. To handle cases of overlapping feature groups, overlapping group Lasso (OGL) [6] and sparse overlapping group Lasso (SOGL) [14] are proposed using feature duplication. Lasso, GLasso and SGL, first developed in regression and binary logistic model, were later generalized to the multinomial problem [15,16,22]. In the GLasso multinomial model, under the implicit assumption that all classes are related to the same set of features, parameter coefficients corresponding to the same feature are grouped. (That is, in multinomial GLasso, coefficients are grouped without the need of prior knowledge.)

All methods described above do not consider the heterogeneity of feature group structure across classes and hence not be able to identify class-specific feature groups. However, our method CCSOGL explicitly incorporates heterogeneous group structure in model building and hence be able to class-wise select feature groups.

Fig. 1 presents an illustrative example of sparsity pattern induced by different regularization methods, including (a)Lasso, (b)GLasso and (c)SGL and our new methods (d)CCOGL and (e)CCSOGL (Panel (d) is a special case of (e) in our formulation. See Section 2.2). In this figure, each heat-map represents a parameter coefficient matrix with each row corresponding to one feature and each column to one class; the small cyan rectangles represent the selected features. In panel (d), each long vertical rectangle in one class represents selected feature groups specific to that class (class-specific topics in text classification or pathways in cancer classification). Panel (e) further extends (d), which selects features within selected groups. The sparsity pattern in panel (e) is often of primary interest in real-world problems as in the motivation described above. Note that Panel (d) and (e) show sparsity patterns, at the feature group level, different from (b) and (c) (“vertical” vs. “horizontal”). In (d) and (e), CCOGL and CCSOGL explicitly uses prior knowledge about feature groups (for example, pathways in cancer) and allows group structures varying across all classes.

The rest of this paper is organized as follows. In Section 2 we first review the sparse overlapping group lasso (SOGL) [14], and formulate our class-conditional sparse overlapping group lasso (CCSOGL) model in multinomial regression. We then present a block coordinate descent based algorithm to solve the model. In Section 3, the performance of our method is compared with other methods using three high dimensional datasets. In Section 4, we conclude our paper with discussion.

2. Methods

Let $\{(x_i, y_i)\}_{i=1}^n$ represent the set of n samples, where $x_i \in \mathbf{R}^P$ is the P-dimensional input vector of features for the i-th sample, and y_i is the output. The design matrix X is organized as an $n \times P$ matrix.

2.1. Sparse overlapping group lasso.

Suppose that there is a group structure $G = \{G_1, \ldots, G_J\}$ among P features $\{f_1, \ldots, f_P\}$ that each feature f_p $(1 \le p \le P)$ is assigned to at least one group G_j $(1 \le j \le J)$. In linear regression and binary classification, let β_0 and $\beta = (\beta_1, \ldots, \beta_P)^T$ denote the intercept and coefficient vector respectively.

The key idea of incorporating the grouping information behind SOGL [14] is to decompose the coefficient vector β into a sum of group-support vectors, denoted by $\omega_\beta = \{\omega^1, \ldots, \omega^J : \sum_{j=1}^J \omega^j = \beta\} \subset \mathbf{R}^P$. Each support vector ω^j satisfies a property that if $f_p \in G_j$, $\omega_p^j \in \mathbf{R}$, otherwise $\omega_p^j = 0$. For instance, in a simple case of 4 features $\{f_1,\ f_2,\ f_3,\ f_4\}$ and 3 groups $G_1 = \{f_1, f_3, f_4\}$, $G_2 = \{f_1, f_2\}$, $G_3 = \{f_2, f_4\}$, β is decomposed as a sum of three group-support vectors:

$$\beta = \omega^1 + \omega^2 + \omega^3$$

$$G_1 : \omega^1 = (\omega_1^1, 0, \omega_3^1, \omega_4^1)^T$$

$$G_2 : \omega^2 = (\omega_1^2, \omega_2^2, 0, 0)^T$$

$G_3 : \omega^3 = (0, \omega_2^3, 0, \omega_4^3)^T$.

Based on this decomposition, the sparse overlapping group lasso is defined as

$$g(\beta) = \inf_{\omega_\beta} \sum_{j=1}^{J} \left(a||\omega^j||_1 + b||\omega^j||_2\right), \tag{1}$$

where $a > 0$ and $b \geq 0$ determine the trade-off between l_1 and l_2 norm. One key property of $g(\beta)$ is that it is a norm (see *Lemma 4.1* in [14] for the proof), meaning that SOGL penalized linear and binary logit model are convex programs:

$$\min_{\beta_0, \beta} E(\beta_0, \beta) + \lambda g(\beta), \tag{2}$$

where $E(\beta_0, \beta)$ is the square loss in linear regression or negative log-likelihood in binary logit model, λ is the tuning parameter. It has been shown that the SOGL could achieve sparsity across and within groups [14].

2.2. CCSOGL classifier.

In the multi-classification problem of K classes, multinomial logistic regression models using *softmax* function calculate probabilities of multiple class memberships as below:

$$P(y = k|x) = \frac{\exp(g_k(x))}{\sum_{l=1}^{K} \exp(g_l(x))} \quad k = 1, \ldots, K, \tag{3}$$

where $g_k(x) = \beta_{k0} + x\beta_k$, $\beta_k = (\beta_{k1}, \ldots, \beta_{kP})^T \in \mathbf{R}^P$. The model parameters are represented by a pair (β_0, β) with $\beta_0 = (\beta_{10}, \ldots, \beta_{K0})$, $\beta = (\beta_{11}, \ldots, \beta_{1P}; \ldots; \beta_{K1}, \ldots, \beta_{KP}) \in \mathbf{R}^{KP}$. We say β_k is the kth vector component of β.

For n samples $(x_1, y_1), \ldots, (x_n, y_n)$, the output $y_i = k$ is encoded as $(y_{i1}, \ldots, y_{iK})$ with $y_{ik} = 1$ and $y_{ih} = 0$ for $h \neq k$, and we write $p_{ik} = P(y_i = k|x_i)$. The (scaled) negative log-likelihood function is:

$$\begin{aligned} L(\beta_0, \beta) &= -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} \cdot \ln p_{ik} \\ &= -\frac{1}{n}\sum_{i=1}^{n}\left[\sum_{k=1}^{K} y_{ik}(\beta_{k0} + x_i\beta_k) \right. \\ &\quad \left. - \ln\sum_{l=1}^{K}\exp(\beta_{l0} + x_i\beta_l)\right]. \end{aligned} \tag{4}$$

In high dimension problems ($P \gg n$), the (unregularized) maximum likelihood approach can lead to severe overfitting. Regularization of coefficients is a popular choice to identify a small number of significant features for model interpretation and improvement of prediction stability. When the feature grouping information is available, along with the consideration that features and groups of features may vary across response classes, we formulate a new sparsity-pursuit penalty “class-conditional sparse overlapping group lasso” (CCSOGL).

Suppose that $G = \{G_1, \ldots, G_J\}$ is the group structure among features $\{f_1, \ldots, f_P\}$, the coefficient vector β is, based on G, decomposed as a sum of class-dependent group support vectors $\omega_\beta = \{\omega_k^j \in \mathbf{R}^{KP} : 1 \le j \le J, 1 \le k \le K, \beta = \sum_{j=1}^{J} \omega_k^j\}$. Each $\omega_k^j = (\omega_{k,1}^j; \ldots; \omega_{k,K}^j)$, where $\omega_{k,h}^j \in \mathbf{R}^P$ for $1 \le h \le K$ is the hth vector component of ω_k^j, has the following property:

(i) For $h \neq k$, $\omega_{k,h}^j = 0$. (ii) The kth vector component $\omega_{k,k}^j$ of ω_k^j is a support vector of β_k for group G_j as in SOGL.

In other words, for the kth class,

$$\beta_k = \sum_{j=1}^{J} \omega_{k,k}^j, \; 1 \le k \le K. \tag{5}$$

Based on this decomposition, CCSOGL is defined as:

$$h(\beta) = \inf_{\omega_\beta} \sum_{j=1}^{J}\sum_{k=1}^{K} (\alpha||\omega_k^j||_1 + (1-\alpha)\sqrt{d_j}||\omega_k^j||_2), \tag{6}$$

where $0 \le \alpha < 1$ controls tradeoff between l_1 and l_2 norm, d_j is the size of the j-th group G_j.

The CCSOGL estimator is given by the convex problem:

$$\min_{\beta_0, \beta} L(\beta_0, \beta) + \lambda h(\beta). \tag{7}$$

2.3. Optimization algorithm for solving CCSOGL.

Instead of solving the optimization problem (7), we use “feature duplication” method as in [6], [14] to reduce it to a non-overlapping convex problem in the expanded feature space:

$$\min_{\beta_0, \{\omega_{k,k}^j\}_{k=1,\ldots,K; j=1,\ldots,J}} \quad S(\beta_0, \{\omega_{k,k}^j\}_{k,j}), \tag{8}$$

$$\begin{aligned} S(\beta_0, \{\omega_{k,k}^j\}_{k,j}) &= L(\beta_0, \{\omega_{k,k}^j\}_{k,j}) \\ &\quad + \lambda \sum_{j=1}^{J}\sum_{k=1}^{K} \left(\alpha||\omega_{k,k}^j||_1 + (1-\alpha)\sqrt{d_j}||\omega_{k,k}^j||_2\right) \end{aligned}$$

$$\begin{aligned} L(\beta_0, \{\omega_{k,k}^j\}_{k,j}) &= -\frac{1}{n}\sum_{i=1}^{n}\left[\sum_{k=1}^{K} y_{ik}\left(\beta_{k0} + x_i\sum_{j=1}^{J}\omega_{k,k}^j\right)\right. \\ &\quad \left. - \ln\sum_{l=1}^{K}\exp\left(\beta_{l0} + x_i\sum_{j=1}^{J}\omega_{l,l}^j\right)\right] \end{aligned}$$

For notational clarity and convenience, it is worth mentioning the following:

(1) For the k-th class, its corresponding vector of coefficients $\beta_k = (\beta_{k1}, \ldots, \beta_{kP}) \in \mathbf{R}^P$ is decomposed into $\{\omega_{k,k}^j \in \mathbf{R}^P : 1 \le j \le J\}$ according to the feature groups and β_k is recovered by $\beta_k = \sum_{j=1}^{J} \omega_{k,k}^j$ (see the example in Section 2.1). Feature duplication can be seen through $x_i\beta_k = x_i \sum_{j=1}^{J} \omega_{k,k}^j$. Also, let $x_i^j \in \mathbf{R}^{d_j}$ represent the sub-vector of the i-th sample x_i such that each component of x_i^j corresponds to its feature in th j-th group G_j respectively, we have $x_i \sum_{j=1}^{J} \omega_{k,k}^j = \sum_{j=1}^{J} x_i^j \omega_{k,k}^j$.

(2) According to the two properties of ω_k^j in Section 2.2, $\omega_{k,k}^j$ is the unique non-zero sub-vector of ω_k^j. Hence $||\omega_{k,k}^j||_m = ||\omega_k^j||_m$ $(m = 1, 2)$.

(3) We can drop out zero components in ω_k^j and $\omega_{k,k}^j$ as they don't affect the objective function (and hence $\omega_k^j = \omega_{k,k}^j \in \mathbf{R}^{d_j}$).

Hence, in the following sections, we use ω_k^j to replace $\omega_{k,k}^j$ to keep notations uncluttered based on the observations above.

Problem (8) is a convex program, and the penalty term is block separable [20]:

$$\begin{aligned} \Omega(\{\omega_k^j\}_{k,j}) &= \sum_{j=1}^{J}\sum_{k=1}^{K} (\alpha||\omega_k^j||_1 + (1-\alpha)\sqrt{d_j}||\omega_k^j||_2) \\ \Omega_k^j(\omega_k^j) &= \alpha||\omega_k^j||_1 + (1-\alpha)\sqrt{d_j}||\omega_k^j||_2. \end{aligned} \tag{9}$$

This implies that the block coordinate descent algorithm ([20], [23]) is well suited for this problem with guaranteed convergence. In the algorithm, we cycle through the parameter blocks and each iteration minimizes a subproblem keeping all but the currently chosen parameter block fixed. In the following description of the algorithm, $(\tilde{\beta}_0, \tilde{\omega}_\beta) = \{\tilde{\beta}_{k0}, \tilde{\omega}_k^j : 1 \le k \le K, 1 \le j \le J\}$ represents the numeric values learned in the previous update; $L(\beta_{k0})$ represents

$L(\beta_0, \{\omega_k^j\}_{k,j})$ as a function of β_{k0} with all coefficients being assigned with the current values except β_{k0}; $L(\omega_k^j)$, $S(\beta_{k0})$ and $S(\omega_k^j)$ are similar.

In the minimization for β_{k0} $(1 \le k \le K)$, as it is not penalized, we update β_{k0} using the Newton–Raphson formula:

$$\tilde{\beta}_{k0} \leftarrow \tilde{\beta}_{k0} - \frac{L'_{k0}}{L''_{k0}}, \tag{10}$$

where $L'_{k0} = \frac{1}{n}\sum_{i=1}^{n}(\tilde{p}_{ik} - y_{ik})$, $L''_{k0} = \frac{1}{n}\sum_{i=1}^{n}\tilde{p}_{ik}(1-\tilde{p}_{ik})$, $\tilde{p}_{ik}$ is the probability calculated by (3) at the current value $(\tilde{\beta}_0, \tilde{\omega}_\beta)$, L'_{k0} and L''_{k0} denote the derivative of 1-st and 2-nd order of $L(\beta_{k0})$ respectively.

In updating the block ω_k^j $(1 \le k \le K,\ 1 \le j \le J)$ with other blocks holding fixed, an optimization subproblem is constructed in which the objective function $Q(\omega_k^j) : \mathbf{R}^{d_j} \to \mathbf{R}$ is a sum of what is called the "majorizing function" $M(\omega_k^j)$ of $L(\omega_k^j)$ and the corresponding penalty block $\lambda\Omega_k^j$ plus a constant.

More specifically,

$$M(\omega_k^j) = L(\tilde{\beta}_0, \tilde{\omega}_\beta) + (\omega_k^j - \tilde{\omega}_k^j)^T \nabla_k^j L(\tilde{\beta}_0, \tilde{\omega}_\beta) + \frac{1}{2t}||\omega_k^j - \tilde{\omega}_k^j||_2^2, \tag{11}$$

$$Q(\omega_k^j) = M(\omega_k^j) + \lambda\Omega_k^j(\omega_k^j) + \lambda C, \tag{12}$$

$$\tilde{\omega}_k^j \leftarrow \arg\min_{\omega_k^j} Q(\omega_k^j), \tag{13}$$

where $\nabla_k^j L = \frac{\partial L}{\partial \omega_k^j} = \frac{1}{n}\sum_{i=1}^{n}(p_{ik} - y_{ik})x_i^{jT}$, t is a properly selected constant, $C = \Omega(\tilde{\omega}_\beta) - \Omega_k^j(\tilde{\omega}_k^j)$. The majorizing function $M(\omega_k^j)$ comes from the "majorize-minimization" algorithm [5] with a nice property that if t is chosen small enough, the third term $\frac{1}{2t}||\omega_k^j - \tilde{\omega}_k^j||_2^2$ will dominate the Hessian term in the Taylor expansion of $L(\omega_k^j)$. As a consequence, the following inequality holds for all $\omega_k^j \in \mathbf{R}^{d_j}$:

$$L(\omega_k^j) \le M(\omega_k^j). \tag{14}$$

Adding the penalty term on $M(\omega_k^j)$ leads to a majorizing function $Q(\omega_k^j)$ for the objective $S(\omega_k^j)$ in (8). That is, for all $\omega_k^j \in \mathbf{R}^{d_j}$,

$$S(\omega_k^j) \le Q(\omega_k^j). \tag{15}$$

Lemma 1. *The cyclic block coordinate descent algorithm for CCSOGL converges.*

Proof. Let $(\beta_0^m, \omega_\beta^m)$ and $(\beta_0^{m+1}, \omega_\beta^{m+1})$ be estimations of the parameters before and after the $(m+1)$th iteration, respectively. The only difference between $(\beta_0^m, \omega_\beta^m)$ and $(\beta_0^{m+1}, \omega_\beta^{m+1})$ is the updated block in the iteration.

If some β_{k0} is updated, it can be easily seen that $S(\beta_0^{m+1}, \omega_\beta^{m+1}) \le S(\beta_0^m, \omega_\beta^m)$.

If the updated block is ω_k^j, we have:

$$\begin{aligned} S(\beta_0^{m+1}, \omega_\beta^{m+1}) &\le Q((\omega_k^j)^{m+1}) \\ &\le Q((\omega_k^j)^m) = S(\beta_0^m, \omega_\beta^m). \end{aligned}$$

The first inequality comes from (15) and the second inequality holds due to (13). □

By completing the square in $M(\omega_k^j)$, minimizing $Q(\omega_k^j)$ is equivalent to minimizing:

$$R(\omega_k^j) = \frac{1}{2\lambda t}||\omega_k^j - [\tilde{\omega}_k^j - t\nabla_k^j L(\tilde{\beta}_0, \tilde{\omega}_\beta)]||_2^2 + \Omega_k^j(\omega_k^j). \tag{16}$$

Note that $R(\omega_k^j)$ is strictly convex, so the optimal minimizer is characterized by the first order condition [10]. This results in the following lemma:

Lemma 2. *The minimizer $\omega_k^{j\star}$ for (16) is given by the following update rule:*

$$\begin{aligned} &\textit{if } ||T_\alpha\left(\frac{\tilde{\omega}_k^j - t\nabla_k^j L(\tilde{\beta}_0, \tilde{\omega}_\beta)}{\lambda t}\right)||_2 \le (1-\alpha)\sqrt{d_j}, \\ &\omega_k^{j\star} = 0, \\ &\textit{if } ||T_\alpha\left(\frac{\tilde{\omega}_k^j - t\nabla_k^j L(\tilde{\beta}_0, \tilde{\omega}_\beta)}{\lambda t}\right)||_2 > (1-\alpha)\sqrt{d_j}, \\ &\omega_k^{j\star} = \left[1 - \frac{(1-\alpha)\sqrt{d_j}\lambda t}{||T_{\alpha\lambda t}(\tilde{\omega}_k^j - t\nabla_k^j L(\tilde{\beta}_0, \tilde{\omega}_\beta))||_2}\right] \\ &\qquad \cdot T_{\alpha\lambda t}(\tilde{\omega}_k^j - t\nabla_k^j L(\tilde{\beta}_0, \tilde{\omega}_\beta)), \end{aligned} \tag{17}$$

where $T_v(x) = \big(S(x_1, v), \ldots, S(x_{d_j}, v)\big)$ $(x \in \mathbf{R}^{d_j})$ *and* $S(u, v) =$ sign$(u)\max\{|u| - v, 0\}$ $(u \in \mathbf{R}, v \ge 0)$ *is the soft-thresholding operator.*

Proof. For ease of notation, we solve the following instead of Problem (16):

$$\min_{x \in \mathbf{R}^{d_j}} f(x) := \frac{1}{2\eta}||x - \tau||_2^2 + \lambda_1||x||_1 + \lambda_2||x||_2. \tag{18}$$

Since the object function is convex, $x^\star$ is the optimal solution if and only if

$$0 \in \partial f(x^\star), \tag{19}$$

where $\partial f(x^\star)$ is the subdifferential of $f(x)$ at $x^\star$. For any x,

$$\partial f(x) = \left\{\frac{1}{\eta}(x - \tau) + \lambda_1 a + \lambda_2 b : a \in \partial||x||_1, b \in \partial||x||_2\right\},$$

where

$$\begin{aligned} \partial||x||_1 = \{&(a_1, \ldots, a_{d_j}) \in \mathbf{R}^{d_j} : \\ &|a_i| \le 1 \text{ if } x_i = 0, a_i = \text{sign}(x_i) \text{ if } x_i \ne 0\} \end{aligned} \tag{20}$$

is the subdifferential of function $||x||_1$, and

$$\partial||x||_2 = \begin{cases} \dfrac{x}{||x||_2} & x \ne 0 \\ \{x : ||x||_2 \le 1\} & x = 0 \end{cases} \tag{21}$$

is the sub-differential of function $||x||_2$.

Case 1. If $x^\star = 0$ is the optimal solution if and only if there exists some $a \in \partial||x||_1$ satisfying

$\max\{|a_i|: 1 \le i \le d_j\} \le 1$ such that

$$||\frac{\tau}{\eta} - \lambda_1 a||_2 \le \lambda_2,$$

which is further equivalent to

$$\min\left\{||\frac{\tau}{\eta} - \lambda_1 a||_2 : a \in \partial||x||_1\right\} \le \lambda_2.$$

To obtain $\min\{||\frac{\tau}{\eta} - \lambda_1 a||_2 : a \in \partial||x||_1\}$, we only need find the minimum of each component. For each component, the minimum is given by 0 if $|\frac{\tau_i}{\eta}| \le \lambda_1$; $\frac{\tau_i}{\eta} - \lambda_1$ if $\frac{\tau_i}{\eta} > \lambda_1$; $\frac{\tau_i}{\eta} + \lambda_1$ if $\frac{\tau_i}{\eta} < -\lambda_1$.

In compact form, that is $S(\frac{\tau_i}{\eta}, \lambda_1)$, where $S(u, v) =$ sign$(u)\max\{|u| - v, 0\}$ $(u \in \mathbf{R}, v \ge 0)$ is the soft-thresholding operator.

Writing $T_v(x) = \big(S(x_1, v), \ldots, S(x_{d_j}, v)\big)$, we have

$$x^\star = 0, \text{ if } ||T_{\lambda_1}\left(\frac{\tau}{\eta}\right)||_2 \le \lambda_2. \tag{22}$$

Case 2. If $x^\star \neq 0$ is the optimal solution (some components of $x^\star$ may be zero), from (19), (20) and (21), we must have for some $a \in \partial||x||_1$,

$$\left(\frac{1}{\eta} + \frac{\lambda_2}{||x^\star||_2}\right)x^\star = -\lambda_1 a + \frac{\tau}{\eta}. \tag{23}$$

Write the equation above component-wise:

$$\left(\frac{1}{\eta} + \frac{\lambda_2}{||x^\star||_2}\right)x_i^\star = -\lambda_1 a_i + \frac{\tau_i}{\eta}. \tag{24}$$

Again, use (20), (24) is just

$$\left(\frac{1}{\eta} + \frac{\lambda_2}{||x^\star||_2}\right)x_i^\star = S\left(\frac{\tau_i}{\eta}, \lambda_1\right) = \begin{cases} 0 & \text{if } x_i^\star = 0 \\ \dfrac{\tau_i}{\eta} - \lambda_1 & \text{if } x_i^\star > 0 \\ \dfrac{\tau_i}{\eta} + \lambda_1 & \text{if } x_i^\star < 0 \end{cases} \tag{25}$$

Consequently (23) is

$$\left(\frac{1}{\eta} + \frac{\lambda_2}{||x^\star||_2}\right)x^\star = T_{\lambda_1}\left(\frac{\tau}{\eta}\right). \tag{26}$$

Take norm on both side, we get

$$||x^\star||_2 = \eta \cdot \left[||T_{\lambda_1}\left(\frac{\tau}{\eta}\right)||_2 - \lambda_2\right] = ||T_{\lambda_1\eta}(\tau)||_2 - \lambda_2\eta. \tag{27}$$

Plug it back into (26),

$$x^\star = \left[1 - \frac{\lambda_2\eta}{||T_{\lambda_1\eta}(\tau)||_2}\right] \cdot T_{\lambda_1\eta}(\tau), \quad \text{if } ||T_{\lambda_1}\left(\frac{\tau}{\eta}\right)||_2 > \lambda_2. \tag{28}$$

With the substitution of $\eta = \lambda t$, $\lambda_1 = \alpha$, $\lambda_2 = (1-\alpha)\sqrt{d_j}$ and $\tau = \tilde{\omega}_k^j - t\nabla_k^j L(\tilde{\beta}_0, \tilde{\omega}_\beta)$, (22) and (28) give the solution to (16). □

From Lemma 2, we see that our CCSOGL model promotes across-group and within-group sparsity: CCSOGL first select feature groups; within the selected feature groups, CCSOGL then performs feature selection. As feature groups are selected class-wise, CCSOGL can indeed achieve the sparsity pattern shown in Panel (e) of Fig. 1.

Integrating all of above leads to Algorithm 1 for solving our CCSOGL penalized multinomial logistic model.

Algorithm 1 Cyclic block coordinate descent for CCSOGL.

```
Initialize $(\beta_0, \omega_\beta)$
repeat
    for $k = 1$ to $K$
        $\beta_{k0} \leftarrow \arg\min L(\beta_{k0})$, using Newton–Raphson
                formula (10)
        for $j = 1$ to $J$
            update each block $\omega_k^j$ according to Lemma 2.
until converge
```

3. Application

In this section, we evaluate and validate our method and compare its performance with selected competing classification methods using several publicly available datasets. Tested methods include Lasso, GLasso, SGLasso, *l*1-regularized *l*2-loss SVM and CCSOGL. The Lasso and GLasso were implemented using the R package *glmnet* [4]; SGL was implemented in R package *msgl*; L1-regularized SVM (L2-loss) was implemented in R package *LiblineaR*. The CCSOGL was implemented in C++ in house interfacing with R through the R package *Rcpp* and *RcppArmadillo* [3], and codes will be publicly available on github.

Table 1
Summary of datasets used in our analysis. In the table, *K*, *n*, *p* and *J* represent number of classes, number of samples, number of features and number of feature sets, respectively; "Collection" represents the gene set collection used from MSigDB.

Dataset	*K*	*n*	*p*	*J*	Collection
NCI60	8	58	2654	103	C4 CM
Brain	5	42	2035	111	C5
Breast	4	195	3582	43	C6

3.1. Data description.

We first used three gene expression datasets to evaluate CCSOGL. The details of datasets are as follows:

- *NCI-60*[1]: NCI60 contains gene expression levels from 60 cell lines with 9 different types of cancer: 6 leukemia, 8 melanoma, 9 non-small-cell lung carcinoma, 7 colon, 6 central nervous system, 8 renal, 8 breast, 2 prostate and 6 ovarian. We removed samples of prostate cancer due to very small class size, resulting in 58 samples from 8 classes remain in analysis. More details for NCI60 can be found in [17].
- *Brain Cancer*[2]: This dataset consists of gene expression profiles from 42 patients with different brain cancer types of the central nervous system. The samples are divided into 5 classes: 8 primitive neuroectodermal tumors (PNET), 10 atypical teratoid/rhabdoid tumors (AT/RT), 10 medulloblastomas, 10 malignant gliomas and 4 human cerebella. See [12] for more information.
- *Breast Cancer Subclass*: The dataset contains gene expression values for 198 samples in 5 breast cancer subclasses: 30 basal-like, 11 HER2, 64 luminal A, 90 luminal B and 3 normal breast-like. 3 samples with normal breast-like were removed from analysis. More information is available in [2]. This dataset can be downloaded form the Gene Expression Omnibus with accession number GSE7390.

Datasets were preprocessed before analysis. We used gene set collections from MSigDB database [18] as our predefined feature groups in CCSOGL. Not all gene sets in one collection were used in our analysis. We first applied Gene Set Enrichment Analysis (GSEA) [18] to filter out irrelevant gene sets using a cutoff of *p*-value $\geq 5\%$. Using irrelevant gene sets will introduce too much noise in our model. Further, genes in the raw data that are not present in the selected gene sets were removed from our analysis. Expression values for each gene were normalized using $x' = (x - \min(x))/(\max(x) - \min(x))$ for numerical convenience. Table 1 provides details of datasets used in our experiments.

3.2. Performance evaluation.

Performances of different sparsity-induced methods were evaluated following the conventional way of performing external cross-validation as done in the closely related works in literature, such as [13,22]. Since classes in the used datasets are unbalanced, solely estimating prediction errors favors models with better predictive performances on the dominant classes and may obscure model predictive behaviors. Hence, in addition to estimating prediction errors, macro F1-score, treating each class equally regardless of sample size, was also used. To obtain stable estimated prediction errors and macro F1-scores, we used (stratified) 4-fold cross-validation and repeated this procedure 10 times.

[1] http://www.broadinstitute.org/mpr/NCI60/.

[2] http://www.broadinstitute.org/mpr/publications/pro-jects/CNS/.

Table 2
Average of 4-fold cross validation error (CVE) for different methods over 10 runs (along with their standard deviations). The best performance is bold faced.

Dataset	Lasso	GLasso	SGL	SVM	CCSOGL
NCI60	0.448	0.398	0.396	0.461	**0.384**
	(0.042)	(0.027)	(0.030)	(0.024)	(0.031)
Brain	0.262	0.246	0.237	0.217	**0.175**
	(0.036)	(0.043)	(0.052)	(0.046)	(0.033)
Breast	0.243	0.236	0.238	0.240	**0.205**
	(0.021)	(0.019)	(0.027)	(0.018)	(0.013)

Table 3
Average of 4-fold macro F1-score for different methods over 10 runs (along with their standard deviations). Smallest standard deviation is starred and highest macro F1-score is bold faced.

Dataset	Lasso	GLasso	SGL	SVM	CCSOGL
NCI60	0.508	0.558	0.557	0.485	**0.565**
	(0.042)	(0.038)	(0.030)	(0.035)	(0.030)*
Brain	0.710	0.699	0.718	0.749	**0.782**
	(0.050)	(0.051)	(0.035)	(0.037)	(0.027)*
Breast	0.622	0.622	**0.643**	0.639	0.623
	(0.040)	(0.029)	(0.043)	(0.031)	(0.009)*

We first estimated prediction error with cross-validation error (CVE). Table 2 reports the 4-fold cross validation errors for different models. We see that the CCSOGL outperforms Lasso group Lasso (GLasso) and sparse group lasso (SGL) due to incorporation of feature group information. In the multi-cancer classification problem, it is translated into the assumption that different cancers are regulated by different genes and pathways. Incorporating the prior knowledge about feature groups and allowing class-specific dependencies of features and feature groups in building the models indeed improve the classifier performance as demonstrated by the experiments.

We also used the 4-fold macro F1-score as an evaluation metric for each method. In multi-class classification, macro F1-score [21] is a performance measurement for classifiers calculated as the average of per-class F1-scores. F1-score is the harmonic mean of *precision* and *recall* for each classification, i.e. $F_1 = 2 \times \frac{precision\times recall}{precision+recall}$, where *precision* is the ratio of samples which are classified as positive are correct and *recall* is the ratio of positive samples that are correctly classified. Here, we calculated the 4-fold macro F1-score as the average of the macro F1-scores in the 4-fold cross validation (one F1-score for each fold). Again, we reported the average of cross validation macro F1-scores over the 10 runs.

Macro F1-score weighs prediction performance for each class equally. Hence, macro F1-score will not be dominated by performances of classifiers on classes with larger sample size, yet be sensitive for performances on classes with smaller sample size. This implies that macro F1-score provides more insights for model evaluation, especially for applications in unbalanced data. As we report average of macro F1-score on multiple runs, models with less variance of macro F1-score indicate more consistency and robustness.

Table 3 presents average of 4-fold cross validation macro F1-scores (along with standard deviations) for different methods on 10 repetitions. As shown in the table, CCSOGL enjoys the smallest variance for three datasets without compromising macro F1-scores, indicating a more robust performance over the competing methods.

3.3. *Class-specific feature group selection.*

CCSOGL incorporates heterogeneous structures of feature groups across classes as described in the motivation. The incorporation brings CCSOGL an advantage over other competing method that CCSOGL is able to achieve class-specific sparsity at group level.

Table 4
Results for class-specific feature group selection: number of selected groups for each class, number of selected groups unique to each class and proportion of the selected groups among all groups.

Dataset	NCI60	Brain	Breast
# Group	{8,7,4,6,3,6,4,7}	{6,6,5,1,8}	{3,10,3,8}
# Unique	{7,3,1,4,1,4,3,4}	{2,2,3,1,3}	{2,4,0,3}
Proportion	0.35	0.16	0.37

Table 4 provides results of CCSOGL for class-wise selecting feature groups. We chose tuning parameters from 4-fold cross-validation and then trained model on the entire datasets. Selected feature groups are obtained form the trained CCSOGL model. From the table, we see that CCSOGL not only achieves group selection, but also uncovers different sparsity pattern across classes: some feature groups are identified that are unique to each class while some feature groups are shared by several classes. This flexibility in group selection for CCSOGL makes the selected model much easier to interpret.

4. Conclusion

In this paper, we have proposed a regularized method so-called CCSOGL for multinomial logistic regression and compared the performance of CCSOGL with other state-of-art sparsity-inducing methods on several benchmark datasets. The CCSOGL method can perform class-dependent selection of feature groups and features to achieve a superior performance in multinomial classification. This flexibility of CCSOGL not only increases predictive accuracy and model robustness, but also potentially discovers the functional feature groups in the classification, which may provides insight on the related field. Our model is efficiently and accurately solved by the cyclic block coordinate descent algorithm. Future development includes generalization of our methods to accommodate multiple types of features and extension to classification problems with ordinal class labels.

Acknowledgments

This work was supported by the US National Science Foundation grant NSF/CCF 1451316.

References

[1] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (Jan) (2003) 993–1022.

[2] C. Desmedt, F. Piette, S. Loi, Y. Wang, F. Lallemand, B. Haibe-Kains, G. Viale, et al., Strong time dependence of the 76-gene prognostic signature for node-negative breast cancer patients in the transbig multicenter independent validation series, Clinical Cancer Res. 13 (11) (2007) 3207–3214.

[3] D. Eddelbuettel, Seamless r and c++ Integration with rcpp, Springer, 2013.

[4] J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, J. Stat. Softw. 33 (1) (2010) 1.

[5] D.R. Hunter, K. Lange, A tutorial on mm algorithms, Am. Stat. 58 (1) (2004) 30–37.

[6] L. Jacob, G. Obozinski, J.-P. Vert, Group lasso with overlap and graph lasso, in: Proceedings of the 26th annual international conference on machine learning, ACM, 2009, pp. 433–440.

[7] P.W. Lee, Transductive hsic lasso, in: SDM, SIAM, 2014, pp. 154–162.

[8] P. Liu, H. Cheng, T.M. Roberts, J.J. Zhao, Targeting the phosphoinositide 3-kinase pathway in cancer, Nat. Rev. Drug Discovery 8 (8) (2009) 627–644.

[9] L. Meier, S. Van De Geer, P. Bühlmann, The group lasso for logistic regression, J. R. Stat. Soc. 70 (1) (2008) 53–71.

[10] Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, 87, Springer Science & Business Media, 2013.

[11] D.W. Parsons, T.-L. Wang, Y. Samuels, A. Bardelli, J.M. Cummins, L. DeLong, N. Silliman, J. Ptak, et al., Colorectal cancer: mutations in a signalling pathway, Nature 436 (7052) (2005) 792.

[12] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, L.M. Sturla, M. Angelo, M.E. McLaughlin, J.Y. Kim, L.C. Goumnerova, P.M. Black, C. Lau, et al., Prediction of central nervous system embryonal tumour outcome based on gene expression, Nature 415 (6870) (2002) 436–442.

[13] N. Rao, C. Cox, R. Nowak, T.T. Rogers, Sparse overlapping sets lasso for multitask learning and its application to fmri analysis, in: Advances in neural information processing systems, 2013, pp. 2202–2210.

[14] N. Rao, R. Nowak, C. Cox, T. Rogers, Classification with the sparse group lasso, IEEE Trans. Signal Process. 64 (2) (2016) 448–463.

[15] N. Simon, J. Friedman, T. Hastie, A blockwise descent algorithm for group-penalized multiresponse and multinomial regression, arXiv preprint arXiv:1311.6529(2013).

[16] N. Simon, J. Friedman, T. Hastie, R. Tibshirani, A sparse-group lasso, J. Comput. Graph. Stat. 22 (2) (2013) 231–245.

[17] J.E. Staunton, D.K. Slonim, H.A. Coller, P. Tamayo, M.J. Angelo, J. Park, U. Scherf, J.K. Lee, W.O. Reinhold, J.N. Weinstein, et al., Chemosensitivity prediction by transcriptional profiling, in: Proceedings of the National Academy of Sciences, 98, 2001, pp. 10787–10792.

[18] A. Subramanian, P. Tamayo, V.K. Mootha, S. Mukherjee, B.L. Ebert, M.A. Gillette, A. Paulovich, S.L. Pomeroy, et al., Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles, in: Proceedings of the National Academy of Sciences, 102, 2005, pp. 15545–15550.

[19] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. (1996) 267–288.

[20] P. Tseng, Convergence of a block coordinate descent method for nondifferentiable minimization, J. Optim. Theory Appl. 109 (3) (2001) 475–494.

[21] G. Tsoumakas, I. Katakis, I. Vlahavas, Mining Multi-label Data, in: Data mining and knowledge discovery handbook, Springer, 2009, pp. 667–685.

[22] M. Vincent, N.R. Hansen, Sparse group lasso and high dimensional multinomial classification, Comput. Stat. Data Anal. 71 (2014) 771–786.

[23] S.J. Wright, Coordinate descent algorithms, Math. Program. 151 (1) (2015) 3–34.

[24] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, J. R. Stat. Soc. 68 (1) (2006) 49–67.