GLSVM: Integrating Structured Feature Selection and Large Margin Classification

Hongliang Fei, Brian Quanz and Jun Huan
EECS Department
University of Kansas
Lawrence, KS, USA
hfei, bquanz, jhuan@ittc.ku.edu

***Abstract*—High dimensional data challenges current feature selection methods. For many real world problems we often have prior knowledge about the relationship of features. For example in microarray data analysis, genes from the same biological pathways are expected to have similar relationship to the outcome that we target to predict. Recent regularization methods on Support Vector Machine (SVM) have achieved great success to perform feature selection and model selection simultaneously for high dimensional data, but neglect such relationship among features. To build interpretable SVM models, the structure information of features should be incorporated. In this paper, we propose an algorithm GLSVM that automatically perform model selection and feature selection in SVMs. To incorporate the prior knowledge of feature relationship, we extend standard 2 norm SVM and use a penalty function that employs a L_2 norm regularization term including the normalized Laplacian of the graph and L_1 penalty. We have demonstrated the effectiveness of our methods and compare them to the state-of-the-art using two real-world benchmarks.**

I. Introduction

Feature selection for high dimensional data has been well investigated in the past. It is well recognized that feature selection is important for improving classification performance since the underlying true model is usually sparse [7], [13], [31]. Furthermore, a parsimonious model is easy to interpret and is preferred in many scientific and industry applications.

Recently high dimensional data with intrinsic feature structure are becoming abundant in many application domains such as bioinformatics, text mining, computer vision, sensor networks and among others. For instance, in microarray classification, genes are features and from databases such as gene ontology, we know that genes are connected together by biological networks [15], [16], [18]. In text mining where key words as features, we have additional information about synonyms or antonyms of the features from databases such as Word Net [6], [20]. In sensor networks, at a given time point regarding the state of the full sensor network, the features are the readings of the sensors, and we usually know the topology or the physical location of the sensors in relation to each other. We call such group of applications where we have prior knowledge about the possible "structure" of features as structured features problem and the new challenge is to incorporate such prior knowledge in the feature selection process to construct better models with improved accuracy, sparseness, and interpretability. Though the structured feature problem is found naturally in supervised and unsupervised learning tasks such as regression, clustering, and classification, in this paper, we focus on classification.

One approach to address the high dimensional structured feature problem is to utilize a separate feature selection method to identify a small subset of features related to labels and use them to build a classification model [4], [5], [12], [27]. In such approach, feature selection and classification are performed separately and hence may not realize the full potential of feature selection. Another approach to address the high dimensional structured feature problem is to devise a regularization learning framework to seek to construct a sparse classification model and hence perform model selection and feature selection simultaneously [2], [22], [28], [31], [26], [30]. For example, Bradley *et al.* [2] first applied L_1 penalty to SVM. Zhu *et al.* [28] proposed an efficient algorithm to compute the entire regularization pathway for the L_1 norm SVM. Wang *et al.* [24] combined L_1 and L_2 penalty together and designed elastic net SVM to select groups of correlated features. Other types of penalties have also been well investigated, such as the F-infinity norm [31], the SCAD penalty [26] and adaptive Lasso penalty [30]. For either approach mentioned above, the model *interpretation* is of high concern. In high dimensional space, we can always find a subset of features highly correlated with outcomes. Regularization does not solve the problem completely and ignoring the structure of features makes model interpretation even harder.

In this paper, we investigate a regularized learning framework for binary SVM that seeks to utilize the structure of the features to guild the feature selection process for constructing better classification models. Our key observation is the availability of the prior structured feature information and our general strategy is to identify sparse models through regularization. Specifically in our method, we formalize the prior structure information as an undirected graph where nodes are features and edges indicate the "closeness" of features. We augment existing SVM learning algorithms with two additional regularization factors: (i) normalized Laplacian of the feature structure graph and (ii)

L_1 penalty. We have designed a practical learning algorithm GLSVM and demonstrated the effectiveness of our method and compared them to the state-of-the-art using two real-world benchmarks.

The rest of the article is organized as follows. Section 2 discusses related work. Section 3 presents background information and detailed discussion of our algorithms. Section 4 presents the experimental study of our algorithms as compared to competing methods. Finally we give a short conclusion and a discussion of the future work.

II. Related Work

Incorporating prior knowledge of features into large margin classifier such as SVM has recently attracted research interest in the machine learning and data mining communities. For instance, Wu et al. [25] enforce both sparsity and heredity principle for the data in which features have a natural hierarchical structure relationship. Gômez-Chova et al. [9] extended SVM with un-normalized graph Laplacian for image classification. Zou & Lin replaced the L_2 norm with the F-infinity norm [31] in SVM to obtain sparsity. This method is recently applied to Microarray classification [29]. A more general framework was investigated in [1].

Our work is different from existing work in that we use a general graph to capture relationship between features for binary SVM and feature selection. In our method we consider a graph as a manifold and we factor in the graph topology using graph Laplacian as a regularization factor. By incorporating both an L_1 penalty and a normalized Laplacian penalty, we enforce model sparsity and smooth variation over the known graph, effectively selecting features that are grouped according to the known graph structure.

III. Methodology

A. Problem Statement

Given a set of n training samples $\mathcal{T} = \{\{\vec{x}_1, y_1\}, \ldots, \{\vec{x}_n, y_n\}\}$ sampled from $\mathcal{X} \times \mathcal{Y}$ our goal is to derive (learn) a mapping for a sample $\vec{x} \in \mathcal{X} \subset \mathbb{R}^p$ to an output $y \in \mathcal{Y}$, called a classification function or classifier. Since we are dealing with high dimensional, low sample size data, $p \gg n$, we focus here on learning a linear classifier. For such data a linear classifier is already complex enough, and multiple such classifiers may perfectly fit the training data. Instead the concern is on generalization performance, asserting bias to help select the correct model, which we aim to achieve using existing prior knowledge as a guide, incorporated through regularization.

B. Large Margin Classification

Our general approach for learning a suitable classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$ is to solve a convex optimization problem for a particular loss function combined with regularization terms to guide the model learning. The general form is:

$$f = \arg\min_{f \in \mathcal{H}_K} \quad C \sum_{i=1}^{n} V(\vec{x}_i, y_i, f) + R(f) \qquad (1)$$

where $K(\vec{x}, \vec{x}') : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a kernel function which defines an inner product (dot product) between samples in $\mathcal{X}$, $\mathcal{H}_K$ is the set of functions in the kernel space, $R(f)$ is a regularization term that is a function of f, and C is a regularization coefficient. V measures the fitness of the function in terms of predicting the class labels for training samples and is called a risk function. The hinge loss function is a commonly used risk function in the form of $V = (1 - y_i f(\vec{x}_i))_+$ and $x_+ = x$ if $x \geq 0$ and zero otherwise, and is the loss function we use. Unlike many other common loss functions, like the binomial loss of logistic regression, the hinge-loss function is purely discriminative in the sense that it is only non-zero for instances that are not well-classified, so that well-classified instances do not affect the loss at all. Here we focus on binary classification, restricting $\mathcal{Y} = \{-1, 1\}$. Furthermore, as mentioned we restrict our focus to linear classifiers, with $K(\vec{x}, \vec{x}')$ simply equal to the inner product of $\vec{x}$ and $\vec{x}'$. A linear classifier takes the form, $f(\vec{x}) = \text{sign}(\vec{w}^T \vec{x} + b)$; $\vec{w}$ controls the orientation of the hyperplane, and b its offset. The large margin approach to selecting a hyperplane is to select the one which provides the largest margin, the distance to the nearest well-classified training point, proportional to $1/||\vec{w}||_2^2$. Thus the basic regularization term is $R(f) = .5\vec{w}^T\vec{w}$. Following the Support Vector Machine (SVM) convention, through introducing slack variables $\epsilon_i, i = 1 \ldots n$ we represent the hinge loss as a 1-norm penalty on $\vec{\epsilon}$, $\sum_{i=1}^{n} \epsilon_i$, with the additional constraints given below. Thus the standard large margin classifier, or SVM with a 1-norm soft margin classifier (adopting common terminology, e.g. [21]), is given by solving the following optimization problem, equation 2:

$$\begin{array}{ll} \min. & \frac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{n} \epsilon_i \\ \text{s.t.} & y_i(\vec{w}^T\vec{x}_i + b) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0, \quad \forall i = 1, ..., n. \end{array} \qquad (2)$$

C. Regularization and Laplacian SVM

We now consider additional regularization $R(f)$ to help bias our model learning when $p \gg n$. In many high dimensional low sample size problems, we have access to *a priori* domain knowledge about the relationship of the features. In our model, we capture such domain knowledge as an undirected graph G, whose nodes correspond to the p features. Edges in the graph G are weighted, where the weight $W_{i,j} > 0$ indicates the "similarity" between the two features. Weight 0 indicates that the two features are not expected to be similar. We incorporate the a priori domain knowledge by adding a Tikhonov regularization factor $\frac{1}{2}\sum_{i,j} W_{i,j}(\beta_i - \beta_j)^2$ to a convex fitness function $\ell(X, \vec{y}; \vec{\beta})$ to enforce that the feature coefficients vary smoothly for neighboring features. Assuming a symmetric weight matrix, the Tikhonov regularization factor could be conveniently written in the matrix format $\vec{\beta}^T L \vec{\beta}$ where L is the *Laplacian* of G given by: $L = D - W$. W is the edge weight matrix, and D is the density matrix of W, defined as $D =$

$(W_{i,j})_{i,j=1}^n$ where $d_{i,j} = \begin{cases} \sum_{k=1}^n W_{i,k} & \text{if } i=j \\ 0 & \text{otherwise} \end{cases}$ To avoid having any feature "dominate" the penalization function, we use the *normalized Laplacian* $\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ to normalize the weight of each feature.

Specifically, we seek to identify a linear classifier, parameterized by a vector $\vec{w}$ to solve the following minimization problem in Equation 3.

$$\begin{array}{ll} \min. & \vec{w}^T(\lambda_1 \mathcal{L} + \frac{1}{2} I)\vec{w} + C\sum_{i=1}^n \epsilon_i \\ \text{s.t.} & y_i(\vec{w}^T \vec{x}_i + b) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0, \quad \forall i = 1, ..., n, \end{array} \tag{3}$$

where I is the $p \times p$ identity matrix, to incorporate the L_2 norm, $||w||_2^2$, margin term. We call such a method LapSVM, standing for Laplacian regularized Support Vector Machine.

D. Sparse Model Construction

The Tikhonov regularization typically generates "smooth" models where model parameters vary smoothly according to the reference graph structure. As pointed out in [19] using graph spectrum analysis, the connection of the Laplacian based regularization and the normal L_2 based regularization used in SVM is that graph Laplacian regularization penalize high frequency (variance) components in the eigenspace and hence the trained model is "smooth", as evaluated on the reference graph structure. The disadvantage of the L_2 norm based methods is that the obtained model is usually not "sparse" in the sense that there is not automatic feature selection. Model sparsity is often desirable for high-dimensional problems for both improved generalization, interpretability, and feature selection. To encourage sparsity and in effect also perform feature selection, below we propose two methods, both incorporate lasso regularization [23], an L_1 norm penalty on model coefficients $\vec{w}$.

GLSVM. The first method that we propose is a direct incorporation of the L_1 penalty in the Laplacian regularized Support Vector Machine. Specifically, we seek a linear classifier, parameterized by a vector $\vec{w}$ to minimize the following equation.

$$\begin{array}{ll} \min. & \vec{w}^T(\lambda_1 \mathcal{L} + \frac{1}{2} I)\vec{w} + C\sum_{i=1}^n \epsilon_i + \lambda_2 ||\vec{w}||_1 \\ \text{s.t.} & y_i(\vec{w}^T \vec{x}_i + b) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0, \quad \forall i = 1, ..., n, \end{array} \tag{4}$$

where $||\vec{w}||_1 = \sum i = 1p|w_i|$ is the L_1 norm of $\vec{w}$. We call this method GLSVM, standing for **G**raph Laplacian and $\mathbf{L}_1$ regularized SVM algorithm. The objective function is a combination of a convex quadratic term and an L_1 norm term, both of which are convex, and the constraints are linear inequalities, thus the problem is convex. Since this problem is convex, general convex solvers can be used to derive the solution efficiently. For instance, for our experiments we applied the convex solver CVX [10], [11], which uses interior point methods. The GLSVM method is clearly related to a group of SVM methods. For example, first, setting $\lambda_1, \lambda_2 = 0$ we arrive at the standard SVM formulation. Next, setting $\lambda_1 = 0$, removing the L_2 norm $||w||_2^2$, and fixing $C = 1$ we arrive at the L_1 SVM [28], the lasso type SVM. In addition, setting $\lambda_2 = 0$, we have LapSVM. Though appealing, the disadvantage of the GLSVM method is that optimization is slightly difficult since we do not have the nice dual quadratic programming solution any more, as commonly used in SVM.

IV. Experimental Study

We have implemented our algorithm of SVM with graph Laplacian based Lasso penalty (GLSVM). We have performed a rigorous evaluation of our learning algorithms in terms of modeling accuracy and feature selection performance using two real-world data sets from different application domains. For comparison, we compared our methods with two state-of-the-art SVM based algorithms with the capability of embedded feature selection. These two algorithms are SVM with Lasso (L_1) penalty [28] (LassoSVM) and SVM recursive feature elimination (SVM RFE) [12]. To demonstrate the utility of embedded feature selection strategy, we also compare our methods with baseline SVM with linear kernel (Linear SVM) and graph Laplacian regularized SVM (without feature selection).

We implemented GLSVM, LassoSVM, and LapSVM in Matlab. In our implementation we use the optimization toolbox provided by Matlab to solve the related quadratic and linear programming problems and use the convex optimization solver CVX [10], [11] to solve convex optimization problems. For base-linear linear SVM we use the LibSVM package [3] and for SVM RFE, we use the Spider toolbox available at http://www.kyb.tuebingen.mpg.de/bs/people/spider/main.html.

A. Data sets

We used the following two data sets for our real-world data study:

Diabetes Data: The data set is obtained from [17]. As in [16], we use only the 34 samples of subjects, including 17 samples with type 2 diabetes and 17 samples with normal glucose tolerance. For prior information of features, we collected all pathway information from the Kyoto Encyclopedia of Genes and Genomes (KEGG) database [14] and use the global test [8] method to obtain related pathways; we select all pathways found to be related to the diabetes outcome by the global test method with a significance P-value of less than 0.1 and keep those 13 non-overlapping pathways that have an associated graph structure. We only focus on these 701 genes.

20 NewsGroup Data: This data set is a collection of approximately 20,000 newsgroup documents, evenly distributed in 20 different newsgroups and features are about 60,000 keywords. The data is available at http://people.csail.mit.edu/jrennie/20Newsgroups/ and we use the second one (by date). We merge the original training (60%) and test

(40%) sets to form a whole data set. To perform classification, we single out two classes with 1942 documents that are very correlated to each other (ms-windows.misc and ibm.pc.hardware). We collect the feature set as a set of 610 key words which occur at least 25 times in the 1942 documents excluding stop words. To build feature graph, we follow the same procedure in [20]. Refer to [20] for more information. We randomly sampled 100 documents from each of the two classes for binary classification.

B. Model Evaluation

Model Construction. For both real-world data sets, we partition the data set into 10-folds to perform 10-fold cross-validation (CV) with 9 folds used for training and 1 fold for testing. We use another 10-fold CV on the training data set to select the regularization parameters for each method with grid search. We then generate a single model from the entire training set with the selected parameters and apply the model to the testing data set for prediction.

Model Comparison. For model comparison, we collect the sensitivity (TP/(TP+FN)), specificity (TN/(TP+FP)) and accuracy ((TP+TN)/S) of the trained model, where TP stands for true positive, FP stands for false positive, TN stands for true negative, FN stands for false negative, and S stands for the total number of samples. All the values (specificity, sensitivity, accuracy) reported are collected from the testing data set only and are averaged across 10-fold CV with 5 replicates in a total of 50 experiments.

It is difficult to compare the selected features directly other than evaluating their classification performance. Toward that end of comparing feature selection performance of different feature selection methods, we record the number of selected features in each cross validation and report the average number of selection frequency for each feature in the experiments. To demonstrate the group feature selection effect, for Microarray data where the feature graph is sparse, we simply collect the number of selected feature clusters (or pathways for Microarray data). For News group data, the feature graph is dense and there is no natural way to partition the graph into "components". We define the *average feature separation* $\bar{d}$ as the average shortest path length of pairs of selected features. That is $\bar{d} = \sum_{i\ j} d(i,j)$ where $i, j \in F$, F is the selected features, $d(i,j)$ is the shortest path length between feature i and j in the original feature graph.

C. Performance

In this subsection, we evaluate the performance of the proposed method compared with LassoSVM, LapSVM, LinearSVM and SVMRFE by applying the algorithms to two real world data sets.

In Table I, we report the average values of test sensitivity, specificity, accuracy, number of selected features, number of selected pathways for the Microarray data, and the average feature separation for the NewsGroup data in the 5 replicates

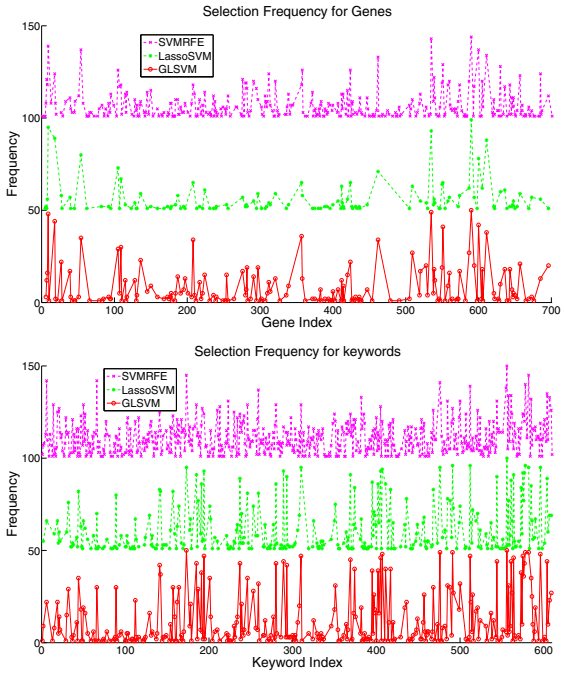

Figure 1. Upper: Selection frequency of each feature for Diabetes. Lower: Selection frequency of each feature for News data.

of 10-fold cross validation (a total of 50 experiments). The standard deviation is between 7% and 15% and we do not list them for simplicity.

As shown in Table I, there is a clear trend that the classification performance with feature selection is better than using all the features, which reflects the importance of feature selection. SVM RFE generally achieves a comparable performance compared with LassoSVM but selects much more features and feature groups. Lasso SVM always selects the smallest though not necessary the optimal feature set (in terms of classification accuracy). The GLSVM builds the best model with highest classification accuracy and the sparest model in terms of selecting the least number of feature groups.

The stability of feature selection is an important metric to evaluate different feature selection methods. We measure the stability by measuring the consistency of feature selection across cross validations. In Figure 1, we compute and plot frequencies of selected features in the 50 experiments (5 replicates of 10-fold cross validation) for the real-world data sets. From the figure, we have the following observations: (i) all four feature selection methods select a small number of features, (ii) such selections are generally stable across different cross validation iterations, and (iii) there is a high level of agreement among the three methods. For example, we observe that there are about 326 features for Diabetes and 179 features for news data that are never selected by any of the feature selection methods.

Table I
AVERAGE SENSITIVITY (SEN), SPECIFICITY (SPE), ACCURACY (ACC), NUMBER OF SELECTED FEATURES (F), THE NUMBER OF SELECTED PATHWAYS FOR THE MICROARRAY DATA ($\mathcal{P}$), OR THE AVERAGE FEATURE SEPARATION FOR NEWSGROUP DATA ($\overline{\mathcal{D}}$). STARS (*) DENOTE THE HIGHEST VALUE OR LOWEST AMONG ALL COMPETING METHODS FOR A DATA SET.

	Data Set:Diabetes					Data Set:News				
Methods	F	$\mathcal{P}$	SEN	SPE	ACC	F	$\overline{\mathcal{D}}$	SEN	SPE	ACC
SVMRFE	43	10	0.613	0.740	0.7	111	1257	0.683	0.797	0.778
LassoSVM	16*	8	0.633	0.729	0.705	66 *	742	0.721	0.846	0.788
GLSVM	25	7*	0.658	0.779*	0.734*	73	659*	0.727	0.866*	0.802*
LapSVM	701	13	0.787*	0.532	0.690	610	8949	0.811*	0.708	0.760
LinearSVM	701	13	0.707	0.667	0.694	610	8949	0.772	0.744	0.758

Though GLSVM may not select the sparsest model comparing to LassoSVM, features selected by GLSVM tend to remain the same across different cross validations. For example, we count the number of same features selected during each cross validation. We observe that GLSVM consistently select 10 genes, 21 keywords respectively. These facts suggest that there is a higher-level of agreement between different cross validation iterations in terms of feature selection. In other words, the feature selection process is much more stable.

We also observe the feature selected by the GLSVM tend to be "clustered" together. To further study the group feature selection effect of our method for the Microarray data, we have singled out all genes that are selected in all 50 experiments and investigated the pathways that these selected genes belong to. We observe that genes selected by GLSVM belongs to 10 pathways, while those selected by LassoSVM belong to 12 pathways and 13 pathways for SVMRFE. Below we list the biological role of six pathways from which all the three feature selection methods frequently select genes: pathway 1 (Gluconeogenesis), pathway 2 (Oxidative phosphorylation), 3 (Alanine and aspartate metabolism), 11 (PPAR signaling pathway), pathway 12 (SNARE interactions in vesicular transport) and 13 (Insulin signaling pathway). Clearly pathway 1, 3, 13 are related to the diabetes diseases.

V. CONCLUSIONS AND FUTURE WORK

In many real-world applications we often have access to prior knowledge of the structure of features in data sets. Here we present a learning framework of integrating model selection and feature selection on networked features to incorporate prior knowledge of structured features in Support Vector Machines. By introducing normalized graph Laplacian as a regularization term, we have designed the GLSVM, combining L_1 and L_2 penalty together to achieve both sparsity and smoothness with respect to the reference feature network. As evaluated on two real-world data sets, our method usually performs a stable feature selection and enjoy better classification accuracy comparing to competing methods. In the future, we consider to incorporate networked features where the edge weight represents the uncertainty to handel uncertainties in the prior structure knowledge.

Acknowledgments

The work is partially supported by the NSF award IIS 0845951 and the Office of Naval Research N00014-07-1-1042.

REFERENCES

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7, 2006.

[2] P. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In *Proceedings of the Fifteenth International Conference of Machine Learning(ICML98*, pages 82–90. Morgan Kaufmann, 1998.

[3] C. Chang and C. Lin. Libsvm: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[4] H. Cheng, X. Yan, J. Han, and C.-W. Hsu. Discriminative frequent pattern analysis for effective classification. In *Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE)*, 2007.

[5] H. Fei and J. Huan. Structure feature selection for graph classification. In *Proc. ACM 17th Conference on Information and Knowledge Management*, 2008.

[6] C. Fellbaum. *WordNet: an electronic lexical database*. the MIT Press, 1998.

[7] J. Friedman, T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. Discussion of boosting papers. *The Annals of Statistics*, pages 102–107, 2004.

[8] J. Goeman, S. A. van de Geer, F. de Kort, and H. C. van Houwelingen. A global test for groups of genes: testing association with a clinical outcome. *Bioinformatics*, 20(1):93–99, 2004.

[9] L. Gomez-Chova, G. Camps-Valls, J. Munoz-Mari, and J. Calpe. Semisupervised image classification with laplacian support vector machines. *Geoscience and Remote Sensing Letters, IEEE*, 5(3):336–340, 2008.

[10] M. Grant and S. Boyd. *CVX: Matlab software for disciplined convex programming*, December 2008. Web page and software available at http://stanford.edu/~boyd/cvx.

[11] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. *Lecture Notes in Control and Information Sciences*, 371:95–110, 2008.

[12] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002 January.

[13] T. Hiastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning;Data Mining, Inference and Prediction*. Springer Verlag, New York, first edition, 2001.

[14] M. Kanehisa, S. Goto, M. Hattori, K. F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, , and M. Hirakawa. From genomics to chemical genomics: new developments in kegg. *Nucleic Acids Research*, 34:D354–357, 2006.

[15] C. Li and H. Li. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24(9):1175–1182, 2008.

[16] L. Liang, V. Mandal, Y. Lu, and D. Kumar. Mcm-test: a fuzzy-set-theory-based approach to differential analysis of gene pathways. *BMC Bioinformatics*, 9(Suppl 6):S16, 2008.

[17] V. Mootha, C. Lindgren, K. Eriksson, A. Subramanian, S. Sihag, J. Lehar, P. Puigserver, E. Carlsson, M. Ridderstraale, E. Laurila, and et al. PGC-1 α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics*, 34(3):267–273, 2003.

[18] M. Y. Park and T. Hastie. Penalized logistic regression for detecting gene interactions. *Biostatistics*, 9(1):30C50, 2008.

[19] F. Rapaport, A. Zinovyev, M. Dutreix, E. Barillot, and J.-P. Vert. Classification of microarray data using gene networks. *BMC Bioinformatics*, 8:35+, 2007.

[20] T. Sandler, P. P. Talukdar, and L. H. Ungar. Regularized learning with networks of features. In *NIPS08*, 2008.

[21] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

[22] M. Song, C. Breneman, J. Bi, N. Sukumar, K. Bennett, S. Cramer, and et al. Prediction of protein retention times in anion-exchange chromatography systems using support vector regression. *Journal of chemical information and computer sciences*, 42(6):1347–1357, 2002.

[23] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58:267–288.

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

[25] S. Wu, H. Zou, , and M. Yuan. Structured variable selection in support vector machines. *Electronic Journal of Statistics*, 2:103–117, 2008.

[26] S. Zhang, A. Golbraikh, S. Oloff, H. Kohn, and A. Tropsha. A novel automated lazy learning QSAR (all-QSAR) approach: Method development, applications, and virtual screening of chemical databases using validated all-qsar models. *J. Chem. Inf. Model.*, 46:1984–1995, 2006.

[27] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007.

[28] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm support vector machines. In *The Annual Conference on Neural Information Processing Systems 16*, 2004.

[29] Y. Zhu, X. Shen, and W. Pan. Network-based support vector machine for classification of microarray samples. *BMC Bioinformatics*, 10, 2009.

[30] H. Zou. An improved 1-norm support vector machine for simultaneous classification and variable selection. In *Proceedings of Eleventh International Conference on Artificial Intelligence and Statistics*, 2007.

[31] H. Zou and M. Yuan. F∞ norm support vector machine. *Statistica Sinica*, 18:379–398, 2008.