On the Use of Adaptive Weights for the $F_{\infty}$-Norm Support Vector Machine

Sungwan Bang$^1$ · Myoungshic Jhun$^2$

$^1$Department of Mathematics, Korea Military Academy; $^2$Department of Statistics, Korea University

(Received July 19, 2012; Revised August 27, 2012; Accepted September 13, 2012)

Abstract

When the input features are generated by factors in a classification problem, it is more meaningful to identify important factors, rather than individual features. The $F_{\infty}$-norm support vector machine (SVM) has been developed to perform automatic factor selection in classification. However, the $F_{\infty}$-norm SVM may suffer from estimation inefficiency and model selection inconsistency because it applies the same amount of shrinkage to each factor without assessing its relative importance. To overcome such a limitation, we propose the adaptive $F_{\infty}$-norm (AF$_{\infty}$-norm) SVM, which penalizes the empirical hinge loss by the sum of the adaptively weighted factor-wise $L_1$-norm penalty. The AF$_{\infty}$-norm SVM computes the weights by the 2-norm SVM estimator and can be formulated as a linear programming (LP) problem which is similar to the one of the $F_{\infty}$-norm SVM. The simulation studies show that the proposed AF$_{\infty}$-norm SVM improves upon the $F_{\infty}$-norm SVM in terms of classification accuracy and factor selection performance.

Keywords: Adaptive weight, $F_{\infty}$-norm penalty, factor selection, feature selection, support vector machine.

1. Introduction

Consider a linear binary classification problem with a training data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^p$ is $p$-dimensional input features for the $i^{th}$ observation and $y_i \in \{-1, +1\}$ denotes its class label. When given new input features $\mathbf{x} \in \mathbb{R}^p$, its classification is performed through the construction of a hyperplane $f(\mathbf{x}) = \beta_0 + \mathbf{x}^T \beta$, where $\beta_0$ is an intercept and $\beta \in \mathbb{R}^p$ is a coefficient vector for the input feature. We can assign one of two possible classes to it through the classification rule $\text{sign}(f(\mathbf{x}))$.

The standard 2-norm SVM estimates the coefficients of the hyperplane $(\hat{\beta}_0, \hat{\beta})$ by maximizing the margin between the training points for the positive and negative classes (Vapnik, 1995; Hastie et al., 2001)

$$\arg \max_{\beta_0, \beta} \frac{1}{||\beta||_2},$$

subject to $y_i \left( \beta_0 + \mathbf{x}_i^T \beta \right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall \ i$, and $\sum_{i=1}^n \xi_i \leq s$.

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0009204).

$^2$Corresponding author: Professor, Department of Statistics, Korea University, Seoul 136-701, Korea.

E-mail: jhun@korea.ac.kr
where \( \xi_i \) \( (i = 1, \ldots, n) \) are slack variables and \( s \) is a prespecified positive number that controls the overlap between the two classes. It can be shown that the optimization problem (1.1) of the 2-norm SVM can be equivalently expressed as the following 'loss + penalty' formulation (Vapnik, 1995; Hastie et al., 2001)

\[
\begin{align*}
\left( \hat{\beta}_0, \hat{\beta}_j \right)^{L_2} &= \arg \min_{\beta_0, \beta} \sum_{i=1}^{n} \left[ 1 - y_i \left( \beta_0 + \mathbf{x}_i^T \hat{\beta}_j \right) \right]_+ + \lambda \| \beta \|_2^2, \\
(1.2)
\end{align*}
\]

where the loss function \((1 - yf)_+\) is called hinge loss and the subscript "+" indicates the positive part \((z_+ = \max(z, 0))\). The tuning parameter \( \lambda \) controls the trade-off between loss and penalty, and there is a one-to-one correspondence between \( s \) in (1.1) and \( \lambda \) in (1.2).

The standard 2-norm SVM has been successfully applied to various classification areas due to its great flexibility and high level of classification accuracy; however, the 2-norm SVM classifier cannot automatically select input features. In classification problem, the feature selection plays important role in model building: excluding important features may produce severely biased estimation results; however, including irrelevant features may make it difficult to interpret the resultant model and reduce its classification accuracy. To accomplish the goal of automatic feature selection in the SVM classifier, Zhu et al. (2003) considered the 1-norm SVM by replacing the ridge penalty in (1.2) (Hoerl and Kennard, 1970) with the lasso penalty (Tibshirani, 1996)

\[
\begin{align*}
\left( \hat{\beta}_0, \hat{\beta}_j \right)^{L_1} &= \arg \min_{\beta_0, \beta} \sum_{i=1}^{n} \left[ 1 - y_i \left( \beta_0 + \mathbf{x}_i^T \hat{\beta}_j \right) \right]_+ + \lambda \| \beta \|_1. \\
(1.3)
\end{align*}
\]

To overcome the lack of oracle property of the lasso penalty, Zou (2007) suggested the hybrid SVM by adopting the adaptive lasso penalty (Zou, 2006)

\[
\begin{align*}
\left( \hat{\beta}_0, \hat{\beta}_j \right)^{H} &= \arg \min_{\beta_0, \beta} \sum_{i=1}^{n} \left[ 1 - y_i \left( \beta_0 + \mathbf{x}_i^T \hat{\beta}_j \right) \right]_+ + \lambda \sum_{j=1}^{p} \left| \hat{\beta}_j^{L_2} \right|^{-\gamma} \left| \beta_j \right|, \\
(1.4)
\end{align*}
\]

where the weights \( |\hat{\beta}_j^{L_2}|^{-\gamma} \) are computed by the 2-norm SVM and \( \gamma > 0 \) is some pre-specified number.

In this paper, we consider a classification problem in which the input features of the classifier are generated by factors; subsequently, the model is best interpreted in terms of significant factors. In such a situation, it is more meaningful to identify important factors, rather than individual features. In a regression problem, some regularization methods have been developed for automatic factor selection that include the group lasso (Yuan and Lin, 2006), the adaptive group lasso (Wang and Leng, 2008), the penalized method using composite absolute penalties (Zhao et al., 2009), and the adaptive sup-norm regularized quantile regression (Bang and Jhun, 2012a). In order to automatically and simultaneously select significant factors to estimate the SVM classifier, Zou and Yuan (2008) proposed the \( F_\infty \)-norm SVM that penalizes the empirical hinge loss by the sum of the factor-wise \( L_\infty \)-norm penalty; however, the \( F_\infty \)-norm SVM may suffer from estimation inefficiency and model selection inconsistency in the same way as the lasso and the 1-norm SVM. As a remedy, we suggest adopting the idea of the adaptive lasso (Zou, 2006) and propose the adaptive \( F_\infty \)-norm SVM, which allows different amounts of shrinkage to be imposed on different factors according to their relative importance.

The rest of the paper is organized as follows. In Section 2, we introduce the adaptive \( F_\infty \)-norm SVM and show that the adaptive \( F_\infty \)-norm SVM can be formulated as a linear programming(LP)
On the Use of Adaptive Weights for the $F_{\infty}$-Norm Support Vector Machine

Section 2. Methodology

2.1. The adaptive $F_{\infty}$-norm SVM

Suppose that the input features of the SVM classifier are generated by $G$ factors, that is, the features $x^T = (x_1, \ldots, x_p)$ are grouped into $G$ factors as $x_i^T = (x_{i1}, \ldots, x_{ijG})$, where $x_{ijG} = (x_j1, \ldots, x_jp_j)$ is a group of $p_j$ features for $j = 1, \ldots, G$ and $\sum_{j=1}^{G} p_j = p$. Then the classifier can be represented by

$$f(x) = \beta_0 + x^T \beta = \beta_0 + \sum_{j=1}^{G} x_{ijG} \beta_{ijG}.$$  \hspace{1cm} (2.1)

where $\beta_{ijG} = (\beta_{j1}, \ldots, \beta_{jp_j})^T \in \mathbb{R}^{p_j}$ is the coefficient vector associated with the $j$th factor. In such a situation, it is more reasonable to automatically and simultaneously select significant factors, rather than individual derived features. To accomplish the goal of automatic factor selection in the SVM classifier, Zou and Yuan (2008) proposed the $F_{1}$-norm SVM, which penalizes the empirical hinge loss by the sum of the factor-wise $L_{1}$-norm penalty

$$\min_{\beta_0, \beta} \sum_{i=1}^{n} \left[ 1 - y_i \left( \beta_0 + \sum_{j=1}^{G} x_{ijG} \beta_{ijG} \right) \right]_+ + \lambda \sum_{j=1}^{G} \lVert \beta_{ijG} \rVert_{1};$$  \hspace{1cm} (2.2)

where the infinite norm is defined by

$$\lVert \beta_{ijG} \rVert_{\infty} = \max \{ |\beta_{j1}|, \ldots, |\beta_{jp_j}| \}, \quad j = 1, \ldots, G.$$  \hspace{1cm} (2.3)

Owing to the nature of the $L_{\infty}$-norm, the $F_{\infty}$-norm SVM is able to simultaneously select significant features in a grouped manner, hence it is a more appropriate tool for factor selection than the 1-norm SVM in (1.3).

In the $F_{\infty}$-norm SVM (2.2), the same tuning parameter $\lambda$ is used for each factor-wise $L_{\infty}$-norm penalty; subsequently, the same amount of shrinkage is imposed on each factor without assessing its relative importance. In this paper, we suggest that different factors should be penalized differently according to their relative importance. In a typical linear regression setting, some researchers have suggested that an excessive penalty applied to important variables can degrade estimation efficiency and may influence model selection consistency (Fan and Li, 2001; Leng et al., 2006; Yuan and Lin, 2007; Zou, 2006). Ideally, small penalties should be used on the significant factors that we want to retain in the model; however, large penalties should be imposed on irrelevant factors in order to eliminate them from the final SVM classifier.

In order to further improve upon the $F_{\infty}$-norm SVM, we adopt the idea of the adaptive lasso from Zou (2006). Suppose that we first fit the 2-norm SVM classifier using all available input features. Then we suggest using the 2-norm SVM estimator $\hat{\beta}_{ijG}^{2G} (j = 1, \ldots, G)$ to construct the adaptively weighted $F_{\infty}$-norm penalty and propose the following adaptive $F_{\infty}$-norm (AF$_{\infty}$-norm) SVM

$$\min_{\beta_0, \beta} \sum_{i=1}^{n} \left[ 1 - y_i \left( \beta_0 + \sum_{j=1}^{G} x_{ijG} \hat{\beta}_{ijG}^{2G} \right) \right]_+ + \lambda \sum_{j=1}^{G} \lVert \beta_{ijG}^{2G} \rVert_{\infty} \lVert \beta_{ijG} \rVert_{\infty}. $$  \hspace{1cm} (2.4)
By using the solution to (2.4), the fitted AF∞-norm SVM classifier is \( \hat{f}(x) = \hat{\beta}_0^{\text{AF}_\infty} + \sum_{j=1}^G z_{i,(j)}^T \hat{\beta}_j^{\text{AF}_\infty} \), and the classification rule is sign(\( \hat{f}(x) \)). The factor-wise adaptively weighted \( L_\infty \)-norm penalty in (2.4) has been considered as the common variable selection in multiple quantile regression (Bang and Jhun, 2012b) as well as in the multicategory SVM (Zhang et al., 2008). It is interesting to note that in the context of quantile regression, Bang and Jhun (2012a) also suggested using a penalty term similar to the one of (2.4) to automatically select significant factors. When each individual feature is considered as a factor, that is, \( p_1 = \cdots = p_G = 1 \), the AF∞-norm SVM reduces to the hybrid SVM described in (1.4). Therefore, the AF∞-norm SVM is a generalization of the hybrid SVM; the AF∞-norm SVM has the capability to automate factor selection in the model fitting, whereas the hybrid SVM contains no information on the factors.

2.2. Computing algorithm

In this section, we show that the optimization problem (2.4) can be formulated as a linear programming (LP) problem. To derive the LP formulation of the AF∞-norm SVM, we introduce \( n \) slack variables such that

\[
\xi_i = \left[ 1 - y_i \left( \beta_0 + \sum_{j=1}^G z_{i,(j)}^T \beta_j \right) \right]_+, \quad i = 1, 2, \ldots, n. \tag{2.5}
\]

With such notation, the optimization problem (2.4) can be expressed as

\[
\begin{align*}
\arg\min_{\beta_0, \beta} & \quad \sum_{i=1}^n \xi_i + \lambda \sum_{j=1}^G \| \beta_j \|_\infty \| \beta_j \|_\infty \\
\text{subject to} & \quad y_i \left( \beta_0 + \sum_{j=1}^G z_{i,(j)}^T \beta_j \right) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0, \quad \forall i.
\end{align*} \tag{2.6}
\]

To further simplify the optimization problem (2.6), let the new variable \( M_j = \| \beta_j \|_\infty \) for \( j = 1, 2, \ldots, G \) and write \( \beta_0 \) as \( \beta_0^+ - \beta_0^- \) and \( \beta_j \) as \( \beta_j^+ - \beta_j^- \), where \( \beta_0^+ \geq 0, \beta_0^- \geq 0, \beta_j^+ = (\beta_{j1}^+, \ldots, \beta_{jG}^+) \geq 0 \), and \( \beta_j^- = (\beta_{j1}^-, \ldots, \beta_{jG}^-) \geq 0 \). Then the adaptively weighted \( F_\infty \)-norm penalty term \( \lambda \sum_{j=1}^G \| \beta_j^+ \|_\infty \| \beta_j^- \|_\infty \) can be linearly reformulated with some linear inequality constraints. By using these variables and notations, the optimization problem (2.4) and (2.6) can be equivalently expressed as

\[
\begin{align*}
\arg\min_{\beta_0, \beta} & \quad \sum_{i=1}^n \xi_i + \lambda \sum_{j=1}^G \| \beta_j^+ - \beta_j^- \|_\infty \| \beta_j^- \|_\infty \\
\text{subject to} & \quad y_i \left( \beta_0^+ - \beta_0^- + \sum_{j=1}^G z_{i,(j)} \left( \beta_j^+ - \beta_j^- \right) \right) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0, \quad \forall i, \\
M_j & \geq \beta_{jl}^+ + \beta_{jl}^-, \quad \forall j, l, \\
\beta_{jl}^+ & \geq 0, \quad \beta_{jl}^- \geq 0 \quad \text{and} \quad \beta_{jl}^+ \geq 0, \quad \beta_{jl}^- \geq 0, \quad \forall j, l.
\end{align*} \tag{2.7}
\]

This LP formulation of the AF∞-norm SVM is similar to the one of the \( F_\infty \)-norm SVM, but the key difference between these two LP formulations is that the AF∞-norm SVM uses the adaptive weights for penalizing different factors according to their relative importance in the objective function (2.7);
however, the \( F_{\infty} \)-norm SVM uses the equal weight for each factor. In the present study, we used the \texttt{lpSolve} package provided in R to implement the above LP problem and our R code is available for interested readers upon request.

3. Simulation Studies

In this section, simulation studies were conducted to evaluate the finite sample performance of the proposed AF\( \infty \)-norm SVM method. For comparison purpose, we also included the 2-norm SVM, the 1-norm SVM, the hybrid SVM, and the F\( \infty \)-norm SVM in the simulation. Two simulated models were considered, which are similar to those in Zou and Yuan (2008). For each of the simulated models, we generated 100 training observations, along with 100 independent observations for the validation set. The models were fitted on training data only, and the validation set was used to select the tuning parameter \( \lambda \). To evaluate the classification accuracy for each method, we also independently generated 10,000 observations as a test set. The model selection performance was measured by the number of correctly selected factors and features (NIC), the number of incorrectly selected factors and features (NC), and the number of times that the true model is correctly identified. To assess the sampling variability, this procedure was repeated 100 times independently. In the tables, the classification error, NC, and NIC were reported on averages over 100 runs. The numbers given in parentheses are the standard deviations of the classification errors.

3.1. Example 1

In this example, we first generated 8 factors \( z_1, \ldots, z_8 \) from the standard normal distribution with \( \text{Cov}(z_j, z_j') = 0.5^{\mid j-j'\mid} \). In addition, 40 random variables \( w_1, \ldots, w_{40} \) were independently generated from the standard normal distribution. Then the 40 covariates were obtained by

\[
\begin{align*}
    x_{1l} &= 2^{-\frac{1}{2}}(z_1 + w_l), \quad l = 1, \ldots, 6, \\
    x_{2l} &= 2^{-\frac{1}{2}}(z_2 + w_{l+4}), \quad l = 1, \ldots, 4, \\
    x_{3l} &= 2^{-\frac{1}{2}}(z_3 + w_{l+10}), \quad l = 1, \ldots, 6, \\
    x_{4l} &= 2^{-\frac{1}{2}}(z_4 + w_{l+16}), \quad l = 1, \ldots, 5, \\
    x_{5l} &= 2^{-\frac{1}{2}}(z_5 + w_{l+21}), \quad l = 1, \ldots, 4, \\
    x_{6l} &= 2^{-\frac{1}{2}}(z_6 + w_{l+25}), \quad l = 1, \ldots, 5, \\
    x_{7l} &= 2^{-\frac{1}{2}}(z_7 + w_{l+30}), \quad l = 1, \ldots, 4, \\
    x_{8l} &= 2^{-\frac{1}{2}}(z_8 + w_{l+34}), \quad l = 1, \ldots, 6.
\end{align*}
\]

The binary response \( y \) was generated by a logistic model with \( P(y = 1) = \exp(f(\mathbf{x}))/\left(1 + \exp(f(\mathbf{x}))\right) \) and \( P(y = -1) = 1 - P(y = 1) \), where the true hyperplane is given by

\[
    f(\mathbf{x}) = 1.2x_{11} - 0.8x_{12} + 1.6x_{13} + 1.5x_{14} - 3x_{15} + x_{16} + x_{21} - 0.9x_{22} - 1.1x_{23} - 1.3x_{24} \\
    + 1.5x_{61} + 2x_{62} - x_{63} - 2.5x_{64} + 3x_{65}.
\]

This model has 3 important factors and 15 important covariates. Table 3.1 summarizes the performance of each method for the above simulated model. As expected, the adaptively regularized methods (i.e., the hybrid-SVM and the AF\( \infty \)-norm SVM) were superior to their non-adaptive counterparts in terms of classification error and model selection performance. We can see that the number of incorrectly selected factors for the proposed AF\( \infty \)-norm SVM is smaller than that of the Hybrid SVM (while number of incorrectly selected variables for the Hybrid SVM is smaller than that of the proposed AF\( \infty \)-norm SVM); however, the proposed AF\( \infty \)-norm SVM outperformed the others in terms of classification error.
Table 3.1. Simulation results for Example 1

<table>
<thead>
<tr>
<th>Method</th>
<th>Test error</th>
<th>No. of factors selected</th>
<th>No. of variables selected</th>
<th>No. of times for true model</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-norm SVM</td>
<td>0.2034 (0.0187)</td>
<td>3.00</td>
<td>15.00</td>
<td>0</td>
</tr>
<tr>
<td>1-norm SVM</td>
<td>0.1857 (0.0203)</td>
<td>3.00</td>
<td>13.38</td>
<td>0</td>
</tr>
<tr>
<td>Hybrid SVM</td>
<td>0.1815 (0.0203)</td>
<td>3.00</td>
<td>12.18</td>
<td>0</td>
</tr>
<tr>
<td>$F_{\infty}$-norm SVM</td>
<td>0.1696 (0.0161)</td>
<td>3.00</td>
<td>15.00</td>
<td>25.00</td>
</tr>
<tr>
<td>AF$F_{\infty}$-norm SVM</td>
<td>0.1561 (0.0168)</td>
<td>2.99</td>
<td>14.96</td>
<td>21.13</td>
</tr>
</tbody>
</table>

The numbers in parentheses are standard deviations.

Table 3.2. Simulation results for Example 2

<table>
<thead>
<tr>
<th>Method</th>
<th>Test error</th>
<th>No. of factors selected</th>
<th>No. of variables selected</th>
<th>No. of times for true model</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-norm SVM</td>
<td>0.1399 (0.0114)</td>
<td>2.00</td>
<td>20.00</td>
<td>0</td>
</tr>
<tr>
<td>1-norm SVM</td>
<td>0.1471 (0.0110)</td>
<td>3.19</td>
<td>10.00</td>
<td>0</td>
</tr>
<tr>
<td>Hybrid SVM</td>
<td>0.1494 (0.0113)</td>
<td>2.96</td>
<td>8.03</td>
<td>0</td>
</tr>
<tr>
<td>$F_{\infty}$-norm SVM</td>
<td>0.1262 (0.0105)</td>
<td>1.30</td>
<td>13.00</td>
<td>24</td>
</tr>
<tr>
<td>AF$F_{\infty}$-norm SVM</td>
<td>0.1239 (0.0090)</td>
<td>0.42</td>
<td>4.20</td>
<td>66</td>
</tr>
</tbody>
</table>

The numbers in parentheses are standard deviations.

3.2. Example 2

In this example, 5 random variables $z_1, \ldots, z_5$ were independently generated from a standard normal distribution. In addition, 60 standard normal variables $\{\varepsilon_j\}_{j=6}^{15}$ were generated. Then the 60 covariates were obtained by

\[
\begin{align*}
x_{1l} &= z_1 + \varepsilon_l, \quad & \text{for } l = 1, \ldots, 10, \\
x_{3l} &= z_3 + \varepsilon_{l+20}, \quad & \text{for } l = 1, \ldots, 10, \\
x_{5l} &= z_5 + \varepsilon_{l+40}, \quad & \text{for } l = 1, \ldots, 10, \\
x_{2l} &= z_2 + \varepsilon_{l+10}, \quad & \text{for } l = 1, \ldots, 10, \\
x_{4l} &= z_4 + \varepsilon_{l+30}, \quad & \text{for } l = 1, \ldots, 10, \\
x_{31} &= \varepsilon_j, \quad & \text{for } j = 6, 7, \ldots, 15.
\end{align*}
\]

The binary response $y$ was generated by a logistic model with

\[
P(y = 1) = \exp(f(x))/(1+\exp(f(x)))
\]

and

\[
P(y = -1) = 1 - P(y = 1),
\]

where the true hyperplane is given by

\[
f(x) = 1 + 4z_1 + z_2.
\]

This model has 2 important factors and 20 important covariates. Table 3.2 reports the performance of each method. In terms of classification error and factor selection performance, the factor-wise penalized methods (i.e., the $F_{\infty}$-norm SVM and the AF$F_{\infty}$-norm SVM) worked better than their corresponding individual penalized methods. We can see that the factor-wise penalized methods always selected all important features; however the individual penalized methods (i.e., the 1-norm SVM and the hybrid SVM) tended to eliminate some important features. In particular, it is shown in Table 3.2 that the AF$F_{\infty}$-norm SVM performed better than the $F_{\infty}$-norm SVM.

4. Concluding Remarks

In this paper, we have proposed the AF$F_{\infty}$-norm SVM that performs simultaneous classification and factor selection. The AF$F_{\infty}$-norm SVM consists of two stages. At the first stage, we use the coefficients of the 2-norm SVM classifier to construct the factor-wise adaptively weighted $L_{\infty}$-norm penalty. Then we solve the adaptively weighted $F_{\infty}$-norm SVM by using the standard LP technique.
Simulation results show that the proposed the AF$_\infty$-norm SVM outperforms the F$_\infty$-norm SVM as well as the individual penalized methods (i.e., the 1-norm SVM and the hybrid SVM) in terms of classification accuracy and model selection performance.

To incorporate the factor information into regularized model fitting, we can also use the group lasso penalty (Yuan and Lin, 2006; Wang and Leng, 2008) as an alternative to the $L_\infty$-norm penalty. Considering computational efficiency, we favor the factor-wise $L_1$-norm penalty because the optimization problem using the group lasso penalty is formulated as a nonlinear programming (NLP) problem. In a simulation study, we know the group information of input features, but this is not available in applications of real data. In such situations, we can employ a clustering method to cluster the features into several groups such as the hierarchical clustering and the partitioning around medoids (PAM) algorithm (Kaufman and Rousseeuw, 1990).

References


