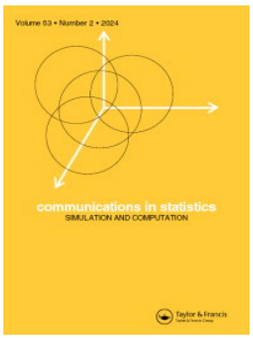


Communications in Statistics - Simulation and Computation

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/lssp20

Pliable lasso for the support vector machine

Theophilus Quachie Asenso, Puyu Wang & Hai Zhang

To cite this article: Theophilus Quachie Asenso, Puyu Wang & Hai Zhang (2024) Pliable lasso for the support vector machine, Communications in Statistics - Simulation and Computation, 53:2, 786-798, DOI: 10.1080/03610918.2022.2032160

To link to this article: https://doi.org/10.1080/03610918.2022.2032160

Published online: 31 Jan 2022.

COMMUNICATIONS IN STATISTICS - SIMULATION AND COMPUTATION®
2024, VOL. 53, NO. 2, 786–798
https://doi.org/10.1080/03610918.2022.2032160

Pliable lasso for the support vector machine

Theophilus Quachie Asenso[a], Puyu Wang[a], and Hai Zhang[a,b]

[a]School of Mathematics, Northwest University, Xi'an, China; [b]Faculty of Information Technology, Macau University of Science and Technology, Macau, China

ABSTRACT

In this article, we study the support vector machine with interaction effects. The pliable lasso penalty, which allows for estimating the main effects of the covariates X and the interaction effects between the covariates and a set modifiers Z is implemented to handle the interaction effect. Interaction variables are included in a hierarchical manner by first considering whether their corresponding main effect variables have been included in the model to avoid over-fitting. The loss function employed is the squared hinge loss, with the pliable lasso penalty and then, the block-wise coordinate descent approach is employed. The results from the simulation and real data show the effectiveness of the pliable lasso in building support vector machine models in situations where interaction effects are involved.

ARTICLE HISTORY
Received 9 August 2021
Accepted 12 January 2022

KEYWORDS
Classification; pliable lasso; support vector machine; variable selection

1. Introduction

In the field of machine learning, support vector machine (SVM) is one of the tools widely used for classification (Vapnik 1995; Huo et al. 2020). To begin with, let $S = \{s_i = (x_i, y_i)\}_{i=1}^n$ be a set of training data, where each $x_i \in \mathbb{R}^p$ is a vector of length p of the predictor variables and $y_i \in \{-1, 1\}$ denotes the class label Y. The linear SVM aims at outputting a hyper-plane $f(x) = x^T\beta + \beta_0$ that separates the two classes of data points. This hyper-plane categorizes new x according to the classification rule $\phi(x) = sign(f(x))$. SMV solves the following unconstrained optimization problem,

$$\min_{\beta} \quad \frac{\lambda}{2}||\beta||_2^2 + \frac{1}{n}\sum_{(x,y)\in S} \ell(\beta; \langle x, y\rangle), \tag{1}$$

where $\langle u, v\rangle$ denotes the standard inner product between vectors u and v and $\ell(\beta; \langle x, y\rangle)$ is the loss function. The parameter λ is the tuning parameter that controls the complexity of the machine. The loss fuction can be represented in two forms (Chang, Hsieh, and Lin 2008); one of which is the L1-SVM, which is the sum of the losses and solves the following optimization,

$$\min_{\beta} \quad \frac{\lambda}{2}||\beta||_2^2 + \frac{1}{n}\sum_{(x,y)\in S} \left[1 - y_i(\beta_0 + x_i^T\beta)\right]_+. \tag{2}$$

The second form is the L2-SVM, which uses the sum of the squared loss (see Lee and Mangasarian 2001) and solves the following optimization problem,

$$\min_{\beta} \quad \frac{\lambda}{2}||\beta||_2^2 + \frac{1}{n}\sum_{(x,y)\in S} \left(\left[1 - y_i(\beta_0 + x_i^T\beta)\right]_+\right)^2. \tag{3}$$

In the above equations, we can see that they have been written in the form *penalty* + *loss*, hence the parameter λ represents the regularization parameter which controls the balance between the

CONTACT Zhang Hai zhanghai@nwu.edu.cn School of Mathematics, Northwest University, Xi'an, China

loss and the penalty. The function $[1-y]_+ = \max(0, 1-y)$ in (2) and (3) is called the hinge loss. Several algorithms have been proposed to solve the above optimization problem. Some employ the standard linear programming approach (see Zhu et al. 2003; Wang, Zhu, and Zou 2006), while others use the coordinate descent algorithm (see Chang, Hsieh, and Lin 2008). See Lee and Mangasarian (2001), (Platt 1998), and Martinez (2017) for other related examples. In this article we propose the pliable lasso approach to train the support vector machine. This approach will allow the SVM model to consider possible interactions among the covariates.

2. Regularization approaches to solving SVM

Instead of solving (1) in a one time operation, Hastie et al. (2004) considered training the SVM on an entire path of λ values. This work gave room for one to consider different regularization approaches for the SVM. The work of Zhu et al. (2003) considered a situation where more variables need to be shrink to zero i.e., situations where sparsity is required. Their work introduced the 1-norm support vector machine and it is written as follows,

$$\min_{\beta} \sum_{i=1}^{n} \left[1 - y_i(\beta_0 + x_i^T\beta)\right]_+ + \lambda||\beta||_1. \tag{4}$$

Just like the SVM with the ridge penalty, the 1-norm SVM also performs some form of shrinkage. However, in the case where λ is very large, some of the coefficients will be zero, hence promoting sparsity and does some feature selection, unlike the ridge penalty SVM. One of the disadvantages of the 1-norm SVM is that in situations where some inputs are highly correlated, and are all relevant to the output, the 1-norm penalty ends up picking few of them and shrinking the rest to zero (Wu, Zou, and Yuan 2008; Martinez 2017). This implies that "group selection" will be a challenge to the 1-norm SVM. To handle variable selection and also to help select groups of correlated variables together, Wang, Zhu, and Zou (2006) proposed a double regularized support vector machine (DrSVM) which uses the elastic net penalty (see Friedman, Hastie, and Tibshirani 2010). It is a combination of the L1-norm penalty and the L2-norm penalty. The optimization function is given as

$$\min_{\beta} \sum_{i=1}^{n} \left[1 - y_i(\beta_0 + x_i^T\beta)\right]_+ \frac{\lambda_2}{2}||\beta||_2^2 + \lambda_1||\beta||_1, \tag{5}$$

where λ_1 and λ_2 are both the regularization parameters. The results from their work shows how efficient the double regularized SVM is in handling variable selection while considering highly correlated input variables. Both the 1-norm SVM and the DrSVM use the quadratic programming approach.

In this article, we consider using another approach in solving the optimization problem (1). Our motivation comes from the coordinate descent method for solving SVM. The loss function will also be the L2-SVM which uses the sum of squares losses (see Chang, Hsieh, and Lin 2008). However, instead of considering only the covariates X, we extend the SVM model to include an additional variable $Z \in \mathbb{R}^{N\times K}$ which serves as a modifying term and we assume that there is an interaction between the covariates and the modifying variables. In this situation, instead of using the linear SVM with the sum of squared loss like Chang, Hsieh, and Lin (2008), we intend to use the pliable lasso penalty to include these interactions in a hierarchical format. Example of such interaction models is the Glinternet models proposed by Lim and Hastie (2015) which is used for logistic regression.

The Pliable lasso is a new interactive model, introduced in 2018 by Robert Tibshirani and Jerome Friedman (see Tibshirani and Friedman 2020). The model is an extension of the original lasso problem. This extension allows the lasso regression model to accept both modifying variable, predictors, and the outcome. We introduce the model in what follows. Given covariates x_i, modifying variables z_i and response variables y_i, an approach to estimating the effects of the covariates x_i and the set of modifying variables z_i is the regression model which is used for estimating the effects of the covariates and the modifying variables. This model is given in the Gaussian form as,

$$\begin{aligned} y &= \beta_0 + Z\theta_0 + \sum_{j=1}^{p} X_j(\beta_j + Z\theta_j) + \epsilon_i \\ &= \beta_0 + Z\theta_0 + X\beta + \sum_{j=1}^{p} (X_j \odot Z)\theta_j + \epsilon_i. \end{aligned} \tag{6}$$

In the Equation (6) above, $\epsilon_i \sim \mathbb{N}(0,1)$, $(X_j \odot Z)$ denotes the $N \times K$ matrix formed by multiplying each column of Z component-wise by the column vector X_j. Adding the pliable lasso penalty to (6) gives the least square regression objective

$$\begin{aligned} M(\beta_0, \theta_0, \beta, \theta) = \frac{1}{2N} \sum (y_i - \beta_0 + Z\theta_0 + \sum_{j=1}^{p} X_j(\beta_j + Z\theta_j))^2 \\ + (1-\alpha)\lambda \sum_{j=1}^{p} (||(\beta_j, \theta_j)||_2 + ||\theta_j||_2) + \alpha\lambda \sum_{j,k} |\theta_{j,k}|_1, \end{aligned} \tag{7}$$

where β_0 and θ_0 are the intercept and main effect term for the modifiers, respectively, β is the vector of β_j terms and θ is $p \times K$ matrix of parameters with jth row θ_j and individual entries θ_{jk}, $||(\beta_j, \theta_j)||_2 + ||\theta_j||_2$ is an overlapping group lasso, included in the penalty. The goal is to minimize $M(\beta_0, \theta_0, \beta, \theta)$.

The pliable lasso has been implemented on some existing statistical models like the Cox survival model (see Du and Tibshirani 2018) and on the multinomial logistic regression (see Asenso, Zhang, and Liang 2020). In this article, the pliable lasso penalty is applied on the support vector machine. Our approach involves the sum of squares of the loss function and the block-wise coordinate descent procedure will be used in optimizing the objective function.

3. The pliable Lasso for SVM

We begin this section by giving the details of the input data which contains N data points of training set $\{(X_i, Z_i, y_i), i = 1, 2, ..., N\}$, with $X_i = (x_{i1}, x_{i2}, ..., x_{ip})$ representing the i^{th} input with dimension p, $Z_i = (z_{i1}, z_{i2}, ..., x_{iK})$ representing the i^{th} input for the modifying variables and $y_i \in \{-1, 1\}$, representing the class of the i^{th} categorical response. We assume basically that some or all of the coefficients are influenced by these modifying variables. We represent the SVM with the pliable lasso penalty as

$$\begin{aligned} \min_{\{\beta_0, \theta_0, \beta, \theta\} \in \mathbb{R}^{(1+K+p+(p\times K))}} \frac{1}{N} \sum_{i=1}^{N} \left(\left[1 - y_i \left(\beta_0 + Z\theta_0 + \left(\sum_{j=1}^{p} X_{ij}\beta_j + W_j\theta_j \right) \right) \right]_+ \right)^2 \\ + \left[(1-\alpha)\lambda \sum_{j=1}^{p} (||(\beta_j, \theta_j)||_2 + ||\theta_j||_2) + \alpha\lambda \sum_{j,k} |\theta_{j,k}|_1 \right]. \end{aligned} \tag{8}$$

The above equation is similar to that of Chang, Hsieh, and Lin (2008). The only difference is the penalties. One can also note that the loss function is similar to the least square loss function, used in the main pliable lasso work (see Tibshirani and Friedman 2020). This allows us to implement the same procedure used in their article. We employ the implementation of block-wise coordinate descent algorithm and follow the same procedure of the Pliable lasso (see Tibshirani and Friedman 2020) but at each pass of the algorithm, we only optimize with the set $\{y_i(\beta_0 + Z\theta_0 + (\sum_{j=1}^{p} X_{ij}\beta_j + W_j\theta_j)) < 1\}$ and represent $\beta_0 + Z\theta_0 + (\sum_{j=1}^{p} X_{ij}\beta_j + W_j\theta_j)$ by $\bar{n}$. The summary of our algorithm is given as algorithm (1) below and the details of the steps involved are shown in the appendix.

Algorithm 1. Algorithm for the SVM pliable lasso model (Plasso-SVM)

Initialization;
for *Decrement* λ **do**
  Initialize Set all $(\beta_0, \quad \beta_j) = 0$, $(\theta_0, \quad \theta_j) = 0 \quad (j = 1, 2, \ldots p)$
    1. compute $\hat{\vec{n}}$, $b = 1(y_i(\hat{\vec{n}}) < 1)$ and $r = [1 - y(\hat{\vec{n}})]_+$
    2. Compute $\hat{\beta}_0$ and $\hat{\theta}_0$.
  For $j \in \{1, 2, \ldots p, 1, 2, \ldots\}$ Check the condition for $(\hat{\beta}_j, \hat{\theta}_j) = 0$.
  if $(\hat{\beta}_j, \hat{\theta}_j) = 0$ **then**
    skip to next *j*;
  else
    Compute $\hat{\beta}_j$ and then check if $\hat{\theta}_j = 0$;
  end
  if $\hat{\theta}_j = 0$ **then**
    Update β_j with the computed $\hat{\beta}_j$ and skip to next *j*.;
  else
    Use the generalized gradient decent procedure to compute $(\hat{\beta}_j, \hat{\theta}_j)$
  end
end

Remark 3.1. In our algorithm, we obtain the β_0 and θ_0 by the regression of the residual (r) on $(1, Z)$ since both β_0 and θ_0 do not attract any penalty.

4. Experiments

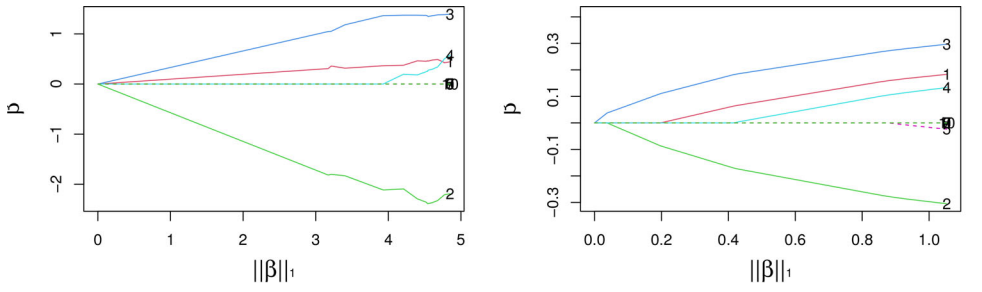
4.1. Simulation

To show the effectiveness of the proposed method (Plasso-SVM), we compared it to two other SVM algorithms which are also executed on regularization path. They are the 1-norm SVM and the doubly regularized SVM (DrSVM). We generated data from the model used in the work of Tibshirani and Friedman (2020) with $N = 100$, $p = 10$ and $p = 500$, $K = 4$ and standard normal predictors. The model is given as

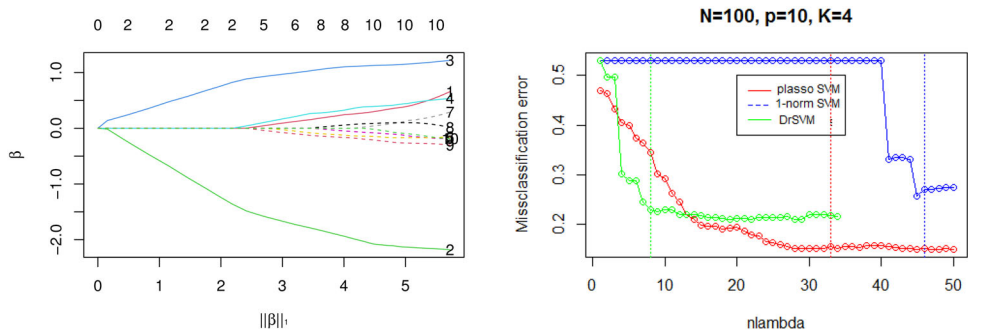
$$\mu = X_1\beta_1 + X_2\beta_2 + X_3(\beta_3 e + 2Z_1) + X_4\beta_4(e - 2Z_2) + .5\varepsilon. \tag{9}$$

Here, $\beta = (2, -2, 2, 2, 0, 0, \ldots)$, Z is $N \times K$ matrix randomly generated from the normal distribution, $\varepsilon \sim N(0, 1)$ and $e = (1, 1, 1, \ldots)$. The response vector μ was then transform into a two class response variable to represent $y \in \{-1, 1\}$. The Figure 1 shows the estimated coefficients for plasso-SVM, 1-norm SVM and the DrSVM after implementing them on the model (9). We can see that our proposed algorithm is also able to select various non zero coefficients for model (9) starting from a position where all $\hat{\beta}_j$ s are zero. From Figure 1, it is clear that the 1-norm SVM selects the least variables and the Plasso-SMV selects more variables than the other two.

We compared the prediction results of the plasso-SVM to the 1-norm SVM and DrSVM to study their behavior in terms of models with interaction effect. All the works were done using the R programming language and the authors for these two models provided us with their R codes to be used as the benchmark models. We allowed both models to function in their default states and supplied X and Z together for both algorithms as the input. Figure 2 shows the results from the test data from the three models. It can be seen that in the case where the data involve some interaction effects, the plasso-SVM performed better than the other two. The accuracy of the three algorithms were measured using the misclassification error and the prediction was done by using the model chosen after performing 5-fold cross validation. The cross validation plot for the pliable lasso is shown in Figure 2a.



(a) Coefficients plot for 1-norm SVM (b) Coefficients plot for DrSVM



(c) Coefficients plot for plasso SVM (d) Test error

Figure 1. Plots for the non-zero of coefficients at each iteration ploted against the L1-norm of the coefficients. The misclassification error plot shows the error from the test data for Plasso SVM, 1-norm SVM and the DrSVM models. Here, nlambda represents the step or the iteration number. The vertical lines indicate the error for chosen model after cross validation.

We further applied the same model (9) on data set, generated with $N = 100$, $p = 500$ and $K = 4$. Figure 2c and d show the coefficient plot and the test error results. From the coefficient plot, it can be seen that the plasso SVM has been able to select all the relevant variables (variables 1–4) and some other non relevant variables for large p as far as model (9) is concerned.

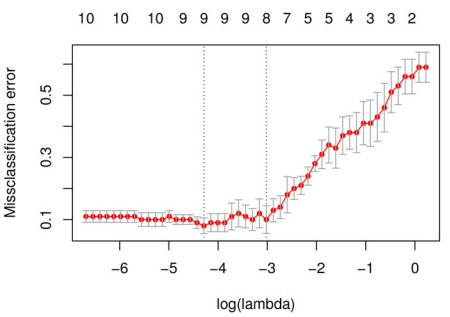
4.2. Training the modifying variable Z from the *known−to−unkown* situation

There are some cases where the modifying variable Z will not be observed from the main data set. There are various methods proposed to estimate the variable and (Tibshirani and Friedman 2020) in their article summarized these situations as situation $unknown - to - unknown$ and $known - to - unknown$. In this section, we consider the same model (9) but in a situation where Z is estimated from the covariates X. We train Z with X by using the lasso from the Glmnet package and then use the result to also predict Z for testing on our model. The results in Figure 3a and b show that the pliable lasso SVM can be used to build SVM with interactive variables in situations where Z is not observed. Using this for prediction (classification) after cross validation, the plasso SVM obtained a misclassification error of 0.136 and the 1-norm SVM and the DrSVM obtained 0.170 and 0.140 error values, respectively.

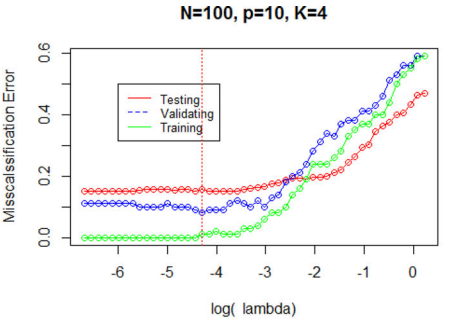
4.3. Real data

4.3.1. Breast cancer data

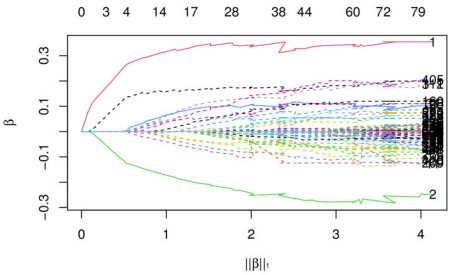
We applied the pliable lasso on the breast cancer data set which belongs to the University of Wisconsin Hospitals, Madison (Wolberg and Mangasarian 1990). The data set was downloaded



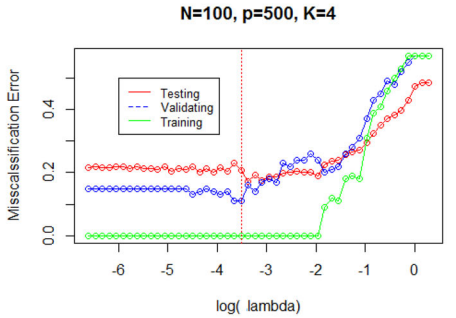
(a) Cross validation plot for $p = 10$ for model (9)



(b) Plot for the train, validation and test error for model (9) for $p = 10$



(c) Coefficient plot for (9) for $p = 500$



(d) Plot for the train, validation and test error for model (9) for $p = 500$

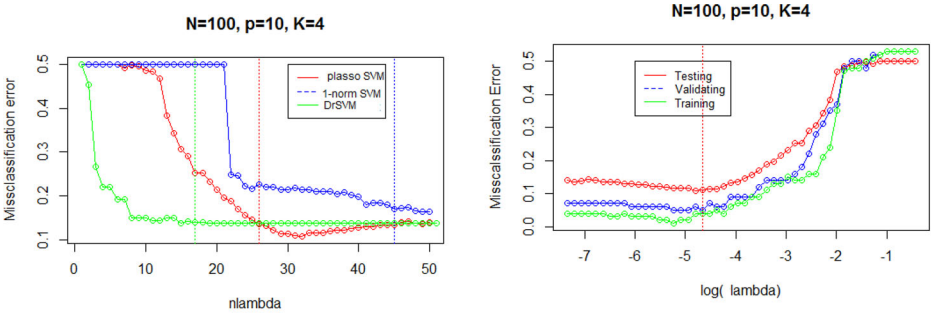
Figure 2. Plots from the test results showing the miscalssification error for each of the λ values. The vertical lines represent the choice of λ from the cross validation.

from the UCI machine learning repository website (Dua and Casey 2019). It contains 699 observations, 10 features, and 2 classes, grouping the breast mass cytology into benign and malignant. In our experiment, we randomly selected 400 observations for training and 282 for testing. There are nine covariates used in the experiment of the main creator of this dataset. Each feature is represented on a scale of 1–10 with 1 being the closest to benign and 10 being the closest to anaplastic. To preprocess this data, we represented each of the 10 scale as a 10×10 dummy variables, given us $p = 90$ after preprocessing. We also obtained Z from the subset of X. This is an example of the known-to-unknown procedure of obtaining Z as specified by Tibshirani and Friedman (2020). We selected the X_{js} with higher uni-variate score from the training set to represent Z. We analyzed the prediction ability of the plasso-SVM by comparing with the 1-norm SVM and the DrSVM. Figure 4 shows the results from this analysis. The plasso SVM obtained a cross validation and test error of 0.020 and 0.046, respectively, with 75 features. The DrSVM obtained 0.030 and 0.053 for cross validation and test error, respectively. The 1-norm svm could not perform as good as the DrSVM and the Plasso-SVM as it obtained a cross validation and test error of 0.3375 and 0.368, respectively.

4.3.2. Gene expression dataset

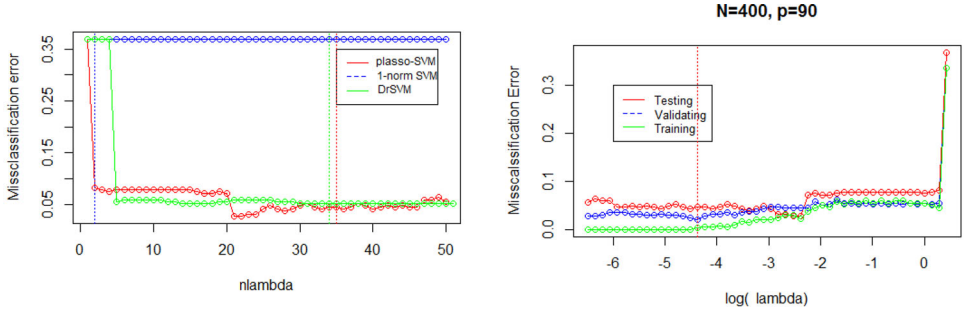
This dataset is part of the work of Golub et al. (1999) and was obtained from the kaggle website.[1] It shows how one can classify the new cases of cancer gene expression monitoring (via DNA microarray). The data was used to group patients into those with acute myeloid leukemia (AML) and those

[1]https://www.kaggle.com/crawford/gene-expression?select=data_set_ALL_AML_independent.csv



(a) Misclassification plot from the trained Z. (b) Training, validating and testing plot for the trained Z

Figure 3. Results from the *known-to-unknown* situation for obtaining the modifying variable *Z*.



(a) Misclassification plot from the breast cancer dataset.

(b) Training, validating and testing plot.

Figure 4. Results from the breast cancer dataset.

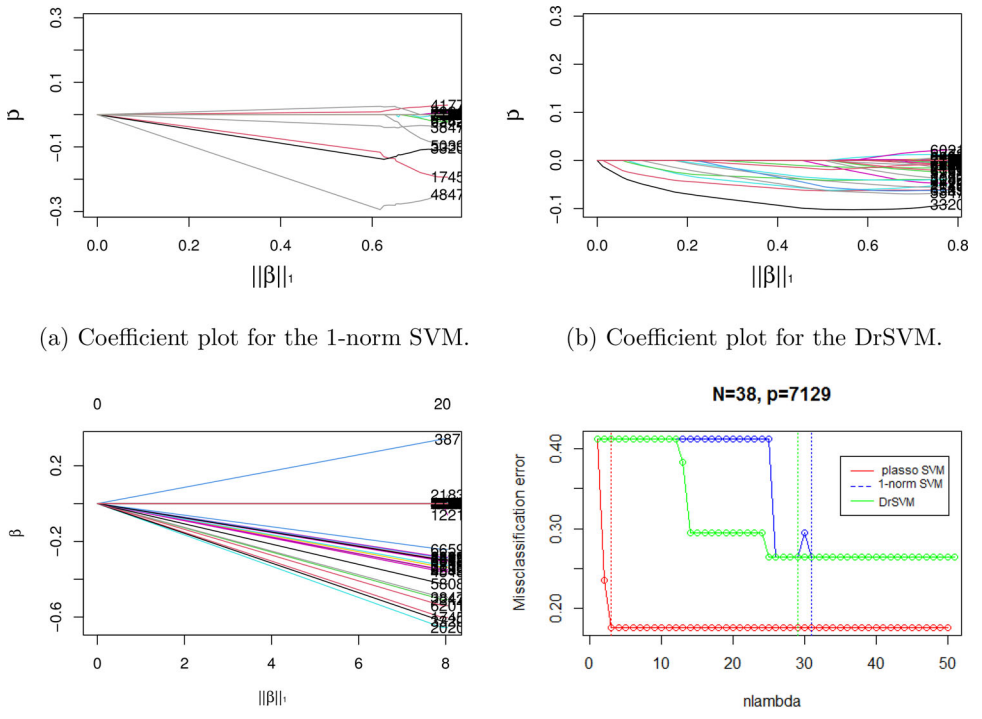
with acute lymphoblastic leukemia (ALL). The dataset contains 38 training samples and 34 test samples with each data point having $p = 7129$ genes. Several SVM approaches have been proposed to classify this type of microarray dataset. However, we are considering a situation of adding an extra variable to the covariates to consider the possibility of interactions among the genes. We selected the interaction variable from the covariates by considering the variables with high univariate Z-score. We compared the results of the plasso-SVM with the DrSVM and the 1-norm SVM. In the two cases, we supplied both the X and Z as input. We chose $\lambda_{\min}$ ratio of 0.1 for the plasso-SVM. Table 1 shows the results from the 5 fold cross validation and the test data. In this situation, the DrSVM obtained the same test error as the 1 norm-SVM but the plasso-SVM still performed well. Figure 5 shows the coefficient plots for the three models as well as the cross validation plot for the plasso-SVM.

4.3.3. Colon cancer dataset

This dataset is part of the work of Alon et al. (1999). For this dataset, the task is to be able to classify tissues as normal or cancer by using microarray data. The original data contains 40 cancer tissues and 22 normal tissues. Each datum is a vector of $p = 2000$. In our work, we randomly selected 32 data set for training and 30 for testing. The data was pre processed before using for the analysis. Each row of the covariates was standardized to have mean 0 and unit variance. Then each column of the consequence matrix was standardized to have zero mean and unit variance. We selected Z from X by choosing the covariates with high univariate Z-score. We then used the GLMNET lasso to train and predict test data for Z from the test data for X. The same data was used by Shevade and Keerthi (2003) and they obtained a test error of 0.177. We compared the

Table 1. Average test error, cross validation error and the number of features selected by using the acute leukemia dataset.

	CV error	Test error	# features/7129
DrSVM	0.178	0.265	27
1-Norm SVM	0.053	0.265	9
Plasso-SVM	0.0285	0.1764	20



(a) Coefficient plot for the 1-norm SVM.

(b) Coefficient plot for the DrSVM.

(c) Coefficient plot for the Plasso-SVM

(d) Test error plot for the three models

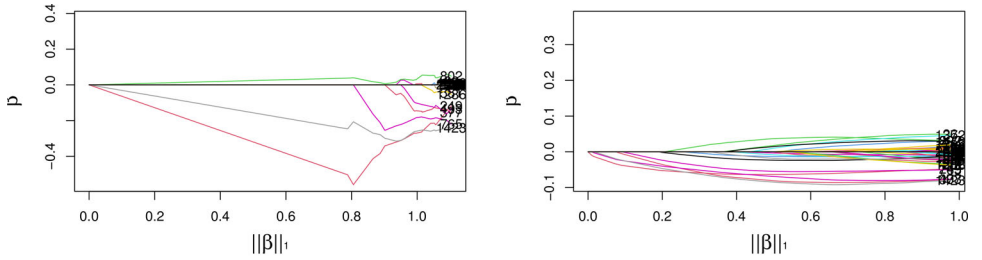
Figure 5. Coefficient plots and test error plot for the acute leukemia dataset.

plasso-SVM with the 1-norm SVM and the DrSVM. The tuning parameters were chosen according to 5-fold cross-validation, and the final model was tested using the test data. In this data set, all the three algorithms obtained a misclassification error of 0.1667. However, plasso-SVM obtained this results with 18 none zero coefficients, the 1-norm SVM with 12 none zero coefficients and the DrSVM with 17 none zero coefficients. Figure 6 shows the coefficient plot for the three algorithms and the plot for the misclassification error. Figure 7 shows the cross validation plot and the training, validation and testing plot for the plasso-SVM.

4.3.4. Prostate cancer dataset

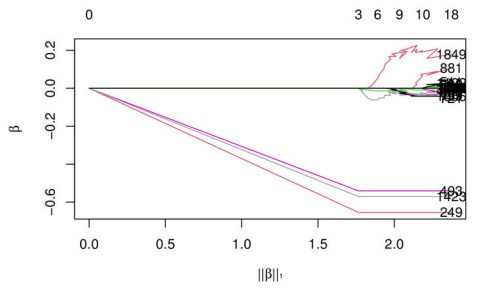
This dataset is part of the work of Singh et al. (2002). It was obtained from the Interdisciplinary Computing and Complex BioSystems website[2] as part of the work of Glaab et al. (2012). The original data contains 52 tumor samples and 22 normal samples as control. Each datum is a vector of $p = 2135$ genes. In our work, we randomly selected 52 data set for training and 50 for testing. The data was pre processed before using for the analysis. We selected Z from X by choosing the covariates with high univariate Z-score. We then used the GLMNET lasso to train and predict Z

[2]http://ico2s.org/datasets/microarray.html

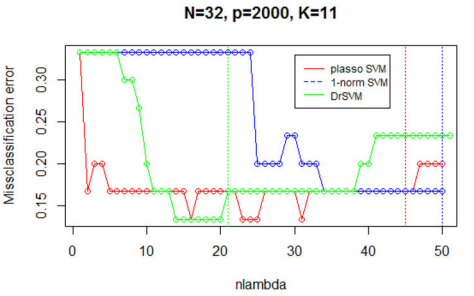


(a) Coefficient plot for the 1-norm SVM.

(b) Coefficient plot for the DrSVM.

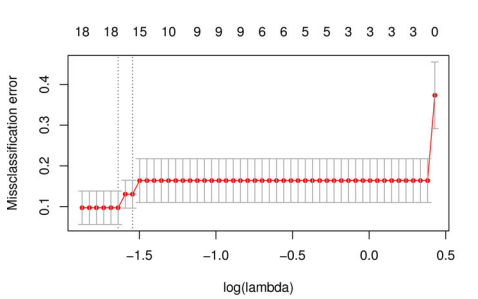


(c) Coefficient plot for the Plasso-SVM

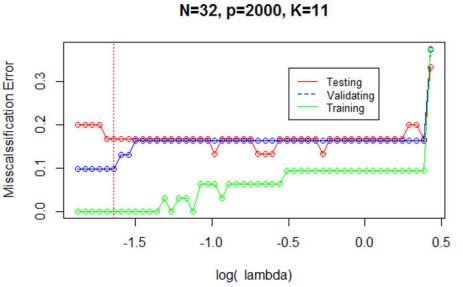


(d) Test error plot

Figure 6. Coefficient plots and test error plots for the DrSVM, 1-Norm SVM and the Plasso-SVM. The vertical lines in the test error plot shows the various model selected by the five-fold cross validation.



(a) Cross validation plot for Plasso-SVM



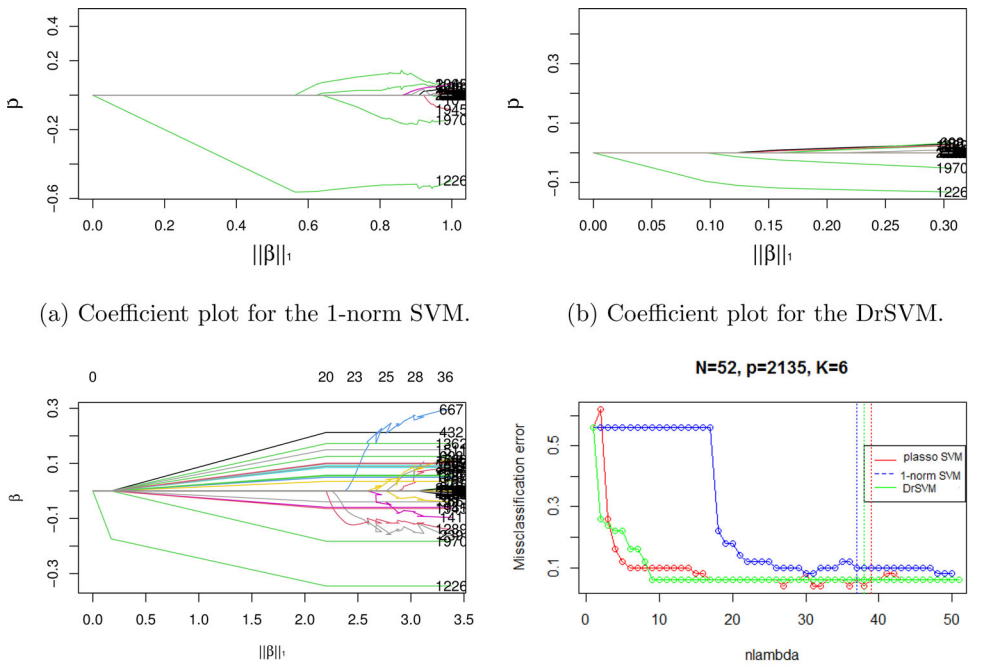
(b) Training, validation and testing plot for the Plasso-SVM

Figure 7. Cross validation plot and the combine plot for training, validating, and testing for the Plasso-SVM on the Colon cancer dataset.

from X. We compared the plasso-SVM with the 1-norm SVM and the DrSVM. The best model was chosen after the 5-fold cross-validation, then the final model is tested on the test data. The plasso-SVM obtained a misclassification error of 0.06 with 28 none zero coefficients, the 1-norm SVM had an error of 0.10 with 6 none zero coefficients and the DrSVM had 0.06 with 7 none zero coefficients. Figure 8 shows the coefficient plot for the three algorithms and the plot for the misclassification error.

5. Conclusion

We have studied the support vector machine for classification problems with main and interaction effects. The procedure involved the implementation of the pliable lasso penalty. This penalty allows for the estimation of the interactions that exist between the covariates X and the



(a) Coefficient plot for the 1-norm SVM.

(b) Coefficient plot for the DrSVM.

(c) Coefficient plot for the Plasso-SVM

(d) Test error plots for the three models

Figure 8. Coefficient plots and test error plots for the DrSVM, 1-Norm SVM and the Plasso-SVM for the prostate cancer dataset. The vertical lines in the test error plot shows the various model selected by the five-fold cross validation.

modifiers Z. The nature of pliable lasso helps to exclude the interaction terms when the corresponding main effects are zero.

Our algorithm involved the implementation of the squared hinge loss with the pliable lasso penalty, which then allowed us to apply the block-wise coordinate descent algorithm to optimize the objective function.

We implemented this result on a simulation as well on real data which included the colon cancer dataset and the prostate cancer dataset. From the results obtained we realized that one can implement the pliable lasso penalty on support vector machine to perform classification problems where there exist some interaction effects between the covariates and the set of modifiers.

Appendix A. Details of the optimization procedure

Given N data points of training set $\{(X_i, Z_i, y_i), i = 1, 2, ..., N\}$, with $X_i = (x_{i1}, x_{i2}, ..., x_{ip})$ representing the i^{th} input with dimension p, $Z_i = (z_{i1}, z_{i2}, ..., x_{iK})$ representing the i^{th} input for the modifying variables and $y_i \in \{-1, 1\}$, representing the class of the i^{th} categorical response, the objective function for the support vector machine with pliable lasso penalty is given as

$$M(\beta_0, \theta_0, \beta, \theta) = \frac{1}{N}\sum_{i=1}^{N}\left(\left[1 - y_i\left(\beta_0 + Z\theta_0 + \left(\sum_{j=1}^{p} X_{ij}\beta_j + W_j\theta_j\right)\right)\right]_+\right)^2 \times \left[(1-\alpha)\lambda\sum_{j=1}^{p}(||(\beta_j, \theta_j)||_2 + ||\theta_j||_2) + \alpha\lambda\sum_{j,k}|\theta_{j,k}|_1\right] \tag{A1}$$

The loss function looks like the least squared problem so it can be solved using the same idea from the Gaussian pliable lasso algorithm. We only optimize with the components for which $\{y_i(\beta_0 + Z\theta_0 + (\sum_{j=1}^{p} X_{ij}\beta_j + W_j\theta_j)) < 1\}$

and represent $\beta_0 + Z\theta_0 + (\sum_{j=1}^{p} X_{ij}\beta_j + W_j\theta_j)$ by $\bar{n}$. From (A1), we obtain the following equations as sub gradient with respect to θ_j and β_j;

$$\frac{\partial M}{\partial \beta_j} = \frac{1}{N}\sum_{i=1}^{N}(-x_{ij})(y_i)\left[1 - y_i\left(\beta_0 + Z\theta_0 + \left(\sum_{j=1}^{p} X_{ij}\beta_j + W_j\theta_j\right)\right)\right]_+ \tag{A2a}$$

$$+(1-\alpha)\lambda u \tag{A2b}$$

$$= \frac{-(yX_j)^T r}{N} + (1-\alpha)\lambda u = 0 \tag{A2c}$$

$$\frac{\partial M}{\partial \theta_j} = \frac{1}{N}\sum_{i=1}^{N}(-W_{jk})(y_i)\left[1 - y_i\left(\beta_0 + Z\theta_0 + \left(\sum_{j=1}^{p} X_{ij}\beta_j + W_j\theta_j\right)\right)\right]_+ \tag{A3a}$$

$$+(1-\alpha)\lambda(u_2 + u_3) + \alpha\lambda v \tag{A3b}$$

$$= \frac{-(yW_j)^T r}{N} + (1-\alpha)\lambda(u_2 + u_3) + \alpha\lambda v = 0 \tag{A3c}$$

Where in the above equations,

$$r_i = \left[1 - y_i\left(\beta_0 + Z\theta_0 + \left(\sum_{j=1}^{p} X_{ij}\beta_j + W_j\theta_j\right)\right)\right]_+ \tag{A4a}$$

$$u = \frac{\beta_j}{||\beta_j, \theta_j||_2}, \quad \text{if} \quad (\beta_j, \theta_j) \neq 0 \quad \text{and} \quad \in \{u : ||u||_2 \leq 1\} \tag{A4b}$$

$$\text{if} \quad (\beta_j, \theta_j) = 0 \tag{A4c}$$

$$u_2 = \frac{\theta_j}{||\beta_j, \theta_j||_2}, \quad \text{if} \quad (\beta_j, \theta_j) \neq 0 \quad \text{and} \quad \in \{u_2 : ||u_2||_2 \leq 1\} \tag{A4d}$$

$$\text{if} \quad (\beta_j, \theta_j) = 0 \tag{A4e}$$

$$u_3 = \frac{\theta_j}{||\theta_j||_2}, \quad \text{if} \quad \theta_j \neq 0 \quad \text{and} \quad \in \{u_3 : ||u_3||_2 \leq 1\} \quad \text{if} \quad (\theta_j) = 0 \tag{A4f}$$

$$v = sign(\theta_j) \tag{A4g}$$

We now state the screening condition which determines whether $(\hat{\beta}_j, \hat{\theta}_j) = 0$. Denote $r = 1 - y_i(\bar{n})$ since we are dealing with the components where $\{y_i(\beta_0 + Z\theta_0 + (\sum_{j=1}^{p} X_{ij}\beta_j + W_j\theta_j)) < 1\}$. Hence $r_{(-j)} = r - y_i(X_j\beta_j + W_j\theta_j)$ denotes the partial residual without the j^{th} group. The condition is that $(\hat{\beta}_j, \hat{\theta}_j) = 0$ if

$$|yX_j^T r_{(-j)}/N| \leq (1-\alpha)\lambda \quad \text{and} \quad ||S(yW_j^T r_{(-j)}/N, \alpha\lambda)||_2 \leq 2(1-\alpha)\lambda. \tag{A5}$$

Remark A.1. 1. By following the work of Asenso, Zhang, and Liang (2020), we obtain the maximum λ value as $\lambda_{\max} = \max|X^T(yr)|/N(1-\alpha)$. However, the parameter $\alpha \in (0,1)$ is always pre-selected.

If the screening condition is not satisfied, we proceed to check for the condition under which $\hat{\beta}_j \neq 0, \quad \hat{\theta}_j = 0$ by checking if

$$||S(y_i W_j^T(r_{(-j)} + y_i X_j\hat{\beta}_j)/n, \alpha\lambda)||_2 \leq (1-\alpha)\lambda, \tag{A6}$$

where

$$\hat{\beta}_j = \left(N \Big/ \sum_{i=1}^{N}(y_i^2(x_{ij})^2)\right)S(y_i X_j^T r_{(-j)}/n, (1-\alpha)\lambda) \tag{A7}$$

If both $\hat{\beta}_j \neq 0$ and $\hat{\theta}_j \neq 0$, take $\gamma_j = (\beta_j, \theta_j)$ and optimize the majorized objective function

$$\begin{aligned} M(\gamma_j) = l(\gamma_0) + (\gamma_j - \gamma_0)^T\nabla l(\gamma_{0j}) + \frac{1}{2t}||\gamma_j - \gamma_0||_2^2 \\ + (1-\alpha)\lambda(||\gamma_j||_2 + ||\theta_j||_2) + \alpha\lambda\sum_{j,k}|\theta_{j,k}|_1, \end{aligned} \tag{A8}$$

following the generalized gradient descent algorithm in (Tibshirani and Friedman 2020), with $l(\gamma_0) = \nabla(r_{(-j)}, \gamma_0) = (-yX_j^T r_{(-j)}/N, -yW_j^T r_{(-j)}/N)$ for the square error loss.

Algorithm 2: Algorithm for the SVM pliable lasso model (plasso-SVM)

Initialization;
for *Decrement* λ **do**
  Initialize set all $(\beta_0, \quad \beta_j) = 0$, $(\theta_0, \quad \theta_j) = 0 \quad (j = 1, 2, \ldots p)$
    1. compute $\hat{\bar{n}}$, r
    2. Compute $\hat{\beta}_0$ and $\hat{\theta}_0$. **NB:** $\hat{\beta}_0$ is computed by regressing $(r, 1)$. Also $\hat{\theta}_0$ is computed by regressing (r, Z).
  For $j \in \{1, 2, \ldots p, 1, 2, \ldots\}$ Check condition (A5) for $(\hat{\beta}_j, \hat{\theta}_j) = 0$.
  if $(\hat{\beta}_j, \hat{\theta}_j) = 0$ **then**
    skip to next *j*;
  else
    Compute $\hat{\beta}_j$ from (A7) and then check if $\hat{\theta}_j = 0$ from (A6);
  end
  if $\hat{\theta}_j = 0$ **then**
    Update β_j with the computed $\hat{\beta}_j$ and skip to next *j*.;
  else
    Repeat until convergence the generalized gradient decent algorithm part in Tibshirani and Friedman (2020) to compute $(\hat{\beta}_j, \hat{\theta}_j)$
  end
end

ORCID

Theophilus Quachie Asenso http://orcid.org/0000-0002-2717-1171

References

Alon, U., N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America* 96 (12):6745–50. doi:10.1073/pnas.96.12.6745.

Asenso, T. Q., H. Zhang, and Y. Liang. 2020. Pliable lasso for the multinomial logistic regression. *Communications in Statistics - Theory and Methods* 0 (0):1–16. doi:10.1080/03610926.2020.1800041.

Chang, K.-W., C.-J. Hsieh, and C.-J. Lin. 2008. Coordinate descent method for large-scale *l*2-loss linear support vector machines. *Journal of Machine Learning Research* 9:1369–98.

Du, W., and R. Tibshirani. 2018. A pliable lasso for the cox model.

Dua, D., and G. Casey. 2019. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences.

Friedman, J., T. Hastie, and R. Tibshirani. 2010. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* 33 (1):1–22. doi:10.18637/jss.v033.i01.

Glaab, E., J. Bacardit, J. M. Garibaldi, and N. Krasnogor. 2012. Using rule-based machine learning for candidate disease gene prioritization and sample classification of cancer gene expression data. *PLoS ONE* 7 (7):e39932.

Golub, T., D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, et al. 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression. *Science (New York, N.Y.)* 286 (5439):531–7. doi:10.1126/science.286.5439.531.

Hastie, T., S. Rosset, R. Tibshirani, and J. Zhu. 2004. The entire regularization path for the support vector machine. *Journal of Machine Learning Research* 5:1391–415.

Huo, Y., L. Xin, C. Kang, M. Wang, Q. Ma, and B. Yu. 2020. Sgl-svm: A novel method for tumor classification via support vector machine with sparse group lasso. *Journal of Theoretical Biology* 486 (110098):110098.

Lee, Y.-J., and O. L. Mangasarian. 2001. Ssvm: A smooth support vector machine for classification. *Computational Optimization and Applications* 20 (1):5–22. doi:10.1023/A:1011215321374.

Lim, M., and T. Hastie. 2015. Learning interactions via hierarchical group-lasso regularization. *Journal of Computational and Graphical Statistics: A Joint Publication of American Statistical Association, Institute of Mathematical Statistics, Interface Foundation of North America* 24 (3):627–54. PMID: 26759522.

Martinez, D. L. 2017. Regularization approaches for support vector machines with applications to biomedical data. *CoRR*. abs/1710.10600.

Platt, J. C. 1998. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, Advances in kernel methods – support vector learning.

Shevade, S., and S. Keerthi. 2003. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics (Oxford, England)* 19 (17):2246–53. doi:10.1093/bioinformatics/btg308.

Singh, D., P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, et al. 2002. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* 1 (2):203–9. doi:10.1016/S1535-6108(02)00030-2.

Tibshirani, R., and J. Friedman. 2020. A pliable lasso. *Journal of Computational and Graphical Statistics* 29 (1): 215–25. doi:10.1080/10618600.2019.1648271.

Vapnik, V. N. 1995. *The nature of statistical learning theory*. New York: Springer-Verlag.

Wang, L., J. Zhu, and H. Zou. 2006. The doubly regularized support vector machine. *Statistica Sinica* 16 (2): 589–616.

Wolberg, W. H., and O. L. Mangasarian. 1990. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences of the United States of America* 87 (23):9193–6. doi:10.1073/pnas.87.23.9193.

Wu, S., H. Zou, and M. Yuan. 2008. Structured variable selection in support vector machines. *Electronic Journal of Statistics* 2 (none):103–17. doi:10.1214/07-EJS125.

Zhu, J., S. Rosset, T. Hastie, and R. Tibshirani. 2003. 1-norm support vector machines. In *Neural information processing systems*, 16. Cambridge, MA: MIT Press.