Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 4 (2011) 1691–1698

Procedia Computer Science

International Conference on Computational Science, ICCS 2011

AUC Maximizing Support Vector Machines with Feature Selection

Yingjie Tian, Yong Shi

Research Center on Fictitious Economy and Data Science, CAS, Beijing, China

Xiaojun Chen

Department of Applied Mathematics, The Hong Kong Polytechnic University, Kowloon, Hong Kong

Wenjing Chen[1,*]

Research Center on Fictitious Economy and Data Science, CAS, Beijing, China

Abstract

In this paper, we proposed a new algorithm, the Sparse AUC maximizing support vector machine, to get more sparse features and higher AUC than standard SVM. By applying p-norm where $0 < p < 1$ to the weight w of the separating hyperplane $(w \cdot x) + b = 0$, the new algorithm can delete less important features corresponding to smaller $|w|$. Besides, by applying the AUC maximizing objective function, the algorithm can get higher AUC which make the decision function have higher prediction ability. Experiments demonstrate the new algorithm's effectiveness. Some contributions as follows: (1) the algorithm optimizes AUC instead of accuracy; (2) incorporating feature selection into the classification process; (3) conduct experiments to demonstrate the performance.

Keywords: Support Vector Machine, AUC, feature selection, p-norm

1. Introduction

Support vector machine (SVM) [1, 2] has been a promising tool in machine learning [3, 4]. However, its success depends upon the tuning of several parameters which affect generalization error. These parameters include such as kernels, penalty parameters, the number of features and etc. For example, when given a training set, a practitioner must first select a subset of features which contribute most to the problem, and then choose those parameters in SVM algorithms leading to good generalization performance. An effective approach is to estimate the generalization error and then search for parameters so that this estimator is minimized. In other words, we should give out an estimation of a generalization error or some other related measures [5, 6] and then evaluate a learned model. This requires that the estimators are both effective and computationally efficient.

[*] *Email address:* whchen1985@126.com,Corresponding author (Wenjing Chen)

[1] This work has been partially supported by grants from National Natural Science Foundation of China(NO. 10601064, NO. 70921061),HK PolyU grant (B-Q10D), and the President Fund of GUCAS

Selection and/or peer-review under responsibility of Prof. Mitsuhisa Sato and Prof. Satoshi Matsuoka
doi:10.1016/j.procs.2011.04.183

Nowadays the area under the receiver operating characteristics (ROC) curve, which corresponds to the Wilcoxon-Mann-Whitney test statistic, is increasingly used as a performance measure for classification systems, especially when one often has to deal with imbalanced class priors or misclassification costs[7, 8]. The ROC curve details the rate of *true positives* against *false positives* over the range of possible threshold values. The area of that curve is the probability that a randomly drawn positive example has a higher decision function value than a random negative example; it is called the AUC (area under ROC curve). [9] established formal criteria (consistency and discriminancy) for comparing AUC and accuracy for learning algorithms and show theoretically and empirically that AUC is a better measure than accuracy, therefore we should use learning algorithms to optimize AUC instead of accuracy.

When the goal of a learning problem is to find a decision function with high AUC value, then it is natural to use a learning algorithm that directly maximizes this criterion. Over the last years, AUC maximizing versions of several learning algorithms have been developed[10]-[16].

However, recent AUC maximizing algorithms does not directly obtain the feature importance. The benefit of feature selection is twofold. It leads to parsimonious models that are often preferred in many scientific problems, and it is also crucial for achieving good classification accuracy in the presence of redundant features[17, 18]. We can combine SVM with various feature selection strategies, Some of them are "filters": general feature selection methods independent of SVM. That is, these methods select important features first and then SVM is applied for classification. On the other hand, some are wrapper-type methods: modifications of SVM which choose important features as well as conduct training/testing. In the machine learning literature, there are several proposals for feature selection to accomplish the goal of automatic feature selection in the SVM[18]-[23], in some of which they applied the l_0-norm, l_1-norm or l_∞-norm SVM and got competitive performance. The interesting one is l_1-norm SVM, where the 2-norm vector w of the objective function is replaced by 1-norm in the standard SVM model. Furthermore, we observe that l_p-norm SVM leads to more sparse solution when p norm is reduced from 2-norm to 1-norm and the more spare solutions when p $(0 < p < 1)$ is decreased further[24, 25].

Therefore, in this paper we will combine AUC maximizing SVM with feature selection via $l_p(0 < p < 1)$-norm. Section 2 will introduce the AUC maximizing SVM and Sparse AUC maximizing SVM is proposed in Section 3, Numerical experiments are conducted in Section 4, Section 5 gives out the conclusions.

2. AUC maximizing SVM

2.1. ROC curve and AUC

For a classification problem, the training set is given by

$$T = \{(x_1^+, 1), ..., (x_{l^+}^+, 1), (x_1^-, -1), \cdots, (x_{l^-}^-, -1)\} \in (R^n \times \{-1, 1\})^l, \tag{1}$$

where $l = l^+ + l^-$. Consider the decision function is on the form $y = \mathrm{sgn}(f(x)) = \mathrm{sgn}((w \cdot x) + b)$, where $y = 1$ if $f(x) \geq 0$, or $y = -1$ else.

ROC curve is a two-dimensional measure of classification performance $f(x)$. It plots the number of true positives on the y-axis against the number of false positives on the x-axis. One of the most interesting point of ROC curve is that if error costs or class distributions are unknown, classifier performance can still be characterized and optimized. Figure 1 [15] depicts an example of the ROC curve of a given classifier. The diagonal line corresponds to the ROC curve of a classifier that predicts the class at random and the performance improves the further the curve is near to the upper left corner of the plot.

The area under the curve, commonly denoted as AUC, is the most frequently used performance measure extracted from the ROC curve. AUC equals to the probability that $f(x)$ assigns a higher value to a randomly drawn positive input x^+ than to a randomly drawn negative input x^-,

$$AUC(f) = Pr(f(x^+) > f(x^-)). \tag{2}$$

When AUC is equal to 1, the classifier achieves perfect accuracy if the threshold is correctly chosen, and a classifier that predicts the class at random has an associated AUC of 0.5. The AUC refers to the true distributions of positive and negative points, but it can be estimated using the training set T

$$AUC = \frac{\sum_{i=1}^{l^+} \sum_{i=1}^{l^-} 1_{(f(x_i^+) > f(x_i^-))}}{l^+ l^-}, \tag{3}$$

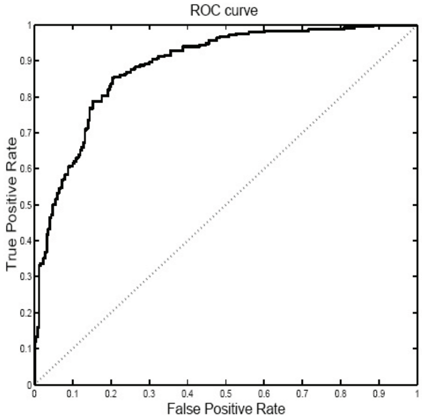

Figure 1: Example of ROC curve . The diagonal line denotes the ROC curve of a random classifier.

where 1_π is defined to be 1 if the predicate π holds and 0 otherwise. Maximizing the AUC is therefore equivalent to maximizing the number of pairs satisfying $f(x_i^+) > f(x_i^-)$.

2.2. AUC maximizing SVM

For the linear decision function with the form $f(x) = (w \cdot x) + b$, AUC turns to

$$AUC = \frac{\sum_{i=1}^{l^+} \sum_{i=1}^{l^-} 1_{(w \cdot (x_i^+ - x_i^-)) > 0}}{l^+ l^-}, \tag{4}$$

if we denote $\xi_{ij} = (w \cdot (x_i^+ - x_i^-))$, then initial AUC Maximizing problem should be [15]

$$\max_{w,\xi} \quad g(\xi) = \sum_{i=1}^{l^+} \sum_{i=1}^{l^-} \Theta(\xi_{ij}), \tag{5}$$

$$\text{s.t.} \quad (w \cdot (x_i^+ - x_j^-)) = \xi_{ij}, i = 1, \cdots, l^+, j = 1, \cdots, l^-, \tag{6}$$

where $\Theta(x)$ is the step function defined as $\Theta(x) = 1_{x>0}$. It is equivalent to

$$\min_{w,\xi} \quad \tilde{g}(\xi) = \sum_{i=1}^{l^+} \sum_{i=1}^{l^-} \tilde{\Theta}(\xi_{ij}), \tag{7}$$

$$\text{s.t.} \quad (w \cdot (x_i^+ - x_j^-)) = -\xi_{ij}, i = 1, \cdots, l^+, j = 1, \cdots, l^-, \tag{8}$$

where $\tilde{\Theta}(x)$ is defined as $\tilde{\Theta}(x) = 1_{x \geq 0}$. First of all, problem (7)~(8) is ill-posed since solutions of the problem may not be unique, another issue is that the objective function is not differentiable over the range of ξ_{ij}. Thus in order to make it well-posed and tractable, a regularization term can be added to the objective function and $\Theta(\xi_{ij})$ can be relaxed to a linear or convex function. The choice of the regularization term is arbitrary and we can select variant norms of w, such as $l_p(p \geq 0)$-norm

$$||w||_p = (\sum_{i=1}^{n} |w_i|^p)^{\frac{1}{p}}, \tag{9}$$

in the sense

$$||w||_0 = \lim_{p \to 0} ||w||_p^p = \lim_{p \to 0} (\sum_{i=1}^{n} |w_i|^p) = \sharp\{i | w_i \neq 0\}, \tag{10}$$

and

$$\|w\|_1 = \lim_{p\to 1} \|w\|_p^p = \lim_{p\to 1}(\sum_{i=1}^{n} |w_i|^p). \tag{11}$$

In [16] l_2-norm of w was selected thus a convex programming was constructed as

$$\min_{w,\xi,\rho} \quad \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{l^+}\sum_{j=1}^{l^-}\xi_{ij}, \tag{12}$$

$$\text{s.t.} \quad (w\cdot(x_i^+ - x_j^-)) \geq \rho - \xi_{ij}, i = 1,\cdots,l^+, j = 1,\cdots,l^-, \tag{13}$$

$$\xi_{ij} \geq 0, i = 1,\cdots,l^+, j = 1,\cdots,l^-, \tag{14}$$

where $\rho > 0$ is a parameter to be chosen prior. Model (12)~(14) is called the AUC maximizing SVM[12]. Of course, we can apply different l_p-norm of w to derive various models, when $p = 1$ is selected problem (12)~(14) turns to be a linear programming[13], and $p = 0$ provides a very simple and easily grasped notion of sparsity, it is in general NP-hard [26, 27] and not really the right notion for empirical work.

3. AUC maximizing SVM with feature selection

3.1. l_p-norm AUC maximizing SVM

In this section, we consider relaxing l_0-norm of w to $l_p(0 < p < 1)$-norm in order to get more sparse solution than l_1-norm, and at the same time achieving more practical applications than l_0-norm, which results in the following problem

$$\min_{w,\xi} \quad \|w\|_p^p + C\sum_{i=1}^{l^+}\sum_{i=1}^{l^-}\xi_{ij}, \tag{15}$$

$$\text{s.t.} \quad (w\cdot(x_i^+ - x_j^-)) \geq 1 - \xi_{ij}, i = 1,\cdots,l^+, j = 1,\cdots,l^-, \tag{16}$$

$$\xi_{ij} \geq 0, i = 1,\cdots,l^+, j = 1,\cdots,l^-. \tag{17}$$

Figure 1 presents the behavior of the scalar function $|w|_p$ for various values of p, showing that as p goes to zero, this measure becomes a count of the nonzeros in w.

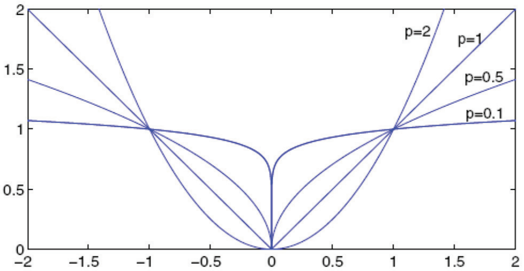

Figure 2: The behavior of $|w|_p$ for various values of p. As p tends to zero, $|w|_p$ approaches the indicator function, which is 0 for $w = 0$ and 1 elsewhere.

3.2. Multi-stage convex relaxation technique

Because of nonconvexity of $\|w\|_p^p$, it is still difficult for problem (15)~(17) to efficiently find the global optimal solution, so we apply multi-stage convex relaxation technique in [28] to solve it. In fact, iteratively reweighted l_1 minimization proposed in [29] is a special case of multi-stage convex relaxation technique. The algorithm is as follows:

Algorithm 1 (Multi-stage convex relaxation procedure for AUC maximizing SVM)

(1) Set the iteration count K to zero, $\alpha_i^{(0)} = 1, i = 1, \cdots, n$;

(2) Solve the weighted AUC Maximizing SVM problem

$$\min_{w,\xi} \quad \sum_{i=1}^{n} \alpha_i^{(K)} |w_i|^2 + C \sum_{j=1}^{l^+} \sum_{k=1}^{l^1} \xi_{ij}, \tag{18}$$

$$\text{s.t.} \quad (w \cdot (x_j^+ - x_k^-)) \geq 1 - \xi_{jk}, j = 1, \cdots, l^+, k = 1, \cdots, l^-, \tag{19}$$

$$\xi_{jk} \geq 0, j = 1, \cdots, l^+, k = 1, \cdots, l^-. \tag{20}$$

and get the solution $(w^{(K)}, \xi^{(K)})$;

(3) Update the weights: for each $i = 1, \cdots, n$,

$$\alpha_i^{(K+1)} = \frac{p}{2} (|w_i^{(K)}| + \varepsilon)^{p-1} |w_i^{(K)}|^{-1} \tag{21}$$

here ε is positive to ensure that the algorithm is well-defined;

(4) Terminate on convergence or where K attains a specified maximum number of iteration K_{max}. Otherwise, increment K and go to step (2).

3.3. Large Scale Problems

Relaxed problems in step (2) of the above algorithm has the serious drawback that the number of constraints is quadratic in the number of training points, so they become very large even for small training set. To cope with this, different strategies are constructed [12, 14, 13].

In this section we will apply a modified technique in [12] because it deal with the reduced problem by one-class svm[30], here we directly solve the relaxed problem in step (2).

Algorithm 2(Approximate relaxed AUC Maximizing SVM)

(1) Given l^+ positive and l^- negative training points, choose appropriate parameter C and integer M;

(2) Generate all $z_{jk} = x_j^+ - x_k^-, j = 1, \cdots, l^+, k = 1, \cdots, l^-$, this results in l^+l^- vectors;

(3) Apply Fast and Exact k-Means (FEKM) these l^+l^- vectors to find M cluster centers;

(4) Solve the problem in step (2) of Algorithm 3 using the M cluster centers.

4. Numerical experiments

In this section, our aim is to provide some empirical analysis of our sparse AUC maximizing Support Vector Machine behavior in two ways: sparse features and higher AUC compared with standard linear C-SVC[1, 2], AUC SVM[16], and l_pSVM[25]. The data sets we used are classical benchmark data sets and a simulation data set.

4.1. Simulation data set

The simulation dataset is generated by the following steps:

(1) Independently generate 100 stochastic vectors $x_i \in R^{20}, i = 1, 2, ..., 100$ as the inputs according to $N(0, 1)$ the normal distribution.

(2) The outputs are determined by the hyperplane $g(x) = [x]_1 + 2[x]_2 + 3[x]3 - 2 = 0$, which means that the output of an input x_i is "+1" if $g(x_i) \geq 0$ and is "-1" if $g(x_i) < 0$.

Note that, in Algorithm 1, the performance depends on the parameters C and p. Therefore, C and p should be adjusted properly. In our experiments, the best value of C and p, q is chosen by 5-fold cross validation. 10 experiments are conducted for this dataset, and the average results are recorded illustrated in Table 1, where the best results are given by the bold form.

Obviously, our sparse AUC maximizing SVM performs well in two ways among four methods. In Table 1, the data in 3th column shows the percentage of the number of the right features over the number of the selected features, which means the bigger the value the better the result. The AUC and Accuracy are computed by averaging the test errors among 10 experiments. From table 1, it is easy to see that sparse AUC maximizing SVM selects more sparse features than C-SVC or AUC-SVM and get higher AUC value than l_p-SVM or not bad than AUC-SVM.

Table 1: Simulation datasets

	$No.\ of\ selected\ features$	$Percentage\ of\ relevent\ features$	AUC	ACC
$C-SVC$	\	\	**0.97**	**0.98**
l_p-SVM	**3.8**	78%	0.91	0.93
$AUC-SVM$	\	\	**0.96**	0.94
$Sparse\ AUC\ SVM$	4.2	77%	**0.96**	0.93

4.2. Real datasets

To test our method on real-world data, three datasets (”heart”, ”Australian”, ”german”) in UCI are used. According to Algorithm 1, the 5-fold cross validations on 3 datasets are conducted to choose the optimal parameters C and p, then the optimal parameters are applied to train the whole training set to select features. Therefore Algorithm 1 and standard linear C-SVC are performed on the new training set constructed by selected features. Compard results are listed in Table 2-Table 4 and Figure 3-Figure 5, where the best results are given by the bold form. We can see that spare AUC SVM is effective in both spares features and higher AUC values than standard linear C-SVC.

Table 2: Compared results on heart

	p	$No.\ of\ selected\ features$	AUC	ACC
$Sparse\ AUC\ SVM$	0.8	4	**0.8437**	**0.8**
$C-SVC$	\	\	0.8266	0.74

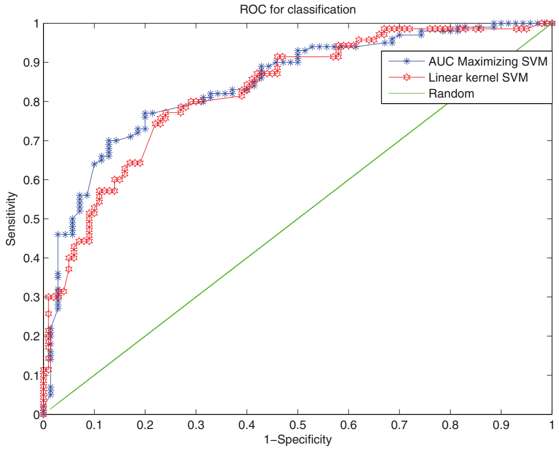

Figure 3: ROC curves

Table 3: Compared results on German

	p	$No.\ of\ selected\ features$	AUC	ACC
$Sparse\ AUC\ SVM$	0.9	24	**0.7546**	**0.73**
$C-SVC$	\	\	0.7358	0.57

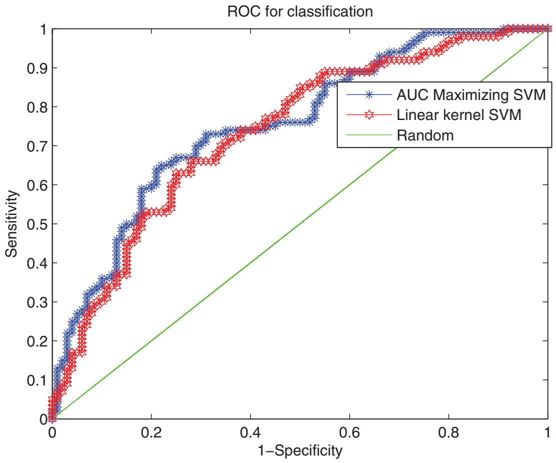

Figure 4: ROC curves

Table 4: Compared results on Australian

	p	$No. of$ $selected$ $features$	AUC	ACC
$Sparse$ AUC SVM	0.9	14	**0.9028**	**0.91**
$C - SVC$	\	\	0.8917	0.88

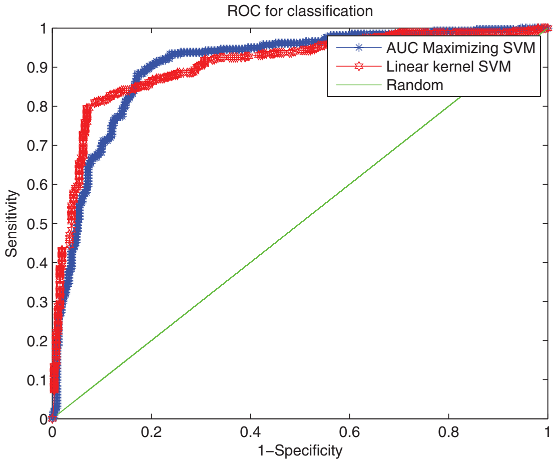

Figure 5: ROC curves

5. Conclusions

In this paper, we proposed a new algorithm – Sparse AUC maximizing SVM which can realize two objectives at the same time: get more sparse features and higher AUC than standard SVM.Some contributions as follows: (1) the algorithm optimizes AUC instead of accuracy; (2) incorporating feature selection into the classification process; (3) conduct experiments to demonstrate the performance.

By changing the 2-norm of w of the separating hyperplane $(w\cdot x)+b = 0$ to p-norm where $0 < p < 1$, This algorithm can delete the less important features corresponding to smaller $|w|$, and by applying the AUC maximizing objective function we can get higher AUC which make the decision function have higher prediction ability. Experiments results proved our algorithm effective and efficient. However, the number of constraints in the programming of this new algorithm is quadratic in the number of training points, so they become very large even for small training set. More efficient methods need to be considered to cope with this problem.

6. Reference

[1] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. Computational Learing Theory, 144-152, 1992.
[2] V. Vapnik. Statistical Learning Theory. New York, NY:Wiley, 1998.
[3] Y. Peng, G. Kou,Y. Shi, Z.X. Chen. A descriptive framework for the field of data mining and knowledge discovery. International Journal of Information Technology and Decision Making (IJITDM), Vol. 7, No. 4, 2008.
[4] Y. shi.Current research trend: Information technology and decision making in 2008.International Journal of Information Technology and Decision Making (IJITDM). Vol. 8, NO. 1, 2009,1-5.
[5] R. Duda, P. Hart, and D. Stork. Pattern classification. Wiley Interscience, 2001.
[6] V. Vapnik and O. Chapelle. Bounds on error expectation for support vector machines. Neural Computation, 12: 2013-2036, 2000.
[7] L. Yan, R. Rodier, M. Mozer, and R. Wolniewicz. Optimizing classifier performance via the wilcoxon-mann-withney statistics. Proceedings of the 20th International Conference on Machine Learning, 2003.
[8] M. Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. ICML Workshop on Learning from Imbalanced Data Sets II, 2003.
[9] J. Huang and C. X. Ling. Using AUC and Accuracy in Evaluating Learning Algorithms. IEEE TRANSACTIONS Transactions on Knowledge and Data Engineering, 17(3): 299-310, 2005.
[10] C. Ferri, P. Flach, and J. Hernandez-Orallo, J. Learning decision trees using the area under the roc curve. Proceedings of the 19th International Conference on Machine Learning. 2002.
[11] T. Fawcett. Using rule sets to maximize ROC performance. Proceedings of the IEEE International Conference on Data Mining, 2001.
[12] U. Brefeld, T. Scheffer. Auc maximizing support vector learning, Proceedings of the 22nd International Conference on Machine LearningłWorkshop on ROC Analysis in Machine Learning, 2005.
[13] D.J.M. Tax, R.P.W. Duin, Y. Arzhaeva. Linear model combining by optimizing the area under the roc curve, Proceedings of the 18th IEEE International Conference on Pattern Recognition, 119-122, 2006.
[14] K. Ataman and W. N. Street. Optimizing Area Under the ROC Curve using Ranking SVMs. In KDD'05, 2005. http://dollar.biz.uiowa.edu/ street/research/kdd05kaan.pdf
[15] A. Rakotomamonjy. Optimizing AUC with support vector machine. In European Conference on Artificial Intelligence Workshop on ROC Curve and AI, 2004.
[16] A. Rakotomamonjy. Support Vector Machines and Area Under ROC curve. 2004.
[17] J. Friedman, T. Hastie, S. Rosset, R. Tibshirani and J. Zhu. Discussion of ”Consistency in boosting” by W. Jiang, G. Lugosi, N. Vayatis and T. Zhang. Ann. Statist. 32, 102-107, 2004.
[18] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm Advances in Neural Information Processing Systems 16, 2004.
[19] I. Guyon, J. Weston, S. Barnhill and V. Vapnik. Gene selection for cancer classification using support vector machines. Machine Learning 46, 389-422, 2002.
[20] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio and V. Vapnik. Feature selection for svms. Advances in Neural Information Processing Systems 13, 2001.
[21] P. Bradley and O. Mangasarian. Feature selection via concave minimization and support vector machines. In International Conference on Machine Learning, Morgan Kaufmann, 1998.
[22] M. Song, C. Breneman, J. Bi, N. Sukumar, K. Bennett, S. Cramer and N. Tugcu. Prediction of protein retention times in anion-exchange chromatography systems using support vector regression. J. Chemical Information and Computer Sciences, 2002.
[23] H. Zou and M. Yuan. The f_∞ norm support vector machine. Statistica Sinica, 18, 379-398, 2008.
[24] W. J. Chen, Y. J. Tian. L_p-norm proximal support vector machine and its applications., International Conference on Computational Science, 2411-2417, 2010.
[25] Y.J. Tian, J. Yu, W.J. Chen. l_p-Norm Support Vector Machine with CCCP. 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery, 4: 1560-1564, 2010.
[26] D. L. Donoho. For most large underdetermined systems of linear equations, the minimal 1-norm solution is also the sparsest solution, Commun. Pure Appl. Math., 59: 797C829, 2006.
[27] B. K. Natarajan. Sparse approximate solutions to linear systems. SIAM J. Comput. 24 (1995), 227-234.
[28] T. Zhang. Multi-stage convex relaxation for learning with sparse regularization. Advances in Neural Information Processing Systems, 22, 2008b.
[29] E. J. Candes, M. B. Wakin, and S. P. Boyd. Enhancing Sparsity by Reweighted l_1 Minimization. Journal of Fourier Analysis and Applications, 14:877-905, 2008.
[30] A. J. Smola, P. J. Bartlett, B. Scholkopf and D. Schuurmans. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA, 2001.