Feature Selection via Scaling Factor Integrated
Multi-Class Support Vector Machines

Jinglin Xu, Feiping Nie, Junwei Han
Northwestern Polytechnical University, Xian, 710072, P. R. China
{xujinglinlove, feipingnie, junweihan2010} @gmail.com

Abstract

In data mining, we often encounter high dimensional and noisy features, which may not only increase the load of computational resources but also result in the problem of model overfitting. Feature selection is often adopted to address this issue. In this paper, we propose a novel feature selection method based on multi-class SVM, which introduces the scaling factor with a flexible parameter to renewedly adjust the distribution of feature weights and select the most discriminative features. Concretely, the proposed method designs a scaling factor with $p$ power to control the distribution of weights adaptively and search optimal sparsity of weighting matrix. In addition, to solve the proposed model, we provide an alternative and iterative optimization method. It not only makes solutions of weighting matrix and scaling factor independently, but also provides a better way to address the problem of solving $\ell_{2,0}$-norm. Comprehensive experiments are conducted on six datasets to demonstrate that this work can obtain better performance compared with a number of existing state-of-the-art multi-class feature selection methods.

1 Introduction

With the rapid advance of computer techniques, immense quantities of high dimensional data have been yielded in many applications, such as computer vision, pattern recognition and data mining, which pose unprecedented challenges for learning tasks due to the curse of dimensionality. Those high dimensional, complex and irrelevant features often make learning models tend to over-fitting and become less comprehensible. In order to utilize this kind of data effectively, feature selection is an essential component to successful data mining [Liu and Motoda, 2007]. As a process of selecting a subset of original features or learning the feature weighting according to certain criteria, feature selection is an important and frequently used dimensionality reduction technique which not only reduces the number of features, removes irrelevant, redundant, or noisy data, but also brings the immediate effects for real applications, such as gait recognition [Kusakunniran et al., 2010], expression recognition [Gupta and Xiao, 2011], image annotation [Huang et al., 2012] and disease diagnosis [Zhu et al., 2014].

During the past few decades, researchers have developed a large amount of feature selection algorithms. Based on the different computational models, feature selection methods can be roughly classified into the three categories [Guyon and Elisseeff, 2003; Zhao et al., 2010]: filter, wrapper and embedded models. The filter model utilizes the intrinsic properties or some global statistical information of the data, and evaluates features without any learning mechanism. Some popular feature selection methods based on filter model include Fisher Score (FS) [Duda et al., 1973], ReliefF (RF) [Liu and Motoda, 2007], mRMR (MR) [Peng et al., 2005], T-test (Ttest) [Montgomery et al., 2009], Chi-Square (Chi) [Liu and Setiono, 1995], Information Gain (IG) [Cover and Thomas, 1991] and so on. Most of these methods are evaluated independently for each feature, therefore they cannot deal with redundant features effectively. The wrapper model [Kohavi and John, 1997] uses a given learning algorithm to select features according to its evaluation criteria. In spite of good performance, they often have expensive computational cost. The embedded model [Lai et al., 2006] performs feature selection as a part of the training process, and often shows good performance since it is coupled with specific classifier. These different models are designed from different perspectives and all have their own strengths and weaknesses. In this paper, we propose a novel method based on embedded model for performing supervised feature selection.

Feature selection methods based on embedded model are coupled with specific classifiers, where Support Vector Machines (SVMs) are the most commonly used classifier. One of the standard embedded methods with SVMs is Recursive Feature Elimination (RFE) [Guyon et al., 2002] which is based on the idea that the importance of a feature should be related to the magnitude of its weight and the feature with the smallest magnitude is removed. In order to reduce the time consumption, $\ell_1$-SVM [Mangasarian, 2006] was proposed, which can obtain a sparse solution by changing the $\ell_2$-norm regularization in SVM to an $\ell_1$-norm. However, considering that the number of selected features using $\ell_1$-SVM is upper bounded by the sample size, a Hybrid Huberized SVM (HHSVM) [Wang et al., 2007] was presented via combining both $\ell_1$-norm and $\ell_2$-norm to form a more structured regu-
larization. Unfortunately, the above methods were designed only for binary classification. Actually many classification problems in practice involve many categories, thus some works [Hou et al., 2011; Yang et al., 2011; He et al., 2012; Han and Kim, 2015] were developed to deal with multi-class feature selection problem based on other models (such as iteratively re-weighted Least Squares and Half-quadratic optimization).

Although original SVMs can be extended to the multi-class case via one vs one or one vs all strategy, it ignores the correlation between classes, since the extended strategy just simply breaks the multi-class problem into several independent binary classification problems. To overcome this drawback, multi-class SVMs [Fan et al., 2008; Chapelle and Keerthi, 2008] are invented to solve them, which is equivalent to solving the following optimization problem:

$$
\min_{w, \xi} \frac{1}{2} \sum_{k=1}^{c} w_k^T X_i w_k + \alpha \sum_{i=1}^{n} \xi_i \\
\text{s.t.} w_k^T x_i - w_k^T x_i \geq 1 - \xi_i, \forall k \neq y_i, \xi_i \geq 0, i = 1 \ldots n
$$

where \(\alpha\) is the regularization parameter to balance the loss and penalty, and the bias \(b_k\) for each class is absorbed by augmenting the vector \(w_k^T\) and each sample \(x_i\) with an additional dimension: \(w_k^T \leftarrow [w_k^T, b_k]\), \(x_i^j \leftarrow [x_i^j, 1]\) . The format of problem (1) can be written as:

$$
\min_{w} \sum_{i=1}^{n} \left(1 - w_k^T x_i + \max_{k \neq y_i} w_k^T x_i\right) + \beta \sum_{k=1}^{c} \|w_k\|^2_2
$$

Given data matrix \(X\) and label matrix \(Y = \{y_{ik}\} \in \mathbb{R}^{n \times c}\), problem (2) can be generalized as:

$$
\min_{W} \frac{1}{2} \|W\|^2_p + C f(W^T X, Y)
$$

2.2 Multi-class SVM with \(\ell_{2,1}\)-norm Regularization

Recently, considering the superiority of \(\ell_{2,1}\)-norm in terms of [Nie et al., 2010], the work [Cai et al., 2011] combined the multi-class hinge loss with \(\ell_{2,1}\)-norm regularization term for the feature selection:

$$
\min_{W} \frac{1}{2} \|W\|_{2,1} + C f(W^T X, Y)
$$

where \(\|W\|_{2,1} = \sum_{i=1}^{d} \|w^i\|_2\), and \(w^i\) and \(w_j\) denote the \(i\)-th row and the \(j\)-th column vector of matrix \(W\), respectively. This work combines multi-class hinge loss with \(\ell_{2,1}\)-norm regularization and is solved by changing \(\ell_{2,1}\)-norm to trace optimization on \(W\).

3 The Proposed Method

3.1 Formulation

In this paper, we tend to learn \(c\) classifications functions \(f_k(x) = w_k^T x, 1 \leq k \leq c\) and perform simultaneous feature selection, namely find a small set of features which are good for all the classifiers. Considering the flexibility of each feature and making \(W\) with optimal sparsity, we propose a novel
framework of scaling factors for performing feature selection in the multi-class cases:

\[
\min \frac{1}{2} \|W\|_F^2 + C \sum_{k=1}^{c} \sum_{i=1}^{n} (1 - y_{ik}) \sum_{v=1}^{d} \theta_v^2 w_{vk} x_{vi} \quad (5)
\]

s.t. \( \theta \geq 0, \theta^T 1 = 1 \)

where \( \theta \) is a vector which consists of \( d \) scaling factors \( \theta_v (v = 1 \ldots d) \), and \( p \geq 1 \) is the power exponent of \( \theta_v \). In the following part, we will further analyze how to solve this framework.

### 3.2 Optimization

**Theorem 1.** Solving problem (5) can be transformed into solving the following problem:

\[
\min \frac{1}{2} \|W\|_{2, \frac{r}{r-1}}^2 + C \sum_{k=1}^{c} \sum_{i=1}^{n} (1 - y_{ik}) \sum_{v=1}^{d} \theta_v^2 w_{vk} x_{vi} \quad (6)
\]

**Proof.** Let \( \bar{w}_{vk} = \theta_v^{\frac{r}{r-1}} w_{vk} \), then \( w_{vk} = \theta_v^{-\frac{r}{r-1}} \bar{w}_{vk} \). After variable substitutions, problem (5) is equivalent to:

\[
\min \frac{1}{2} \sum_{k=1}^{c} \sum_{v=1}^{d} \bar{w}_{vk}^2 + C \sum_{k=1}^{c} \sum_{i=1}^{n} (1 - y_{ik}) \sum_{v=1}^{d} \bar{w}_{vk} x_{vi} \quad (7)
\]

s.t. \( \theta \geq 0, \theta^T 1 = 1 \)

Making the change of variables \( w_{vk} \leftarrow \bar{w}_{vk} \), problem (7) can be rewritten as:

\[
\min \frac{1}{2} \sum_{k=1}^{c} \sum_{v=1}^{d} w_{vk}^2 + C \sum_{k=1}^{c} \sum_{i=1}^{n} (1 - y_{ik}) \sum_{v=1}^{d} w_{vk} x_{vi} \quad (8)
\]

s.t. \( \theta \geq 0, \theta^T 1 = 1 \)

For updating scaling factors \( \theta \), we fix other variables and simplify the first term of problem (8) as follows:

\[
\min \frac{1}{2} \sum_{k=1}^{c} \sum_{v=1}^{d} w_{vk}^2 \Leftrightarrow \min \frac{1}{2} \sum_{k=1}^{c} \sum_{v=1}^{d} \|w^v\|_2^2 \Leftrightarrow \min \frac{1}{2} \sum_{k=1}^{c} \sum_{v=1}^{d} \theta_v^2 h_v \quad (9)
\]

where \( r = -p(r \leq -1) \) and \( h_v = \sum_{k=1}^{c} w_{vk}^2 = \|w^v\|_2^2 \). The Lagrangian function of problem (9) is:

\[
L(\theta_v, \lambda, \mu_v) = \sum_{v=1}^{d} h_v - \lambda (\sum_{v=1}^{d} \theta_v - 1) - \sum_{v=1}^{d} \mu_v \theta_v \quad (10)
\]

where \( \lambda \) and \( \mu_v \geq 0 \) are the Lagrangian multipliers.

Taking derivatives of \( L(\theta_v, \lambda, \mu_v) \) with respect to \( \theta_v \) and setting it to zero, and then according to the KKT condition \( \mu_v \theta_v = 0 \), we can calculate the solution of problem (9):

\[
\theta_v = \frac{(h_v)^{\frac{r}{r-1}}}{\sum_{s=1}^{d} (h_s)^{\frac{r}{r-1}}} \quad (11)
\]

Substituting original variables \( (p \) and \( w^v \)) back to Eq.(11), there is:

\[
\theta_v = \frac{(\|w^v\|_2^\frac{r}{r-1})^{\frac{r}{r-1}}}{\sum_{s=1}^{d} (\|w^s\|_2^\frac{r}{r-1})^{\frac{r}{r-1}}} \quad (12)
\]

### Algorithm 1 : Solving problem (5)

**Input:** Data \( X \in \mathbb{R}^{d \times n} \), label \( Y \in \mathbb{R}^{n \times c} \), parameters \( C \) and \( p \).

**Initialization:** Set \( t = 0 \).

1. Initialize \( W^{(0)} = \{ w_{vk} = 1 \} \).
2. Initialize scaled factor \( \theta \), where its \( v \)-th element is \( \theta_v = \frac{1}{c} \).

**Repeat:**

1. Calculate scaled factor of each feature \( \theta_{t+1} \) by Eq.(12).
2. Update feature weighting matrix \( W_{t+1} \) by solving problem (15) using Cramer’s algorithm [Fan et al., 2008].
3. \( t = t + 1 \)

**Until converges.**

**Output:** \( W \) and \( \theta \).

Using Eq.(12), the objective function of the first term in problem (8) can be further rewritten as:

\[
\sum_{v=1}^{d} \frac{\|w^v\|_2^2}{\theta_v^{\frac{r}{r-1}}} = \left( \sum_{v=1}^{d} \frac{\|w^v\|_2^2}{\theta_v^{\frac{r}{r-1}}} \right)^1 + p = \|W\|_{2, \frac{r}{r-1}}^p \quad (13)
\]

Therefore, we can arrive at problem (6) after substituting Eq.(13) into problem (8). This completes the proof and realizes rewriting problem (5) without an explicit \( \theta \) by using \( W \).

For the proposed method, we utilize an alternative and iterative method to solve \( W \) and \( \theta \) in problem (5). When \( W \) is fixed, we can use Eq.(12) to update \( \theta \); and then update \( W \) when \( \theta \) has been obtained. When \( \theta \) is fixed, problem (5) becomes a classical multi-class SVM model and is simplified as:

\[
\min \frac{1}{2} \|W\|_F^2 + C \sum_{k=1}^{c} \sum_{i=1}^{n} (1 - y_{ik}) \sum_{v=1}^{d} w_{vk} x_{vi} \quad (14)
\]

where \( \bar{x}_{vi} = \theta_v^{\frac{r}{r-1}} x_{vi} \) is known. Given \( \bar{X} = (\bar{x}_{vi}) \), problem (14) can be generalized as:

\[
\min \frac{1}{2} \|W\|_F^2 + C f(W^T \bar{X}, \bar{Y}) \quad (15)
\]

which can be solved by using [Fan et al., 2008; Cai et al., 2011]. For clarity, we summarize our method in Algorithm 1.
For problem (16), we also use the alternative and iterative method to solve $W$ and $\theta$. If $W$ is fixed, we can use Eq.(12) with $p = 1$ to update $\theta$, and then solve problem (15) to update $W$ if $\theta$ has been calculated.

### 3.3 Convergence

**Theorem 2.** The Algorithm 1 monotonically decreases the objective of problem (5) in each iteration and converges to the optimum of the problem.

**Proof.** Suppose that at the $t$-th iteration, we have obtained $W_t$ and $\theta_t$. In the $t+1$-th iteration, the updated $W$ is denoted as $W_{t+1}$. Considering that problem (5) can be rewritten as problem (6), we have:

$$W_{t+1} = \arg\min_W \frac{1}{2}\|W_{t+1}\|_{2, \frac{2}{1}}^2 + Cf(W_{t+1}^T X, Y)$$  \hspace{1cm} (18)

Referring to the way of argumentation for [Chang et al., 2014], through rewriting Eq.(18) we have:

$$\frac{1}{2}\|W_{t+1}\|_{2, \frac{2}{1}}^2 + Cf(W_{t+1}^T X, Y) \leq \frac{1}{2}\|W_t\|_{2, \frac{2}{1}}^2 + Cf(W_t^T X, Y)$$  \hspace{1cm} (19)

Thus, Eq.(19) proves that problem (5) and its variant problem (6) are lower bounded and their objective function value decreases after each iteration. \( \square \)

### 4 Experiment

#### 4.1 Experiment Setup

In our experiments, six different public datasets are adopted to illustrate the performance of different feature selection methods. These datasets include five microarray datasets (LUNG, CAR, SRBCT, GLIOM and MLLML)$^4$ and one primary tumor dataset (BrainT2)$^4$. These datasets are widely used by many previous feature selection methods such as [Padungweang et al., 2012; Liu et al., 2013; Hong et al., 2016] to evaluate their performance. All datasets are standardized to be zero-mean and normalized by standard deviation, and satisfy that in each dataset the number of samples is much less than the number of the features. We show more information of the datasets in Table 1.

<table>
<thead>
<tr>
<th>Property</th>
<th>Datasets</th>
<th>#Classes(c)</th>
<th>#Samples(n)</th>
<th>#Features(d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUNG</td>
<td>5</td>
<td>203</td>
<td>3312</td>
<td></td>
</tr>
<tr>
<td>CAR</td>
<td>11</td>
<td>174</td>
<td>9182</td>
<td></td>
</tr>
<tr>
<td>BrainT2</td>
<td>4</td>
<td>50</td>
<td>10367</td>
<td></td>
</tr>
<tr>
<td>SRBCT</td>
<td>4</td>
<td>83</td>
<td>2229</td>
<td></td>
</tr>
<tr>
<td>GLIOM</td>
<td>4</td>
<td>50</td>
<td>4434</td>
<td></td>
</tr>
<tr>
<td>MLLML</td>
<td>3</td>
<td>72</td>
<td>12582</td>
<td></td>
</tr>
</tbody>
</table>

In order to demonstrate the effectiveness of the proposed method ($p > 1$, denoted as square $\ell_{2, \frac{2}{1}}^2$-norm (SL2P)) and its simple version ($p = 1$, denoted as square $\ell_{2,1}$-norm (SL21)) for feature selection, we compare them with six classical feature selection methods (i.e. IG [Cover and Thomas, 1991], Ttest [Montgomery et al., 2009], FS [Duda et al., 1973], Chi [Liu and Setiono, 1995], RF [Liu and Motoda, 2007] and M-R [Peng et al., 2005]) as well as two widely used methods which are multi-class hinge loss with $\ell_2$-norm regularization (denoted as CS) [Fan et al., 2008] and multi-class hinge loss with $\ell_{2,1}$-norm regularization (denoted as L21) [Cai et al., 2011]. Because our work concerns multi-class feature selection, we do not compare binary feature selection method (like $\ell_1$-SVM).

In addition, for each dataset, we randomly sample 50% instances as training data and the remaining are used as test data. We tune parameters based on the training set and predict classification accuracy based on the test set. We choose the classification accuracy as metric, achieved by Crammer and Singer’s SVM [Fan et al., 2008] classifier using the selected features. The numbers of selected features are set as $\{10, 20, 30, 40, 50, 60, 70, 80\}$, respectively. A good feature selection algorithm should select a few of features that result in high classification accuracy.

#### 4.2 Experiment Results

In our experiments, we utilize the Crammer’s method [Fan et al., 2008] for solving multi-class SVM to evaluate the effectiveness of selected features. As shown in Table 2 and 3, in terms of the classification accuracy, it is obvious that proposed methods (SL21 and SL2P) outperform all other feature selection methods on six datasets.

In Tables 2 and 3, for almost all the datasets, the proposed SL2P method can achieve significant improvements compared with six filter feature selection methods. These results demonstrate that incorporating feature selection as a part of the training process and obtaining feature weighting analytically from a learning mechanism are more efficient. In contrast, filter feature selection methods select features without involving any learning mechanism.

Furthermore, compared with two multi-class SVM methods (CS and L21), the proposed SL21 method obtains better results on LUNG, SRBCT, GLIOM and MLLML datasets. For example, in Table 2, the proposed SL21 method averages achieve $12.6316\%$, $12.6984\%$, $4.1667\%$ and $12.7415\%$ improvements, respectively. Similarly, compared with CS and L21 methods, the proposed SL2P method can obtain the best results on LUNG, CAR, BrainT2, SRBCT, GLIOM and MLLML datasets. This is because the $\ell_{2, \frac{2}{1}}^2$-norm is more efficient than other norms (Frobenius-norm and $\ell_{2,1}$), by renewedly scaling factors and tuning parameter $p$ to allocate each feature weight with optimal sparsity. For example, in Table 2, the SL2P method achieves $11.5789\%$, $20.9459\%$, $14.2858\%$, $15.8730\%$, $12.5001\%$ and $16.6667\%$ improvements, respectively.

Besides, the proposed SL2P method can achieve higher classification accuracy than the proposed SL21 method. For example, in Table 3, it can achieve $3.1579\%$, $1.3513\%$,
Table 2: Classification accuracy of SVM as classifier on top 20 selected features.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Datasets</th>
<th>IG</th>
<th>Ttest</th>
<th>FS</th>
<th>Chi</th>
<th>RF</th>
<th>MR</th>
<th>CS</th>
<th>L21</th>
<th>SL21</th>
<th>SL2P</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUNG</td>
<td>75.7895</td>
<td>76.8421</td>
<td>71.5789</td>
<td>68.4211</td>
<td>78.9474</td>
<td>83.1579</td>
<td>76.8421</td>
<td>83.1579</td>
<td><strong>92.6316</strong></td>
<td>91.5789</td>
<td></td>
</tr>
<tr>
<td>CAR</td>
<td>74.3243</td>
<td>48.6486</td>
<td>41.8919</td>
<td>70.2703</td>
<td>74.3243</td>
<td>66.2162</td>
<td>55.4054</td>
<td>78.3784</td>
<td>77.0270</td>
<td><strong>87.8378</strong></td>
<td></td>
</tr>
<tr>
<td>BrainT2</td>
<td>64.2857</td>
<td>42.8571</td>
<td>64.2857</td>
<td>53.5714</td>
<td>64.2857</td>
<td>71.4286</td>
<td>64.2857</td>
<td>60.7143</td>
<td><strong>82.1429</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRBCT</td>
<td>68.2540</td>
<td>69.8413</td>
<td>82.5397</td>
<td>73.0159</td>
<td>77.7778</td>
<td>80.9524</td>
<td>66.6667</td>
<td>88.8889</td>
<td>90.4762</td>
<td><strong>93.6508</strong></td>
<td></td>
</tr>
<tr>
<td>GLIOM</td>
<td>50.0000</td>
<td>33.3333</td>
<td>63.8889</td>
<td>52.7778</td>
<td>58.3333</td>
<td>63.8889</td>
<td>66.6667</td>
<td>69.4444</td>
<td>72.2222</td>
<td><strong>80.5556</strong></td>
<td></td>
</tr>
<tr>
<td>MLLML</td>
<td>88.2353</td>
<td>82.3529</td>
<td>88.2353</td>
<td>90.1961</td>
<td>92.1569</td>
<td>88.2353</td>
<td>72.5490</td>
<td>90.1961</td>
<td>94.1176</td>
<td><strong>98.0392</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Classification accuracy of SVM as classifier on top 40 selected features.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Datasets</th>
<th>IG</th>
<th>Ttest</th>
<th>FS</th>
<th>Chi</th>
<th>RF</th>
<th>MR</th>
<th>CS</th>
<th>L21</th>
<th>SL21</th>
<th>SL2P</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUNG</td>
<td>80.0000</td>
<td>83.1579</td>
<td>77.8947</td>
<td>74.7368</td>
<td>81.0526</td>
<td>86.3158</td>
<td>78.9474</td>
<td>89.4737</td>
<td>91.5789</td>
<td><strong>93.6842</strong></td>
<td></td>
</tr>
<tr>
<td>CAR</td>
<td>77.0270</td>
<td>54.0541</td>
<td>48.6486</td>
<td>77.0270</td>
<td>82.4324</td>
<td>67.5676</td>
<td>70.2703</td>
<td>89.1892</td>
<td>91.8919</td>
<td><strong>93.2432</strong></td>
<td></td>
</tr>
<tr>
<td>BrainT2</td>
<td>57.1429</td>
<td>57.1429</td>
<td>53.5714</td>
<td>50.0000</td>
<td>64.2857</td>
<td>64.2857</td>
<td>67.8571</td>
<td>67.8571</td>
<td><strong>85.7143</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRBCT</td>
<td>76.1905</td>
<td>77.7778</td>
<td>79.3651</td>
<td>80.9524</td>
<td>80.9524</td>
<td>80.9524</td>
<td>68.2540</td>
<td>88.8889</td>
<td>92.0635</td>
<td><strong>98.4127</strong></td>
<td></td>
</tr>
<tr>
<td>GLIOM</td>
<td>72.2222</td>
<td>33.3333</td>
<td>72.2222</td>
<td>61.1111</td>
<td>63.8889</td>
<td>63.8889</td>
<td>69.4444</td>
<td>69.4444</td>
<td>75.0000</td>
<td><strong>83.3333</strong></td>
<td></td>
</tr>
<tr>
<td>MLLML</td>
<td>88.2353</td>
<td>86.2745</td>
<td>94.1176</td>
<td>88.2353</td>
<td>92.1569</td>
<td>86.2745</td>
<td>84.3137</td>
<td>96.0784</td>
<td>96.0784</td>
<td><strong>100.0000</strong></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Comparisons of ten feature selection methods on LUNG, CAR, BrainT2, SRBCT, GLIOM and MLLML datasets, respectively, in terms of classification accuracy using SVM as classifier.
For two multi-class SVMs methods (CS and L21), there is only one parameter C which can be adjusted on training data by grid search. Concretely, from $10^{-1}$ to $10^1$ with 0.1 step, we search for optimal parameters with the highest accuracy. For the proposed SL21 and SL2P methods, two parameters C and p need to be tuned. The parameter C controls the trade-off between discrimination and sparsity, and the parameter p emphasizes the proper sparsity of the feature weighting. They all play important roles in our methods. For $p = 1$, there is only one parameter C in Algorithm 1. For $p > 1$, there are two parameters C and p in Algorithm 1.

We also vary these two parameters from $10^{-1}$ to $10^3$ with 0.1 step and search for optimal parameters corresponding to the highest accuracy, where their results satisfy constraints of $\theta$ in the end. All classification results are conducted on each selected feature set from six datasets and methods by Crammer and Singer’s SVM [Fan et al., 2008].

5 Conclusion

To tackle the challenge of high feature dimensionality and noisy features in multi-class classification, this work proposes a novel and general feature selection framework, the square $\ell_{2,1}$-norm method, via flexibly scaling factor for each feature, where parameter $p$ is the power exponent of scaling factor. The proposed methods capture a small number of more discriminative features across multiple classes by tuning different values of $p$, and shows the connection between $\ell_{2,\frac{2}{1}}$-norm method and $\ell_{2,1}$-norm method. Furthermore, the theoretical proof of the convergence has also been provided. Compared with the state-of-the-art methods, the proposed methods can always achieve the best performance.

Acknowledgments

This work was supported in part by the National Science Foundation of China under Grants 61522207 and 61473231.

References

[Cai et al., 2011] Xiao Cai, Feiping Nie, Heng Huang, and Chris Ding. Multi-class $\ell_{2,1}$-norm support vector machine. In ICDM, pages 91–100, 2011.


