Group Feature Selection with Multiclass Support Vector Machine

Fengzhen Tang[a], Lukáš Adam[b,*], Bailu Si[a]

[a] *State key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, No.114, Nanta Street, Shenyang, Liaoning Province, 110016, China.*
[b] *Southern University of Science and Technology, 1088 Xueyuan Ave, Nanshan Qu, Shenzhen, Guangdong Province, 518055, China.*

Abstract

Feature reduction is nowadays an important topic in machine learning as it reduces the complexity of the final model and makes it easier to interpret. In some applications, the features arise from multiple sources and it is not so important to select the individual features as to select the important sources. This leads to a group feature selection problem. In this paper, we consider the group feature selection in the multiclass classification setting based on the framework of support vector machines. We reformulate it as a sparse problem by prescribing the maximum number of active groups and propose a novel method based on the ADMM algorithm. We proposed the method in such a way that the main computational load is performed in the first iteration and the remaining iterations can be computed fast. This allows us to handle large problems. We demonstrate the good performance of our method on several real-world datasets.

Keywords: Group feature selection, Support vector machine, Multiclass support vector machine, Alternating direction method of multipliers, EEG channel selection

1. Introduction

Feature selection is an important procedure in many machine learning applications such as text classification or DNA analysis. It aims at selecting a small number of features which contain no irrelevant or redundant features. Besides identifying the important features, it helps to reduce the computational load and may improve the classification performance. In this work, we focus on supervised feature selection. They can be roughly grouped into three categories: Filter, wrapper, and embedded methods [13]. Filter methods evaluate the relevance of features via univariate statistics. The wrapper approach repeatedly uses a classifier to search for relevant features. Embedded methods perform variable selection as part of the learning procedure. Since filter methods usually evaluate all features independently they perform worse than wrapper or

*Corresponding author
Email addresses: tangfengzhen@sia.cn (Fengzhen Tang), adam@utia.cas.cz (Lukáš Adam), sibailu@sia.ac.cn (Bailu Si)

embedded methods. Since embedded methods are more computationally efficient than wrapper methods while maintaining comparable selection results, we focus mainly on them in this paper.

In many applications, data are obtained from multiple sources and each source produces several features [29]. For example, in the EEG (Electroencephalography) signal classification [20], signals are obtained by attaching multiple electrodes to a person's head. The signal emitted from each electrode is then represented via several coefficients. Thus the input features possess group structure where the coefficients corresponding to one electrode form one group. In this case, feature selection imposed on individual features may not reveal this structure information. Hence, instead of finding important individual features, finding important feature groups is more suitable in this scenario. This leads to the problem of group feature selection.

So far, several research works related to group feature selection have been presented, such as group Lasso [38, 27, 22], sparse group Lasso [28], and Bayesian Group Lasso [25]. However, these group feature selection methods were mainly based on square loss and logistic loss for regression and classification analysis. There does exist one work exploring hinge loss popularized by Support Vector Machines (SVM) [12]. But this work only targets at regression and binary classification, leaving multiclass support vector machine unexplored. Even though, group feature selection for multiclass classification problem can be simply decoupled to group feature selection for several binary classification problems [12] via one-against-rest or one-against-one strategies. However, this way will not be able to identify relevant feature groups that simultaneously works well for all classes. There do exist several works related to this simultaneous multiclass feature selection [11]. But these works did not consider the group feature selection.

There are many methods for feature selection, see the excellent reviews [14, 16, 31]. In this manuscript, we will concentrate on feature selection via the powerful classification algorithm multiclass support vector machines [32]. In [31] the authors introduced sparsity regularization in the linear dynamic analysis. The recursive feature elimination, which starts with the whole set of features and removes one feature at every iteration, was extended for multiclass support vector machines in [39, 11]. A framework of scaling factors is also introduced for multiclass support vector machines to perform feature selection across multiclass [11, 36]. Paper [30] extends variational relevant vector machine [5] to group feature selection.

Many methods are based on the (group) Lasso regularization. Outside of the SVM context, they have been employed in [27] for generalized linear models, in [22] for logistic regression models, in [18] for overlapping groups, or in [23] to automatically select salient nodes in deep neural networks. In the SVM context, the group Lasso is either applied directly as in [36] or it is argued that the group Lasso is a convex approximation of the group zero norm [9]. Some methods attack this group zero norm [21] or necessary optimality conditions are used to solve the problem [1, 24].

In this paper, we propose a novel sparse group feature selection method for multiclass support vector machine (MSVM). Our method can choose an optimal subset of features in a grouped manner simultaneously working well for all classes. We mention the generalization of the recursive feature elimination to select groups instead of individual features. The main result is to use the all-together approach for MSVM of [35], consider the group zero norm and solve it via the ADMM method [7].

The group Lasso and group zero norm terms are usually placed in the objective. Since this means one additional hyperparameter, we place it into the constraints. We derive a special decomposition for the ADMM method such that most of the work is done in the first iteration and the remaining iterations are relatively cheap. The ADMM method is known to quickly provide a reasonable solution estimate but the convergence to optimality may be slow. Since we are interested only in determining relevant feature groups, we stop ADMM once the features groups are stabilized. The actual model coefficient can then be computed by restricting the original features to the selected features by any classification technique. We show good performance of our method on several real-world datasets.

This paper is organized as follows: Section 2 introduces support vector machines and multiclass group feature selection. In Section 3 we first generalize the recursive feature elimination to handle groups. Then we formulate the MSVM for feature selection and propose how to solve it via the ADMM method. We comment on the computational complexity and provide a comparison with the group Lasso method. Finally, in Section 4 we show the good performance of our method on several real-world datasets.

2. Sparsity Inducing Terms

In this section, we provide a brief introduction to support vector machines, group feature selection, and multiclass feature selection methods. In the last part, we then combine all these part to derive an optimization problem for multiclass group feature selection based on support vector machines.

In the whole manuscript, we assume to have n pairs of training data $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the feature vector corresponding to i-th sample, and y_i is the class label of $\boldsymbol{x}_i$. Unless stated otherwise, we assume that there are K classes with labels $1, \ldots, K$.

2.1. Support Vector Machines

Support vector machines (SVMs) are powerful supervised algorithms for classification. Originally designed for binary classification, their decision boundary is represented by a linear function $\boldsymbol{w}^\top \boldsymbol{x} + b$, where $\boldsymbol{w} \in \mathbb{R}^d$ is the weight vector and $b \in \mathbb{R}$ is the shift of the separating hyperplane. When the labels are $+1$ or -1, the idea of SVMs is to maximize the margin between the samples and the separating hyperplane by solving

$$\begin{aligned} \underset{w,b,\xi}{\text{minimize}}\ & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^n \xi_i \\ \text{subject to}\ & y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0,\ i = 1, \ldots, n, \end{aligned} \tag{1}$$

Here, the first term of the objective is the regularization, the second term measures the classification error and $C > 0$ specifies the trade-off between them. For a general $\boldsymbol{x}$, the decision rule is based on the sign of $\boldsymbol{w}^\top \boldsymbol{x} + b$.

Many real-world classification tasks involve multiclass classification. There are two popular approaches. In the one-vs-one and one-vs-rest approaches, the multiclass classification problem is divided several times into binary classification problems and

multiple models are built. The class with most "wins" is selected as the predicted class. However, since each model considers a different coefficient vector $\boldsymbol{w}$, this approach is not suitable for feature selection. The authors of [35] employed the all-together approach and assigned a separating hyperplane $\boldsymbol{w}_k^\top \boldsymbol{x} + b_k$ to every class k. The goal is then again to maximize the margin, which yields the following problem

$$\begin{aligned} \underset{w,b,\xi}{\text{minimize}}\ & \frac{1}{2}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|_2^2 + C\sum_{i=1}^{n}\sum_{k\neq y_i}\xi_{ik} \\ \text{subject to}\ & (\boldsymbol{w}_{y_i}-\boldsymbol{w}_k)^\top \boldsymbol{x}_i + b_{y_i} - b_k \geq 1-\xi_{ik}, \\ & \xi_{ik}\geq 0,\ i=1,\ldots,n,\ k\neq y_i. \end{aligned} \tag{2}$$

Similarly as for binary SVMs (1), to a general $\boldsymbol{x}$ a class with the highest value of $\boldsymbol{w}_k^\top \boldsymbol{x} + b_k$ is assigned. For a detailed review of both binary and multiclass SVMs we refer to [8, 32].

2.2. Group Feature Selection via Group Lasso

In many applications, there are J pairwise disjoint feature groups $G_1,\ldots,G_J$ and the task is to select a small number of the relevant feature groups instead of a small number of the relevant features. Many embedded group feature selection methods are based on the group Lasso defined by

$$\|\boldsymbol{w}\|_{2,1}^{\text{group}} := \sum_{j=1}^{J}\|\boldsymbol{w}_{G_j}\|_2 = \sum_{j=1}^{J}\sqrt{\sum_{f\in G_j} w_f^2}, \tag{3}$$

where $\boldsymbol{w}_{G_j}$ is the restriction of $\boldsymbol{w}$ to group G_j. The group sparsity inducing term $\|\boldsymbol{w}\|_{2,1}^{\text{group}}$ is usually added to the objective.

2.3. Multiclass Feature Selection via Group Lasso

While in the previous section we considered two classes with multiple feature groups, here we consider K classes without any group structure. Similarly as before, it is possible to use the group Lasso to select features

$$\|\boldsymbol{w}\|_{2,1}^{\text{mclass}} := \sum_{j=1}^{d}\|\boldsymbol{w}_{\cdot,j}\|_2 = \sum_{j=1}^{d}\sqrt{\sum_{k=1}^{K} w_{k,j}^2}. \tag{4}$$

By $w_{k,j}$ we understand the j-th component of $\boldsymbol{w}_k$ and by $\boldsymbol{w}_{\cdot,j}$ the K-dimensional vector composed by $w_{k,j}$ for all k. Paper [21] argued that $\|\boldsymbol{w}\|_{2,1}^{\text{mclass}}$ is just a convex approximation of the group zero norm

$$\|\boldsymbol{w}\|_{0,1}^{\text{mclass}} := \sum_{j=1}^{d}\|\boldsymbol{w}_{\cdot,j}\|_0 = \#\{j|\ w_{k,j}\neq 0 \text{ for some } k\}. \tag{5}$$

They further observed that this sparse inducing term can be written as a difference of convex (DC) functions and used a DC algorithm to solve the resulting problem. In [36] the authors used a clever technique combining both of the approaches above: They considered $\|\boldsymbol{w}\|_{p,1}^{\text{mclass}}$ for a general $p\in[0,1]$ and updated p automatically.

2.4. Multiclass Group Feature Selection via Group Lasso

To get a group multiclass sparsity induding term, we combine the group sparsity inducing term $\|\boldsymbol{w}\|_{2,1}^{\text{group}}$ from (3) with the multiclass sparsity inducing term $\|\boldsymbol{w}\|_{2,1}^{\text{mclass}}$ from (4) to obtain

$$\|\boldsymbol{w}\|_{2,1} := \sum_{j=1}^{J} \|\boldsymbol{w}_{\cdot,G_j}\|_2 = \sum_{j=1}^{d} \sqrt{\sum_{k=1}^{K} \sum_{f \in G_j} w_{k,f}^2}. \tag{6}$$

Building on the notation above, by $\boldsymbol{w}_{\cdot,G_j}$ we understand the union of $\boldsymbol{w}_{\cdot,f}$ for all $f \in G_j$, thus all coefficients of the weight vector corresponding to the given group. Theoretically, we could add (6) directly to the multiclass SVM problem (2) but this would mean one more hyperparameter which could be difficult to tune [2]. Instead we decided to work with the generalization of $\|\boldsymbol{w}\|_{0,1}^{\text{mclass}}$ from (5) to define the following multiclass group zero norm

$$\|\boldsymbol{w}\|_{0,1} := \sum_{j=1}^{J} \|\boldsymbol{w}_{\cdot,G_j}\|_0 = \#\{j|\ w_{k,f} \neq 0 \text{ for some } k \text{ and some } f \in G_j\}. \tag{7}$$

Then we modify problem (2) by prescribing the maximum number of important groups $s_{\max}$:

$$\begin{aligned} \underset{w,b,\xi}{\text{minimize}}\ & \frac{1}{2}\sum_{k=1}^{K} \|\boldsymbol{w}_k\|_2^2 + C\sum_{i=1}^{n}\sum_{k\neq y_i} \xi_{ik} \\ \text{subject to}\ & (\boldsymbol{w}_{y_i} - \boldsymbol{w}_k)^\top x_i + b_{y_i} - b_k \geq 1 - \xi_{ik}, \\ & \xi_{ik} \geq 0,\ i = 1,\ldots,n,\ k \neq y_i, \\ & \|\boldsymbol{w}\|_{0,1} \leq s_{\max}. \end{aligned} \tag{8}$$

3. Simultanous Multiclass Group Feature Selection

In this section, we present two methods for multiclass group feature selection. The first method is a simple adaptation of the recursive feature elimination. The second method solves directly (8) via the ADMM method. For the second method, we then provide a basic analysis of computational complexity and compare with the "standard" group Lasso.

3.1. Multiclass Recursive Group Feature Elimination

First, we extend the multiclass recursive feature elimination (RFE) method to group feature selection. The algorithm starts with an active list of all features and removes one group after another. At every iteration, the weights $\boldsymbol{w}_k$ are computed with zeros at the inactive features by any all-together classification method. Then for every active group j, the score

$$s_j := \frac{1}{|G_j|}\|\boldsymbol{w}_{\cdot,G_j}\|_2^2 = \frac{1}{|G_j|}\sum_{k=1}^{K}\sum_{f\in G_j} w_{k,f}^2$$

is computed and the group with the smallest score is made inactive.

3.2. Solving (8) *with ADMM*

In this section, we apply the alternating direction method of multipliers (ADMM) to (8). The ADMM [7] is a popular optimization method which has been already used to solve classification problems in SVMs [4, 37]. However, to the best of our knowledge, it has never been used for multiclass group selection. The ADMM is a dual ascent method, where the gradient of the dual objective is computed only approximately. Since it is a dual method, usually a convexity is required for convergence proofs. We make use of the recent results of [15] where a convergence of ADMM was shown also for some non-convex problems with sparsity constraints.

To simplify the notation, we first write (8) in a compact form. Recall that K is the number of classes, d the number of features and n the number of samples. We define $\boldsymbol{w} := [\boldsymbol{w}_1; \ldots; \boldsymbol{w}_K] \in \mathbb{R}^{Kd}$ and $\boldsymbol{b} := [b_1; \ldots; b_K] \in \mathbb{R}^K$ and collect ξ_{ik} into one vector $\boldsymbol{\xi} \in \mathbb{R}^{(K-1)n}$. Finally we collect the first constraint in (8) into a matrix $A_w \in \mathbb{R}^{(K-1)n \times Kd}$ containing in every row only $\boldsymbol{x}_i^\top$ a $-\boldsymbol{x}_i^\top$ and a matrix $A_b \in \mathbb{R}^{(K-1)n \times K}$ containing in every row only 1 a -1. Then we may rewrite (8) into

$$\begin{aligned} \underset{w,b,\xi}{\text{minimize}}\ & \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C1^\top \boldsymbol{\xi} \\ \text{subject to}\ & A_w \boldsymbol{w} + A_b \boldsymbol{b} \geq 1 - \boldsymbol{\xi}, \\ & \boldsymbol{\xi} \geq 0, \\ & \|\boldsymbol{w}\|_{0,1} \leq s_{\max}. \end{aligned} \tag{9}$$

By combining both constraints on $\boldsymbol{\xi}$ we obtain $\boldsymbol{\xi} \geq \max\{1 - A_w \boldsymbol{w} - A_b \boldsymbol{b}, 0\}$. This results in (8)

$$\begin{aligned} \underset{w,b}{\text{minimize}}\ & \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C1^\top \max\{1 - A_w \boldsymbol{w} - A_b \boldsymbol{b}, 0\} \\ \text{subject to}\ & \|\boldsymbol{w}\|_{0,1} \leq s_{\max}. \end{aligned} \tag{10}$$

To apply ADMM to (10), we need to make this problem separable with only linear constraints. For the separability, we introduce artificial variables $\boldsymbol{y} = 1 - A_w \boldsymbol{w} - A_b \boldsymbol{b}$ and $\boldsymbol{z} = \boldsymbol{w}$ while for the linear constraints, we use the standard argument of enforcing the constraint in objective by using the indicator function

$$I(\|\boldsymbol{z}\|_{0,1} \leq s_{\max}) = \begin{cases} 0 & \text{if } \|\boldsymbol{z}\|_{0,1} \leq s_{\max}, \\ +\infty & \text{otherwise.} \end{cases}$$

Then problem (10) reads

$$\begin{aligned} \underset{w,b,y,z}{\text{minimize}}\ & \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C1^\top \max\{\boldsymbol{y}, 0\} + I(\|\boldsymbol{z}\|_{0,1} \leq s_{\max}) \\ \text{subject to}\ & \boldsymbol{y} = 1 - A_w \boldsymbol{w} - A_b \boldsymbol{b}, \\ & \boldsymbol{z} = \boldsymbol{w}. \end{aligned} \tag{11}$$

Now we introduce the scaled version of the augmented Lagrangian [7, Section 3.1.1] by

$$\begin{aligned} L(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{y}, \boldsymbol{z}; \boldsymbol{\lambda}, \boldsymbol{\mu}) := & \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C1^\top \max\{\boldsymbol{y}, 0\} + I(\|\boldsymbol{z}\|_{0,1} \leq s_{\max}) \\ & + \frac{\rho}{2}\|\boldsymbol{y} - 1 + A_w \boldsymbol{w} + A_b \boldsymbol{b} + \boldsymbol{\lambda}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{z} - \boldsymbol{w} + \boldsymbol{\mu}\|_2^2. \end{aligned} \tag{12}$$

Here $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are the multipliers associated with the first and second constraint in (11), respectively and $\rho > 0$ is an arbitrary parameter.

ADMM is an iterative algorithm. In iteration l, the current iterate $(\boldsymbol{w}^l, \boldsymbol{b}^l, \boldsymbol{y}^l, \boldsymbol{z}^l, \boldsymbol{\lambda}^l, \boldsymbol{\mu}^l)$ is updated in the following four steps

$$(\boldsymbol{w}^{l+1}, \boldsymbol{b}^{l+1}, \boldsymbol{y}^{l+1}) = \operatorname*{argmin}_{w,b,y} L(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{y}, \boldsymbol{z}^l; \boldsymbol{\lambda}^l, \boldsymbol{\mu}^l), \tag{13a}$$

$$\boldsymbol{z}^{l+1} = \operatorname*{argmin}_{z} L(\boldsymbol{w}^{l+1}, \boldsymbol{b}^{l+1}, \boldsymbol{y}^{l+1}, z; \boldsymbol{\lambda}^l, \boldsymbol{\mu}^l), \tag{13b}$$

$$\boldsymbol{\lambda}^{l+1} = \boldsymbol{\lambda}^l + \boldsymbol{y}^{l+1} - 1 + A_w \boldsymbol{w}^{l+1} + A_b \boldsymbol{b}^{l+1}, \tag{13c}$$

$$\boldsymbol{\mu}^{l+1} = \boldsymbol{\mu}^l + \boldsymbol{z}^{l+1} - \boldsymbol{w}^{l+1}. \tag{13d}$$

Even though (13a) is a quadratic programming problem, it does not have a closed-form solution. Since ADMM does not provide an exact gradient for the dual ascent but only its approximation, we further approximate (13a) by solving first for $(\boldsymbol{w}^{l+1}, \boldsymbol{b}^{l+1})$ and only then for $\boldsymbol{y}^{l+1}$. When solving (13a) with respect to $(\boldsymbol{w}, \boldsymbol{b})$, we can omit two terms of the Lagrangian L which do not depend on $\boldsymbol{w}$ and $\boldsymbol{b}$. Then we get

$$\operatorname*{minimize}_{w,b} \ \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \frac{\rho}{2}\|\boldsymbol{y}^l - 1 + A_w \boldsymbol{w} + A_b \boldsymbol{b} + \boldsymbol{\lambda}^l\|_2^2 + \frac{\rho}{2}\|\boldsymbol{z}^l - \boldsymbol{w} + \boldsymbol{\mu}^l\|_2^2.$$

Since this is a quadratic unconstrained problem, it is equivalent to setting its derivatives with respect to w and b to zero. This yields

$$\begin{aligned} \boldsymbol{w} + \rho A_w^\top (\boldsymbol{y}^l - 1 + A_w \boldsymbol{w} + A_b \boldsymbol{b} + \boldsymbol{\lambda}^l) - \rho(\boldsymbol{z}^l - \boldsymbol{w} + \boldsymbol{\mu}^l) = 0, \\ \rho A_b^\top (\boldsymbol{y}^l - 1 + A_w \boldsymbol{w} + A_b \boldsymbol{b} + \boldsymbol{\lambda}^l) = 0. \end{aligned} \tag{14}$$

This is a system of linear equations. We comment on a precise way of solving it later in Section 3.3. The second part of solving (13a) amounts to minimizing it with respect to $\boldsymbol{y}$. Ignoring again the constant terms, this is equivalent to

$$\operatorname*{minimize}_{y} \ C1^\top \max\{\boldsymbol{y}, 0\} + \frac{\rho}{2}\|\boldsymbol{y} - 1 + A_w \boldsymbol{w}^{l+1} + A_b \boldsymbol{b}^{l+1} + \boldsymbol{\lambda}^l\|_2^2. \tag{15}$$

This problem can be decomposed into multiple problems in one real variable and analysis of all possible cases shows that the solution takes the form

$$\boldsymbol{y}^{l+1} = T_{\text{soft}}(1 - A_w \boldsymbol{w}^{l+1} - A_b \boldsymbol{b}^{l+1} - \boldsymbol{\lambda}^l, \rho^{-1} C1), \tag{16}$$

where $T_{\text{soft}}(\boldsymbol{t}_1, \boldsymbol{t}_2)$ is the shifted soft-thresholding operator [7, Section 4.4.3]

$$T_{\text{soft}}(\boldsymbol{t}_1, \boldsymbol{t}_2) = \max\{\min\{\boldsymbol{t}_1, 0\}, \boldsymbol{t}_1 - \boldsymbol{t}_2\}.$$

Since only two terms in L depend on z, solving (13b) amounts to

$$\begin{aligned} &\operatorname*{minimize}_{z} \ \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{w}^{l+1} + \boldsymbol{\mu}^l\|_2^2 \\ &\text{subject to } \|\boldsymbol{z}\|_{0,1} \leq s_{\max}. \end{aligned}$$

This problem has a closed-form solution. For each $j = 1, \ldots, J$ define the score s_j corresponding to the feature group G_j by

$$s_j := \|\boldsymbol{w}^{l+1}_{\cdot,G_j} - \boldsymbol{\mu}^{l}_{\cdot,G_j}\|_2^2 = \sum_{k=1}^{K} \sum_{f \in G_j} (w^{l+1}_{k,f} - \mu^{l}_{k,f})^2. \tag{17}$$

Then the solution of (13b) componentwise reads

$$z^{l+1}_{k,f} = \begin{cases} w^{l+1}_{k,f} - \mu^{l}_{k,f} & \text{if } s_j \text{ is among } s_{\max} \text{ largest score values and } f \in G_j; \\ & \text{otherwise.} \end{cases} \tag{18}$$

The multiplier updates (13c) and (13d) are trivial.

3.3. Numerical Solution Procedure

The computationally most demanding part from the previous section is solving system (14). After some linear algebra and scaling by ρ^{-1}, it can be written in the form

$$\Big[\underbrace{(1+\rho^{-1})\begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}}_{Q_1} + \underbrace{\begin{pmatrix} A_w^\top \\ A_b^\top \end{pmatrix}}_{Q_2^\top} \begin{pmatrix} A_w & A_b \end{pmatrix}\Big] \begin{pmatrix} \boldsymbol{w}^{l+1} \\ \boldsymbol{b}^{l+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{z}^l + \boldsymbol{\mu}^l - A_w^\top(\boldsymbol{y}^l - 1 + \boldsymbol{\lambda}^l) \\ -A_b^\top(\boldsymbol{y}^l - 1 + \boldsymbol{\lambda}^l) \end{pmatrix}. \tag{19}$$

In every iteration, we need to invert $Q_1 + Q_2^\top Q_2$. Note that it is a fixed matrix with dimension $K(d+1) \times K(d+1)$. Moreover, it is a positive semidefinite matrix and can be made positive definite matrix by either adding a small multiple of the identity matrix to Q_1 or by considering $\boldsymbol{b}$ as one feature group, which would change Q_1 into the identity matrix. Now there are two possibilities.

If the number of features d is smaller or comparable to the number of samples n, we can compute the Cholesky decomposition of $Q_1 + Q_2^\top Q_2$, thus to find a regular lower triangular matrix B_1 with

$$B_1 B_1^\top = (1+\rho^{-1})\begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} A_w^\top \\ A_b^\top \end{pmatrix}\begin{pmatrix} A_w & A_b \end{pmatrix}. \tag{20}$$

Then (19) reads

$$\begin{pmatrix} \boldsymbol{w}^{l+1} \\ \boldsymbol{b}^{l+1} \end{pmatrix} = (B_1^{-1})^\top B_1^{-1} \begin{pmatrix} \boldsymbol{z}^l + \boldsymbol{\mu}^l - A_w^\top(\boldsymbol{y}^l - 1 + \boldsymbol{\lambda}^l) \\ -A_b^\top(\boldsymbol{y}^l - 1 + \boldsymbol{\lambda}^l) \end{pmatrix}. \tag{21}$$

Since B_1 is a lower triangular matrix, system (21) is easy to solve.

If the number of features d is large, the computation from the previous paragraph is infeasible, and it may be advantageous to use the Woodbury matrix identity to obtain

$$(Q_1 + Q_2^\top Q_2)^{-1} = Q_1^{-1} - Q_1^{-1} Q_2^\top (I + Q_2 Q_1^{-1} Q_2^\top)^{-1} Q_2 Q_1^{-1}. \tag{22}$$

Note that Q_1 can be made a diagonal matrix with positive entries by one of the two possibilities mentioned above, and thus Q_1^{-1} is simple to compute. Moreover, the

positive definite matrix $I + Q_2 Q_1^{-1} Q_2^\top$ has dimension $(K-1)n \times (K-1)n$ and if there is a reasonable number of samples n, the Cholesky decomposition can be again computed to get a lower triangular matrix B_2 with

$$B_2 B_2^\top = I + Q_2 Q_1^{-1} Q_2^\top. \tag{23}$$

Then (19) is due to (22) and (23) equivalent to

$$\begin{pmatrix} \boldsymbol{w}^{l+1} \\ \boldsymbol{b}^{l+1} \end{pmatrix} = \left(Q_1^{-1} - Q_1^{-1} Q_2^\top (B_2^{-1})^\top B_2^{-1} Q_2 Q_1^{-1} \right) \begin{pmatrix} \boldsymbol{z}^l + \boldsymbol{\mu}^l - A_w^\top (\boldsymbol{y}^l - 1 + \boldsymbol{\lambda}^l) \\ -A_b^\top (\boldsymbol{y}^l - 1 + \boldsymbol{\lambda}^l) \end{pmatrix}. \tag{24}$$

The great advantage of these approaches is that it suffices to compute one Cholesky decomposition and then every iteration of ADMM is cheap.

We summarize this whole procedure in Algorithm 3.1. The usual termination criterion is based on a change in primal and dual variables. However, since we are interested in determining the important feature groups, we stop ADMM when the feature groups stabilize and then run the multiclass SVM classification on the selected features.

Algorithm 3.1 For solving (8)

Compute B_1 or B_2 from the Cholesky decomposition in (20) or (23)
Set initial data $(\boldsymbol{y}^0, \boldsymbol{z}^0, \boldsymbol{\lambda}^0, \boldsymbol{\mu}^0)$
for $l = 1, \ldots,$ maxIter **do**
    Update $(\boldsymbol{w}^{l+1}, \boldsymbol{b}^{l+1})$ via (21) or (24)
    Update $\boldsymbol{y}^{l+1}$ via (16)
    Update $\boldsymbol{z}^{l+1}$ via (18)
    Update $\boldsymbol{\lambda}^{l+1}$ via (13c) and $\boldsymbol{\mu}^{l+1}$ via (13d), respectively
    if $s_{\max}$ largest scores from (17) did not change **then**
        break
    end if
end for
Obtain a classifier on the selected features

3.4. Connection to the Group Lasso

The usual approach is to consider the convex term $\|\boldsymbol{w}\|_{2,1}$ in the objective instead of the discontinuous term $\|\boldsymbol{w}\|_{0,1}$ in the constraints as in our problem (8). This results in

$$\begin{aligned} \text{minimize } & \frac{1}{2} \sum_{k=1}^{K} \|\boldsymbol{w}_k\|_2^2 + C \sum_{i=1}^{n} \sum_{k \neq y_i} \xi_{ik} + \hat{C} \|\boldsymbol{w}\|_{2,1} \\ \text{subject to } & (\boldsymbol{w}_{y_i} - \boldsymbol{w}_k)^\top x_i + b_{y_i} - b_k \geq 1 - \xi_{ik}, \\ & \xi_{ik} \geq 0,\ i = 1, \ldots, n,\ k \neq y_i. \end{aligned} \tag{25}$$

Even though (8) and (25) look similar, they are some key differences between them. First, (25) is a convex problem while (8) is a non-convex problem. Second, (25) is simpler to solve while (8) is closer to the group feature selection problem. Third, problem (25)

contains hyperparameters C and $\hat{C}$ while problem (8) contains hyperparameters C and $s_{\max}$. The second set of hyperparameters should be easier to tune as $s_{\max}$ has, on the contrary to $\hat{C}$, a direct interpretation as the maximal number of selected groups.

Since (25) is a convex quadratic program, if we apply ADMM to it, we have guaranteed convergence. The ADMM would result in the same procedure as Algorithm 3.1 and the only difference is the $\boldsymbol{z}^{l+1}$ update (18) which would take the form

$$z_{k,f}^{l+1} = \begin{cases} w_{k,f}^{l+1} - \mu_{k,f}^{l} & \text{if } s_j \geq \frac{2}{\rho}\hat{C}, \text{ where } j \text{ is unique group index with } f \in G_j, \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

In other words, while our method in (18) always sets $s_{\max}$ features to be non-zero, (26) updates $\boldsymbol{z}$ in the identical way but the number of non-zero features changes in every iteration. Since the latter algorithm has guaranteed convergence and Algorithm 3.1 is very similar to it, this could explain the good convergence of the ADMM method as observed in the numerical section.

4. Numerical Experiments and Discussion

In this section, we show the good numerical performance of our methods. We consider the *RFE* method proposed in Section 3.1 and the *ADMM* method from Section 3.2. While the *RFE* method starts with the full set of groups and removes one group after another, the *ADMM* method fixes the maximal number of groups $s_{\max}$ and finds $s_{\max}$ feature groups. Since this makes the *RFE* method unsuitable for problems with a large number of feature groups, we concentrate mainly on the *ADMM* method.

We consider three classes of real-world datasets. All three classes follow a different goal. The first category (datasets GSA [33, 26], USPS [17] and Smartphones [3]) shows that our methods outperform other group feature selection methods. The second category (datasets RNA-Seq 1 and RNA-Seq 2 [34]) shows that even for datasets with a large number of features, the *ADMM* method selects group features which achieve high accuracy. The third category (dataset K3B [6]) is a neuroscience application where the relevant features are known. We show that our methods can select these features. The datasets are summarized in Table 1 and can be found online.[1]

All data were normalized and each dataset was randomly divided into the training and testing sets 100 times. The depicted results are averages over all these trials. Concerning hyperparameters, we chose $\rho = 1000$ for all experiments. For categories 1 and 3, we determined C by 5-fold cross-validation on the whole dataset and considered all possible values of the maximal number of groups $s_{\max}$. For category 2, we fixed $C = 1$ and chose $s_{\max}$ from $\{10, 20, \ldots, 100\}$. After the *ADMM* method selects the relevant features, any linear model can be used to determine the weights $\boldsymbol{w}$. We used LIBSVM [10].

[1] GSA, Smartphones and RNA-Seq 1 are from the UCI depository, USPS can be downloaded from https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/multiclass.html, RNA-Seq 2 is a manual modification of RNA-Seq 1 and K3B is from http://www.bbci.de/competition/iii/.

Table 1: Dataset description.

Dataset	Classes	Samples	Samples train	Features	Groups
GSA	6	13910	300/600/1200	128	16
USPS	10	9298	500/1000/1500	256	16
Smartphones	6	10299	300/600/1200	561	18
RNA-Seq 1	5	801	100	20531	20531
RNA-Seq 2	5	801	100	20530	4106
K3B	4	180	126	180	60

4.1. Category 1: Comparison with Existing Methods

We followed the setting from [12] and considered three datasets. The Gas Sensor Array (GSA) dataset contains information from 16 chemical sensors exposed to 6 gases at different concentration levels. Each sensor provided 8 features, which resulted in 16 groups and 128 features. The goal is to discriminate the six different gases. The Smartphones dataset is built from recordings of 17 signals of 30 subjects performing six activities (walking, walking upstairs, walking downstairs, sitting, standing, and lying) while wearing a smartphone. Features such as mean, correlation, or autoregressive coefficients were subsequently extracted from these 17 signals. Besides, one additional group of features was obtained by averaging the signals in a signal window sample. This resulted in 18 feature groups with a different number of features in each group. The USPS dataset contains handwritten digits, each represented by a 16×16 matrix. Following [12], each column of this representation is regarded as one group.

Since there are not many multiclass group feature selection methods, and especially not in the SVM context, we compare the performance with the Bayesian Group Feature Selection for Support Vector Machines (BGFS-SVM) method [12], where the feature selection in the multiclass setting is tackled via decoupling it into several binary group feature selection problems via the one-vs-rest strategy. This paper also implemented the group Lasso (G-Lasso) [38] and the sparse group Lasso (SG-Lasso) [28]. As in [12], we randomly select for each dataset training instances from each class with the size of $\{50, 100, 200\}$ and the rest instances are used as the test set.

The average prediction accuracies of our *ADMM* and *RFE* methods are given in Figure 1. The performance of both methods is fairly similar: When averaged over all values of $s_{\max}$, the *ADMM* was better four times while the *RFE* gave a better performance five times. Moreover, especially for the GSA and Smartphones datasets, the performance did not decline when more than half of the groups were omitted.

In [12], the authors showed the performance of the BGFS-SVM method for 7, 11 and 8 maximal groups for the datasets GSA, USPS and Smartphones, respectively. In Table 2, we depict the performance of the *ADMM* method for these values. Following the name BGFS-SVM, we denote our method GFS-MSVM (Group Feature Selection for Multiclass Support Vector Machines). In 7 out of 9 cases, our method GFS-MSVM outperformed the results presented in [12], in the remaining two cases, it was only slightly worse.

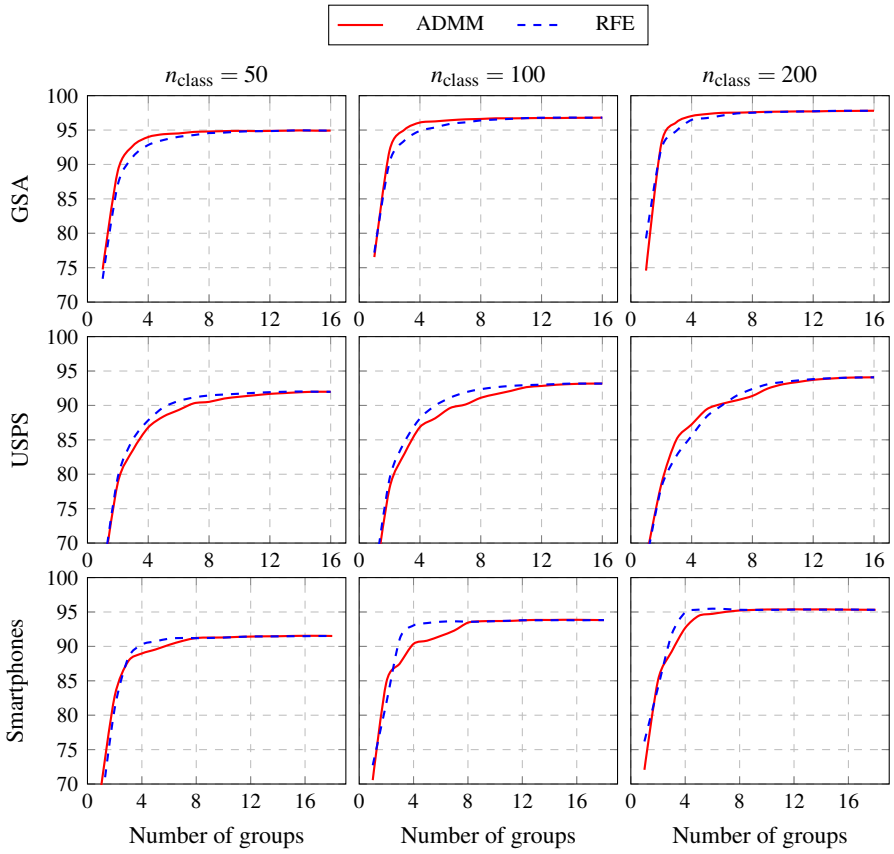

Figure 1: The average prediction accuracy[%] of our methods on the datasets from category 1.

4.2. Category 2: Datasets with a Large Number of Features

In the RNA Seq dataset, for 801 patients with five different kinds of tumors (BRCA, KIRC, COAD, LUAD and PRAD) the expression levels of 20531 genes were measured by the illumina HiSeq platform. On this dataset, we will show that the *ADMM* method is able to reduce this large number of features into a small one while keeping a large prediction accuracy for the tumor type. As in the previous category, we selected 20 samples for each class as the training samples, which resulted in 100 training samples. Since there is no information about the genes, we considered two cases: When each group consists of one gene (denoted RNA Seq 1) and when every five genes were grouped into one group (denoted RNA Seq 2). For the latter dataset, we adjusted the maximum number of feature groups $s_{\max}$ such that the number of selected features were the same for both datasets.

The average accuracy can be seen in the left part of Figure 2. The performance on both datasets is very high. Theoretically, the performance on RNA Seq 1 should always be superior as there are more possibilities how to choose the given number of features.

Table 2: Average prediction acuracy[%] on the benchmark datasets from category 1. Results for G-Lasso, SG-Lasso and BGFS-SVM are taken from [12]. Our method is denoted GFS-MSVM.

Dataset	n_{class}	G-Lasso	SG-Lasso	BGFS-SVM	GFS-MSVM
GSA	50	79.8±4.4	83.0±4.8	84.3±1.7	**94.7±1.1**
	100	84.3±3.6	85.3±4.0	92.0±1.6	**96.6±0.6**
	200	86.4±3.5	88.3±2.7	**98.1±0.8**	97.5±0.4
USPS	50	69.0±1.3	71.5±1.5	86.6±1.2	**91.5±0.7**
	100	69.6±1.4	72.2±0.9	90.7±0.4	**92.7±0.6**
	200	72.7±1.2	74.1±1.2	92.5±0.3	**93.4±0.9**
Smartphones	50	57.5±3.5	58.2±3.5	82.1±1.6	**91.2±0.8**
	100	71.3±4.1	72.3±2.8	92.6±0.8	**93.5±1.0**
	200	73.3±4.1	74.1±2.8	**95.6±0.3**	95.2±0.4

However, for a larger number of $s_{\max}$, the performance was better on the RNA Seq 2 dataset. This was likely caused by the fact that the grouping of genes decreased the size of the search space for the second dataset.

In the right part of Figure 2 we show the experimental convergence of the *ADMM* method on RNA Seq 1. The x-axis denotes the genes while the y-axis denotes the iteration of the *ADMM* method until it converges in iteration 885. The black vertical lines show in which iterations a given gene was selected in the $s_{\max} = 30$ most important genes. We can see that the convergence displays the beneficial property of stability of selected genes; some genes were even selected during all iterations.

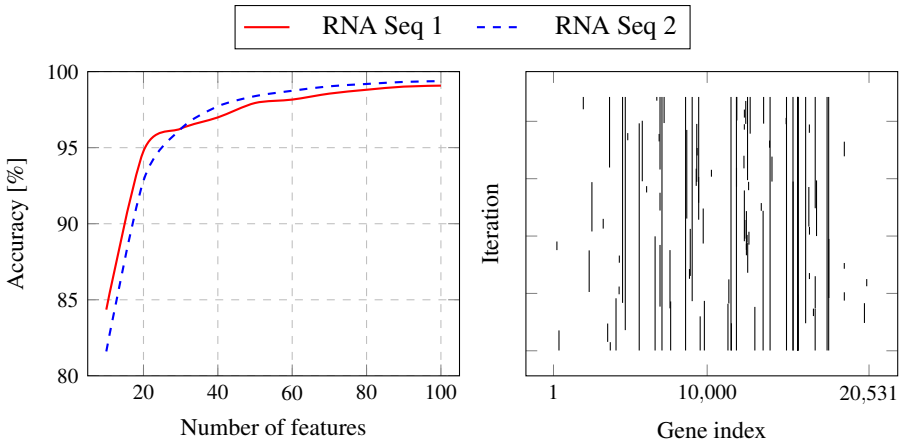

Figure 2: Performace of the *ADMM* method on the datasets from category 2. The left figure shows the average precistion accuracy. The right figure depicts one particular run showing which genes (x-axis) were selected during which iteration (y-axis).

4.3. Category 3: Neuroscience Application

In this last category, we consider the neuroscience application where several electrodes were attached to the patient's head. The patient was then asked to imagine certain actions (left-hand, right-hand, foot or tongue movements) and the EEG signal of his brain activity was recorded. We used the dataset IIIa from the BCI competition. During the experiment, 60 EEG channels were recorded with a 64-channel EEG amplifier from Neuroscan. There were 180 labeled trials.

Following [19], we represented each electrode signal by the autoregressive model $AR(p)$ over the 4s window in which the imaginary movements were performed. The p coefficients formed a group of features corresponding to that EEG channel. Model order $p = 3$ provided the best mean classification accuracy, which confirms the findings of [20]. This representation resulted in 60 groups of features, each with 3 features.

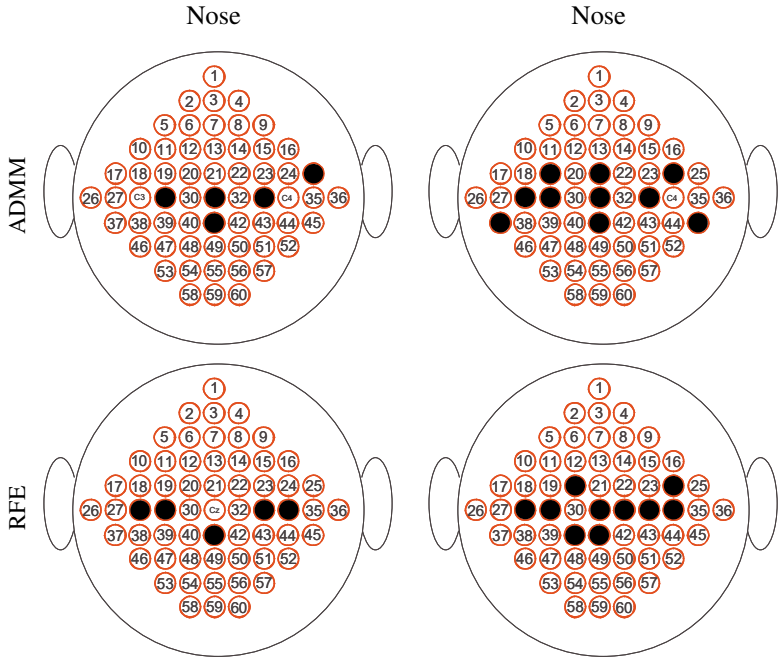

Figure 3: Selected electrodes for the *ADMM* and *RFE* methods for the dataset in category 3.

In Figure 3 we depict the selected channels. Channels 27-35 are located on the primary motor cortex and represent the movement of hand and foot. Channels 39-43 are on the primary somatosensory cortex and are responsible for the movement of tongue, lips and jaws. Channels 18-24 are located on the premotor cortex and on the supplementary motor area and encode the intention, selection and control of movements. Channels 10-11, 15-16, 17, and 25 are close to the Broca's area which is critically involved in language comprehension and semantics. Finally, channels 37 and 45 are on the Wernicke's area which is involved in processing words.

From its definition, the *RFE* method produces nested results, thus all the five most important channels are contained in the ten most important channels. This does not hold true for the *ADMM* method. For the most five important channels, the *ADMM* method identified channels 31, 41, 29, and 33 from the motor areas. These channels are from the areas related to the limb and mouth movement. In addition, the *ADMM* method also included channel 25 as it conveys the semantic information of the imaginary movements. The distribution pattern of the top five channels selected by the *RFE* method is very similar but it omitted the channel from the semantic area.

For the top ten channel, the *ADMM* method added more channels from motor areas. Although the result does not conserve the top five channels, it covers most of them. The top ten channels selected by the *ADMM* method include channels 37 and 45 from the language area. This demonstrates that the *ADMM* method is able to extract semantic or abstract concept information from the neural activity. In general, all the channels selected by the *ADMM* method are directly related to the task performed by the patient.

5. Conclusion

Group feature selection selects features in a grouped manner and is useful to improve the interpretability and the prediction performance of models. While a lot of work related to the group feature selection focuses on binary classification problems, we targeted multiclass classification problems. We formulated the group feature selection problem as a sparse learning problem in the framework of multiclass support vector machine and solved it by the ADMM method. We provided several improvements to increase the speed of the algorithm. The effectiveness of the proposed method has been demonstrated on several real-world datasets.

Acknowledgements

Dr. Fengzhen Tang's work is supported by the State Key Laboratory of Robotics (Grant No. Y7C120E101), Dr. Bailu Si's work is supported by the Distinguished Young Scholar Project of the Thousand Talents Program of China (grant No. Y5A1370101), while Lukáš Adam's work is funded by the Ministry of Science and Technology of China (Grant No. 2017YFC0804003) and by the National Natural Science Foundation of China (Grant No. 61329302).

Bibliography

[1] L. Adam and M. Branda. Nonlinear chance constrained problems: Optimality conditions, regularization and solvers. *J. Optim. Theory Appl.*, 170(2):419–436, 2016.

[2] L. Adam and M. Branda. Sparse optimization for inverse problems in atmospheric modelling. *Environmental Modelling & Software*, 79:256 – 266, 2016.

[3] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *Proceedings on European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)* , 2013.

[4] P. Balamurugan, A. Posinasetty, and S. Shevade. ADMM for training sparse structural SVMs with augmented l_1 regularizers. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 684–692, 2016.

[5] C. Bishop and M. E. Tipping. Variational relevance vector machines. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 46–53. Morgan Kaufmann, January 2000.

[6] B. Blankertz, K. R. Müller, D. J. Krusienski, G. Schalk, J. R. Wolpaw, A. Schlögl, G. Pfurtscheller, J. R. Millán, M. Schröder, and N. Birbaumer. The BCI competition III: validating alternative approaches to actual BCI problems. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):153–159, 2006.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011.

[8] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[9] E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted l_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, 2008.

[10] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[11] O. Chapelle and S. Keerthi. Multi-class feature selection with support vector machines. In *Proc. Am. Stat. Ass*, 2008.

[12] C. Du, C. Du, S. Zhe, A. Luo, Q. He, and G. Long. Bayesian group feature selection for support vector learning machines. In *PAKDD 2016, Part I*, pages 239–252, 2016.

[13] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, 2003.

[14] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2009.

[15] R. Hesse, D. R. Luke, and P. Neumann. Alternating Projections and Douglas-Rachford for Sparse Affine Feasibility. *IEEE Transactions on Signal Processing*, 62(18):4868–4881, 2014.

[16] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu. Joint embedding learning and sparse regression: a framework for unsupervised feature selection. *IEEE Transactions on Cybernetics*, 44(6):793–804, 2017.

[17] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

[18] L. Jacob, G. Obozinski, and J. P. Vert. Group lasso with overlaps and graph lasso. In *International Conference on Machine Learning*, pages 433–440, 2009.

[19] B. H. Jansen, J. R. Bourne, and J. W. Ward. Autoregressive estimation of short segment spectra for computerized EEG analysis. *IEEE Transactions on Biomedical Engineering*, BME-28(9):630–638, 1981.

[20] T. N. Lal, M. Schroder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Scholkopf. Support vector channel selection in BCI. *IEEE Transactions on Biomedical Engineering*, 51(6):1003–1010, 2004.

[21] H. A. Le Thi and M. C. Nguyen. DCA based algorithms for feature selection in multi-class support vector machine. *Annals of Operations Research*, pages 1–28, 2016.

[22] L. Meier, S. V. D. Geer, and P. Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society*, 70(1):53–71, 2010.

[23] T. Ochiai, S. Matsuda, H. Watanabe, and S. Katagiri. Automatic node selection for deep neural networks using group lasso regularization. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5485–5489, 2017.

[24] L. Pan, N. Xiu, and J. Fan. Optimality conditions for sparse nonlinear programming. *Science China Mathematics*, 60(5):759–776, 2017.

[25] S. Raman, T. J. Fuchs, P. J. Wild, E. Dahl, and V. Roth. The bayesian group-lasso for analyzing contingency tables. In *International Conference on Machine Learning*, pages 881–888, 2009.

[26] I. Rodriguez-Lujan, J. Fonollosa, A. Vergara, M. Homer, and R. Huerta. On the calibration of sensor arrays for pattern recognition using the minimal number of experiments. *Chemometrics and Intelligent Laboratory Systems*, 130:123 – 134, 2014.

[27] V. Roth and B. Fischer. The group-lasso for generalized linear models:uniqueness of solutions and efficient algorithms. In *International Conference on Machine Learning*, pages 848–855, 2008.

[28] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. A sparse-group lasso. *Journal of Computational & Graphical Statistics*, 22(2):231–245, 2013.

[29] N. Subrahmanya and Y. C. Shin. Sparse multiple kernel learning for signal processing applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 788 – 798, 2010.

[30] N. Subrahmanya and Y. C. Shin. A variational bayesian framework for group feature selection. *International Journal of Machine Learning & Cybernetics*, 4(6):609–619, 2013.

[31] H. Tao, C. Hou, F. Nie, Y. Jiao, and D. Yi. Effective discriminative feature selection with nontrivial solution. *IEEE Transactions on Neural Networks & Learning Systems*, 27(4):796–808, 2016.

[32] V. Vapnik. *Statistical Learning Theory*. New York ; Chichester : Wiley, 1998.

[33] A. Vergara, S. Vembu, T. Ayhan, M. A. Ryan, M. L. Homer, and R. Huerta. Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, 166-167:320 – 329, 2012.

[34] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J. M. Stuart, and C. G. A. R. Network. The cancer genome atlas pan-cancer analysis project. *Nat Genet*, 45(10):1113–1120, 2013.

[35] J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In *Proceedings of European Symposium on Artificial Neural Networks*, pages 219–224, Bruges, Belgium, April 21-23 1999.

[36] J. Xu, F. Nie, and J. Han. Feature selection via scaling factor integrated multi-class support vector machines. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3168–3174. AAAI Press, 2017.

[37] G. B. Ye, Y. Chen, and X. Xie. Efficient variable selection in support vector machines via the alternating direction method of multipliers. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 832–840. PMLR, 2011.

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

[39] X. Zhou and D. P. Tuck. MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data. *Bioinformatics*, 23(9):1106–1114, 2007.