2010 International Conference on Technologies and Applications of Artificial Intelligence

Smooth LASSO for Classification

Li-Jen Chien
Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, 106 Taiwan
D8815002@mail.ntust.edu.tw

Zhi-Peng Kao
Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, 106 Taiwan
M9515040@mail.ntust.edu.tw

Yuh-Jye Lee
Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, 106 Taiwan
yuh-jye@mail.ntust.edu.tw

***Abstract*—The sparse model character of 1-norm penalty term of Least Absolute Shrinkage and Selection Operator (LASSO) can be applied to automatic feature selection. Since 1-norm SVM is also designed with 1-norm (LASSO) penalty term, this study labels it as LASSO for classification. This paper introduces the smooth technique into 1-norm SVM and calls it smooth LASSO for classification (SLASSO) to provide simultaneous classification and feature selection. In the experiments, we compare SLASSO with other approaches of "wrapper" and "filter" models for feature selection. Results showed that SLASSO has slightly better accuracy than other approaches with the desirable ability of feature suppression.**

***Keywords*-classification; feature selection; least absolute shrinkage and selection operator; smooth technique; support vector machine;**

I. Introduction

This paper focuses on the feature selection problem in the support vector machine for binary classification. Feature suppression is very important in the development of new techniques in bioinformatics that utilize gene microarrays [18] for prognostic classification, drug discovery, and other tasks. Many studies use 2-norm SVM for solving the classification problems. However, 2-norm SVM classifier can not automatically select input features. 1-norm SVM can automatically discard irrelevant features by estimating corresponding variables by zero. Thus, 1-norm SVM is both a wrapper method [13] and an automatic relevance determination (ARD) model [22]. When there are many noise features in training set, 1-norm SVM has significant advantages over 2-norm SVM. Methods for simultaneous classification and feature selection have grown popularly. In the past few years, 1-norm penalty term for feature selection has attracted a lot of attention. Tibshirani [23] proposed LASSO as a colorful name via using 1-norm penalty for feature selection in the regression problem. Osborne et al. [20] made compact descent and homotopy method to computational problems associated with implementing LASSO. Zhu et al. [26], Mangasarian [16] and Zou [27] used 1-norm SVM to attain the goal of automatic feature selection in the classification problem. In the above studies, 1-norm penalty is able to cause most coefficients to be exactly zero and generate sparse solutions especially in the high dimensional feature space.

In fact, LASSO adds 1-norm regularization into the objective function to control the model complexity. The LASSO model typically has many zero elements and thus shares characteristics of both shrinkage variable and feature selection for regression and classification problems. LASSO is piece-wise linear and not differential. Researchers can transform LASSO into a linear inequalities problem, but the number of these inequalities can be large and the number of variables is doubled. The goal of LASSO is to handle the problems that the number of features is larger than the number of data points and attains automatic feature selection. Consequently, we would not like to increase the number of variables. For these reasons, this study proposes smooth LASSO for classification (SLASSO).

The basic idea of SLASSO is to convert 1-norm SVM [26] problem into a non-smooth unconstrained minimization problem, and then use standard smoothing techniques of mathematical programming [6], [7], [15] to smooth this unconstrained minimization problem. The Newton-Armijo Algorithm can be used for solving SLASSO since the problem is infinitely differentiable. Experiments test SLASSO on some benchmark datasets. Results show that SLASSO has slightly better accuracy than other approaches with the desirable ability of feature suppression.

The following briefly describe some notations used in this paper. For notational convenience, the training dataset is rearranged as an $m \times n$ matrix A, and $A_i = (x^i)'$ corresponds to the ith row of A. Column vectors of ones and zeros are denoted by bold $\mathbf{1}$ and $\mathbf{0}$ respectively. For a vector $x \in R^n$, the plus function x_+ is defined as $(x_+)_i = \max\{0, x_i\}$, $i = 1, \ldots, n$. For a vector $v \in R^m$, the $diag(v)$ is an $m \times m$ diagonal matrix with vector v along its diagonal. This operator on v is available in MATLAB [17]. For $x \in R^n$ and $1 \leq p < \infty$, the p-norm will be

978-0-7695-4253-9/10 $26.00 © 2010 IEEE
DOI 10.1109/TAAI.2010.48

IEEE computer society

denoted as $\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}$. If f is a real valued function defined on the n-dimensional real space R^n, the gradient of f at x is denoted by $\nabla f(x)$ which is a row vector in R^n and the $n \times n$ Hessian matrix of second partial derivatives of f at x is denoted by $\nabla^2 f(x)$.

The rest of this paper is organized as follows. Section II provides a brief introduction of 1-norm SVM (LASSO for classification). Section III presents the formulation of the smooth LASSO. Section IV describes Newton-Armijo algorithm and implementation of smooth LASSO. Section V presents the numerical results. Section VI concludes the study.

II. LASSO and 1-norm Support Vector Machine

Regularization term is usually added in objective function for the purpose of coefficient shrinkage. 1-norm and 2-norm regularization term are mostly used. Ridge regression uses 2-norm regularization term as follows:

$$\hat{\beta}^{ridge} = \arg\min ||Y - \beta X||_2^2 + \lambda ||\beta||_2^2, \tag{1}$$

$\lambda \geq 0$ is the shrinkage parameter. Unlike ridge regression, LASSO uses 1-norm regularization term as follows:

$$\hat{\beta}^{LASSO} = \arg\min ||Y - \beta X||_2^2 + \lambda ||\beta||_1. \tag{2}$$

Many studies [22], [23] indicated that 1-norm regularization is better than 2-norm regularization term in coefficient shrinkage. The problem (3) is 1-norm SVM formulation [16], [26] and it can also generate a very sparse model used in feature selection.

$$\begin{array}{rlcl} \min\limits_{(w,b,\xi) \in R^{(n+1+m)}} & \|w\|_1 + C\|\xi\|_1 & & \\ subject\ to: & D(Aw + \mathbf{1}b) + \xi & \geq & \mathbf{1} \\ & \xi & \geq & \mathbf{0}. \end{array} \tag{3}$$

The main advantages of 1-norm SVM are very effective in reducing input space features for linear kernels and the number of kernel functions for nonlinear SVM [16]. Unlike 1-norm SVM is used for classification problems, LASSO is originally used for regression problems. However, LASSO is a colorful name for feature selection method with 1-norm penalty term. No matter how the purposes are different, this study treats LASSO as the property of the methods with an 1-norm penalty term. This study names 1-norm SVM as LASSO for classification through the following sections. Section III combines the strategy of SSVMs [7], [15] and LASSO for classification to propose smooth LASSO for classification.

III. SLASSO

Following the SSVM methodology [15] of SSVM$_1$ [7] and SSVM$_2$ [15] , the absolute b is appended to the objective function of problem (3). Thus, the original $\|w\|_1$ in the objective function of problem (3) is replaced by $\|(w, b)\|_1$ as follows:

$$\begin{array}{rl} \min\limits_{(w,b,\xi) \in R^{(n+1+m)}} & \|w\|_1 + |b| + C\|\xi\|_1 \\ subject\ to: & D(Aw + \mathbf{1}b) + \xi \geq \mathbf{1} \\ & \xi \geq \mathbf{0}. \end{array} \tag{4}$$

Then, problem (4) can be converted to an explicit linear program as follows:

$$\begin{array}{rl} \min\limits_{(\widetilde{w},\xi) \in R^{(n+1+m)}} & \mathbf{1}'((\widetilde{w})_+ + (-\widetilde{w})_+) + C\mathbf{1}'\xi \\ subject\ to: & D\widetilde{A}((\widetilde{w})_+ - (-\widetilde{w})_+) + \xi \geq \mathbf{1} \\ & \xi \geq \mathbf{0}, \end{array} \tag{5}$$

where the following substitution for $\widetilde{w}$ and $\widetilde{A}$ have been made:

$$\widetilde{w} = \begin{bmatrix} w \\ b \end{bmatrix}, \quad \widetilde{A} = \begin{bmatrix} A & \mathbf{1} \end{bmatrix}.$$

The slack variable ξ in the objective function of problem (5) is replaced by $(\mathbf{1} - D\widetilde{A}((\widetilde{w})_+ - (-\widetilde{w})_+))_+$. Hence, problem (5) is converted into an unconstrained optimization problem as follows:

$$\begin{array}{rl} \min\limits_{\widetilde{w} \in R^{n+1}} & \mathbf{1}^T((\widetilde{w})_+ + (-\widetilde{w})_+) + \\ & C\mathbf{1}^T(\mathbf{1} - D\widetilde{A}((\widetilde{w})_+ - (-\widetilde{w})_+))_+. \end{array} \tag{6}$$

Obviously, the objective function in problem (6) is not twice differentiable so that Newton method can not be applied to solve the problem. Therefore, SLASSO on classification employs a smoothing function [5] to replace the original plus function. In SSVM, the plus function x_+ is approximated by a smooth $p\text{-}function$, $p(x, \alpha) = x + \frac{1}{\alpha}\log(1 + e^{-\alpha x}), \alpha > 0$. Note that if the value of α increases, the $p(x, \alpha)$ will approximate the plus function more accurately. Next, the $p(x, \alpha)$ is taken into problem (6) to replace the plus function as following:

$$\begin{array}{rl} \min\limits_{\widetilde{w} \in R^{n+1}} & \mathbf{1}^T(p(\widetilde{w}, \alpha) + p(-\widetilde{w}, \alpha)) + \\ & C\mathbf{1}^T p(\mathbf{1} - D\widetilde{A}(p(\widetilde{w}, \alpha) - p(-\widetilde{w}, \alpha)), \alpha). \end{array} \tag{7}$$

Thus, the objective function in problem (7) is twice differentiable and can be solved using a fast Newton method. However, Newton method might lead to the oscillation phenomenon. To avoid the phenomenon, the Armijo stepsize is employed to make the solution convergent globally. Section IV describes a Newton-Armijo algorithm for solving the smooth problem (7).

IV. Newton-Armijo Algorithm for SLASSO on classification

Taking the advantage of the twice differentiability of the objective function of problem (7), this section prescribe a Newton algorithm with an Armijo stepsize [4] that makes the algorithm globally convergent. Before introducing the

Newton-Armijo Algorithm of SLASSO for classification, we denote $\Phi_\alpha(\widetilde{w})$ to represent the objective function (7) for convenience. The Newton-Armijo Algorithm for solving problem (7) is as follows:

Newton-Armijo Algorithm of SLASSO for classification: Set the parameter values C and δ while C and δ are set by a tuning procedure. Start with any $(\widetilde{w}) \in R^{n+1}$. Having $(\widetilde{w}^i)$, stop if the gradient of the objective function of (7) is zero, that is $\nabla\Phi_\alpha(\widetilde{w}^i) = 0$. Else compute $(\widetilde{w}^{i+1})$ as follows:

(i) **Newton Direction**: Determine direction $d^i \in R^{n+1}$ by setting equal to zero the linearization of $\nabla\Phi_\alpha(\widetilde{w})$ around $(\widetilde{w}^i)$ which gives $n+1$ linear equations in $n+1$ variables:

$$(\nabla^2\Phi_\alpha(\widetilde{w}^i) + \delta I)d^i = -\nabla\Phi_\alpha(\widetilde{w}^i)^T. \qquad (8)$$

(ii) **Armijo Stepsize** [2]: Choose a stepsize $\lambda_i \in R$ such that:

$$\widetilde{w}^{i+1} = \widetilde{w}^i + \lambda_i d^i \qquad (9)$$

where $\lambda_i = \max\{1, \frac{1}{2}, \frac{1}{4}, \ldots\}$ such that :

$$\Phi_\alpha(\widetilde{w}^i) - \Phi_\alpha((\widetilde{w}^i) + \lambda_i d^i) \geq -\eta\lambda_i\nabla\Phi_\alpha(\widetilde{w}^i)d^i \qquad (10)$$

where $\eta \in (0, \frac{1}{2})$.

Note that selecting a good initial solution for the Newton-Armijo algorithm can speed up the time for finding the optimal unique solution. Furthermore, the proposed smoothing algorithm is globally convergent to a unique solution. Section V describes the implementation details.

We describe how to implement the Newton-Armijo algorithm of SLASSO for classification using MATLAB code. The most important ingredients of the Newton-Armijo algorithm are the gradient and Hessian matrix of $\Phi_\alpha(\widetilde{w})$ in (7). Then we apply the chain rule from calculus and some elementary matrix algebra to get the following formula:

$$\begin{aligned}\nabla\Phi_\alpha(\widetilde{\omega}) = \;& p^{'}(\widetilde{\omega},\alpha) - p^{'}(-\widetilde{\omega},\alpha) - Cdiag(p^{'}(\widetilde{\omega},\alpha)+\\ & p^{'}(-\widetilde{\omega},\alpha))\widetilde{A}^T Dp^{'}(s,\alpha)\end{aligned} \qquad (11)$$

and

$$\begin{aligned}\nabla^2\Phi_\alpha(\widetilde{\omega}) = \;& diag(p^{''}(\widetilde{\omega},\alpha) + p^{''}(-\widetilde{\omega},\alpha)) -\\ & Cdiag(diag(p^{''}(\widetilde{\omega},\alpha) -\\ & p^{''}(-\widetilde{\omega},\alpha))\widetilde{A}^T Dp^{'}(s,\alpha)) +\\ & Cdiag(p^{'}(\widetilde{\omega},\alpha) +\\ & p^{'}(-\widetilde{\omega},\alpha))\widetilde{A}^T diag(p^{''}(s,\alpha))\widetilde{A}diag(p^{'}(\widetilde{\omega},\alpha) +\\ & p^{'}(-\widetilde{\omega},\alpha))\end{aligned} \qquad (12)$$

where

$$s = \mathbf{1} - D\widetilde{A}(p(\widetilde{w},\alpha) - p(-\widetilde{w},\alpha)). \qquad (13)$$

For data matrix $\widetilde{A} \in R^{m\times(n+1)}$, this study provides two different ways to calculate the direction d^i in the Newton iteration (9). For this purpose define:

$$\begin{aligned}U :=\;& \delta I + diag(p^{''}(\widetilde{\omega},\alpha) + p^{''}(-\widetilde{\omega},\alpha))\\ & -Cdiag(diag(p^{''}(\widetilde{\omega},\alpha) - p^{''}(-\widetilde{\omega},\alpha))\widetilde{A}^T Dp^{'}(s,\alpha)),\\ E :=\;& diag(p^{'}(\widetilde{\omega},\alpha) + p^{'}(-\widetilde{\omega},\alpha))\widetilde{A}^T sqrt(Cdiag(p^{''}(s,\alpha))).\end{aligned} \qquad (14)$$

Then, it follows from (14) that:

$$\nabla^2\Phi_\alpha(\widetilde{w}^i) + \delta I = EE^T + U,$$

which is the matrix whose inverse is needed in the Newton iteration (8).

We use $(EE^T + U)^{-1}$ directly for the case $m \gg n$. For $m \ll n$ case, we apply the Sherman-Morrison-Woodbury identity [10] as follows:

$$(EE^T + U)^{-1} = U^{-1} - U^{-1}E(I + E^TU^{-1}E)^{-1}E^TU^{-1}.$$

Note that an $m \times m$ linear system of equations instead of an $(n+1)\times(n+1)$ makes the proposed algorithm very fast when $m \ll n$ but m is relatively small and the inverse U^{-1} of U is trivial to calculate since U is a diagonal matrix.

V. Experimental Tests

A. Data Presentation and Experimental Setup

The computational configuration was a P4 2.8GHz computer with 1GB of memory and Windows XP operating system, which Matlab 7.0 is installed. The range of tuning parameters C and δ sets $[10^{-2}, 10^4]$ and $[10^{-3}, 10^3]$ respectively. Statistics and descriptions of datasets are as follows:

1) Acute Leukemia Dataset: In the acute leukemia dataset [11], there are 25 acute myeloid leukemia (AML) and 47 acute lymphoblastic leukemia (ALL) samples which are taken from 72 patients. Each sample has 7129 genes obtained from microarray experiments. In ALL class, the 47 samples are further grouped into 38 B-lineage cell ALL (B-Cell ALL) and 9 T-lineage cell ALL (T-Cell ALL) samples. The acute leukemia dataset contains a training set and an independent test set. The training set has 38 samples which include 11 AML and 27 ALL samples (19 B-Cell, 8 T-Cell). There are 34 samples in the test set which consist of 14 AML and 20 ALL samples (19 B-Cell, 1 T-Cell). The summary of this data set is shown in Table I. Since the acute leukemia data set contains three categories, this study converts this trinary classification problem into two binary classification problems in the experiments. One is used to distinguish AML form ALL and another is used to classify B-Cell ALL and T-Cell ALL. By verifying parameters of stratified 5-fold cross validation on training set, we build up the model for whole training set and directly report the number of genes and testing accuracy of testing set in Table III.

Table I
Summary of the acute leukemia microarray gene expression dataset

Acute Leukemia Dataset (72 × 7219)			
	Training	Test	Total
AML	11	14	25
B-Cell ALL	19	19	38
T-Cell ALL	8	1	9
Total	38	34	72
# Genes	7129		

2) Colon Cancer Dataset: Microarray gene expression values for 22 normal and 40 colon cancer tissues are collected. Each sample has 6500 genes which are obtained from microarray experiments. The colon cancer dataset [1] collected 2000 genes with the highest minimal intensity across the 62 tissues. By verifying parameters of stratified 5-fold cross validation, we report the number of genes and average testing accuracy over the accuracies of stratified 5-fold data in Table IV.

3) Multiple Myeloma Dataset: Multiple myeloma dataset is available at: http://myeloma.uams.edu/research/, and processed by David Page and his colleagues [21]. Multiple myeloma is characterized by malignant plasma cells that reproduce uncontrollably. The Plasma cells are a type of white blood cell that produces and secretes antigen-specific antibodies. Multiple myeloma plasma cells tend to localize within the bone marrow, although they may be found in other parts of the body as well. In the multiple myeloma dataset, there are 74 myeloma patients and 31 healthy donors. Each sample has 7008 genes obtained from the patients using plasma cells. There are two measurements in each one of the 7008 genes which are called the average difference (AD) and absolute call (AC) respectively. The AD is a floating point number, so we do not handle anything to utilize the classifier which requires an input of real numbers. The AC is one of three nominal values: A (Absent), M (Marginal) or P (Present). Thus, each nominal value is mapped into a three dimensional binary vector. The A, M and P are mapped into (001), (010) and (001) respectively. The AC feature space is transformed form a 7008-dimensional space into a $7008 \times 3 = 21024$ real-valued dimensional space. Clearly, the AD and AC are combined to a $21024 + 7008 = 28032$ real-valued dimensional space. A detailed description of multiple myeloma dataset can refer to [21]. By verifying parameters of stratified 10-fold cross validation, we report the *leave-one-out correctness (looc)*, total running times, average number of features per fold and the overall number of different features for NLPSVM [9], LPNewtonSVM [16] and SLASSO later in Table II.

4) Seven Other Datasets: There are six UCI Machine Learning Repository [3] datasets: Ionosphere, BUPA Liver, Pima Indians, Cleveland Heart Problem, Housing, and WDBC in the comparisons. Another dataset involved is the Galaxy Dim dataset [19]. By verifying parameters of stratified 10-fold cross validation, Results report the average time, training/testing accuracy and features over the accuracies of stratified 10-fold data in Table V.

B. Numerical Results and Comparisons

In the acute leukemia dataset, we reported the numbers of the selected genes and the misclassified samples in Table II. The previous results done by [11], [25], [12], [14] are included for comparison purpose. We also list the results with LIBLINEAR [8]. For discriminating AML samples from ALL samples, Guyon [12] and IFFS [14] select 8 genes and 14 genes, respectively, and SLASSO select 8 genes and has 1 misclassified sample. However, NLPSVM [9] and LPNewtonSVM [16] is not suitable for this situation. For distinguishing T-Cell ALL samples from B-Cell ALL samples, Weston [25], the weight score approach, IFFS [14], NLPSVM [9], LPNewtonSVM [16], and SLASSO select 5 genes, 20 genes, 9 genes, 3 genes, 2 genes, and 4 genes, respectively. SLASSO has the desirable performance.

Table II
The numerical results of the acute leukemia dataset

Acute Leukemia Dataset (72 × 7129) (Tested by independent test samples)				
Method	ALL/AML		B-Cell/T-Cell	
	# Genes	Errors	# Genes	Errors
Golub [11]	50	2	N/A	N/A
Weston [25]	20	0	5	0
Guyon [12]	8	0	N/A	N/A
Weight Score Approach [14]	10	1	20	0
IFFS [14]	14	0	9	0
LIBLINEAR [8]	12	2	5	0
NLPSVM [9]	4	3	3	0
LPNewtonSVM [16]	3	6	2	0
SLASSO	8	1	4	0

N/A Denotes “Not Available”

In the colon cancer dataset, SLASSO selects 3.8 genes and has a misclassified sample in stratified 5-fold cross validation test sets averages. SLASSO has the satisfying result. We summarized these results in Table III. The previous results done by [14], [24], [25] and result of LIBLINEAR [8] are also included.

In the multiple myeloma dataset, We report the *leave-one-out correctness (looc)*, total running times, average number of features per fold, and the overall number of different features used in the 105 folds of testing in Table IV. Best results is represented in bold. SLASSO has the best looc performance.

On seven other datasets, we report training and testing correctness and number of features which are all averages over ten folds in Table V. The column Time is the total time over ten folds. Best results are in bold. For the feature

Table III
The numerical results of the colon cancer dataset

Colon Cancer Dataset (62×2000) (Tested by stratified 5-fold cross validation)		
Method	Tumor/Normal	
	# Genes	Errors
Weston [25]	15	1.5
Guyon [12]	8	3
Weston [24]	20	1.7
Weight Score Approach [14]	20	1.6
IFFS [14]	5	1.4
LIBLINEAR [8]	21.2	1.8
NLPSVM [9]	5.8	2
LPNewtonSVM [16]	2.4	2.6
SLASSO	3.8	1

Table IV
The numerical results of the Multiple Myeloma dataset

Dataset m × n	NLPSVM looc Time (Sec.) Avg. Features Overall Features	LPNewtonSVM looc Time (Sec.) Avg. Features Overall Features	SLASSO looc Time (Sec.) Avg. Features Overall Features
Myeloma			
105×28032	87.62 %	85.71 %	**100** %
	244.22	**232.95**	545.31
	11.686	**3.143**	6.981
	17	**4**	8

suppression, SLASSO performs well with slightly better accuracy.

VI. Conclusion

This paper proposes Smooth Least Absolute Shrinkage and Selection Operator (SLASSO) to solve 1-norm penalty problems and the results showed that it can select features automatically and effectively. Inspired by SSVM_1, this study uses the same smooth methodology to solve LASSO for classification. When it encounters a problem with large numbers of features, this study applies the Sherman-Morrison-Woodbury identity [10] to decrease the training time. In the classification testing of SLASSO, this study compares SLASSO with other approaches of "wrapper" and "filter" models for feature selection. Results showed that SLASSO has slightly better accuracy than other approaches and performs well in feature suppression.

Table V
The numerical results of seven other datasets

Data set/Size	Algorithm	Time	Train%	Test%	Features
Ionosphere	NLPSVM	2.266	**88.8**	87.65	9
351 × 34	LPNewton	2.047	86.85	86.76	9
	SLASSO	**1.875**	87.22	**87.94**	**8.3**
BUPA Liver	NLPSVM	1.313	69.65	69.12	5.8
345 × 6	LPNewton	**0.141**	69.39	68.24	**5.1**
	SLASSO	2.25	**70.06**	**68.82**	5.8
Pima Indians	NLPSVM	2.328	**77.28**	77.11	5.5
768 × 8	LPNewton	**2.094**	76.76	76.32	4.4
	SLASSO	2.422	75.32	**78.03**	**4.3**
Cleveland	NLPSVM	**0.297**	**84.46**	84.14	7.3
296 × 13	LPNewton	1.391	83.03	84.14	7.6
	SLASSO	2.063	84.57	**84.48**	**7.2**
Housing	NLPSVM	**1.766**	**86.78**	85.2	8.7
506 × 13	LPNewton	1.875	85.9	84.4	7
	SLASSO	2.521	85.53	**85.4**	**6.7**
WDBC	NLPSVM	**2.531**	**97.23**	95.89	8.7
569 × 30	LPNewton	2.563	95.07	95.36	9
	SLASSO	5.375	96.97	**96.43**	**8.2**
Galaxy Dim.	NLPSVM	10.52	**94.8**	**94.61**	**5.7**
4192 × 14	LPNewton	**1.109**	94.64	94.44	**5.7**
	SLASSO	9.516	93.97	94.08	6

References

[1] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Cell Biology*, 96:6745–6750, 1999.

[2] L. Armijo. Minimization of functions having Lipschitz-continuous first partial derivatives. *Pacific Journal of Mathematics*, 16:1–3, 1966.

[3] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007. http://www.ics.uci.edu/ mlearn/MLRepository.html.

[4] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, second edition, 1999.

[5] C. Chen and O. L. Mangasarian. A class of smoothing functions for nonlinear and mixed complementarity problems. *Computational Optimization and Applications*, 5(2):97–138, 1996.

[6] X. Chen and Y. Ye. On homotopy-smoothing methods for variational inequalities. *SIAM Journal on Control and Optimization*, 37:589–616, 1999.

[7] L. J. Chien, Y. J. Lee, Z. P. Kao, and C. C. Chang. Robust 1-norm soft margin smooth support vector machine. In *Proc. of the 11th International Conference on Intelligent Data Engineering and Automated Learning*, 2010 (to appear).

[8] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[9] G. M. Fung and O. L. Mangasarian. A feature selection newton method for support vector machine classification. *Computational Optimization and Applications*, 28:185–202, 2004.

[10] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, Maryland, 3rd edition, 1996.

[11] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439), 1999.

[12] I. Guyon, J. Watson, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1/2/3):191–202, 2002.

[13] R. Kohavi and G. John. Wrapper fo feature subset selection. *Artificial Intelligent Journal*, pages 273–324, 1997.

[14] Y. J. Lee, C. C. Chang, and C. H. Chao. Incremental forward feature selection with application to microarray gene expression. *Journal of Biopharmaceutical Statistics*, 18(5):827–840, 2008.

[15] Y. J. Lee and O. L. Mangasarian. SSVM: A smooth support vector machine. *Computational Optimization and Applications*, 20:5–22, 2001.

[16] O. L. Mangasarian. Exact 1-norm support vector machines via unconstrained convex differentialble minimization. *Journal of Machine Learning Research*, 7:1517–1530, 2006.

[17] MATLAB. *User's Guide*. The MathWorks, Inc., Natick, MA 01760, 1994–2001.

[18] M. Molla, M. Waddell, D. Page, and J. Shavlik. Using machine learning to design and interpret gene-expression microarrays. *AI Magazine, Special Issue on Bioinformatics*, 2003.

[19] S. Odewahn, E. Stockwell, R. Pennington, R. Humphreys, and W. Zumach. Automated star/galaxy discrimination with neural networks. *Astronomical Journal*, 103(1):318–331, 1992.

[20] M. R. Osborne, B. Presnell, and B. A. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20:389–404, 2000.

[21] D. Page, F. Zhan, J. Cussens, M. Waddell, J. Hardin, B. Barlogie, and J. Shaughnessy. Comparative data mining for microarrays: A case study based on multiple myeloma. Technical report, Computer Sciences Department, University of Wisconsin, November 2002.

[22] V. Roth. The generalized lasso. *IEEE Transactions on Neural Networks*, 15(1):16–28, 2004.

[23] R. Tibshirani. Regression shrinkage and selection via the lasso. *J.R.S.S.B*, 58(1):267–288, 1996.

[24] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping. Use of the zero-norm with linear models and kernel methods. *Jorunal of Machine Learning Research*, 3:1439–1461, 2003.

[25] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In *Advances in Neural Information Processing Systems 13*, pages 668–674, 2001.

[26] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm support vector machines. In *Advances in Neural Information Processing Systems 16-NIPS2003*. MIT Press, 2003.

[27] H. Zou. An improved 1-norm svm for simultaneous classification and variable selection. *Eleventh International Conference on Artificial Intelligence and Statistics*, 2007.