Feature Import Vector Machine: A General Classifier with Flexible Feature Selection

Samiran Ghosh[1,2*] **and Yazhen Wang**[3]

[1]*Department of Family Medicine & Public Health Sciences, Wayne State University, Detroit, MI 48201, USA*

[2]*Center of Molecular Medicine and Genetics, Wayne State University, Detroit, MI 48201, USA*

[3]*Department of Statistics, University of Wisconsin, Madison, WI, USA*

Received 21 November 2013; revised 22 September 2014; accepted 12 November 2014
DOI:10.1002/sam.11259
Published online in Wiley Online Library (wileyonlinelibrary.com).

Abstract: The support vector machine (SVM) and other reproducing kernel Hilbert space (RKHS) based classifier systems are drawing much attention recently owing to its robustness and generalization capability. General theme here is to construct classifiers based on the training data in a high dimensional space by using all available dimensions. The SVM achieves huge data compression by selecting only few observations that lie close to the boundary of the classifier function. However when the number of observations is not very large (small n) but the number of dimensions/features is large (large p), then it is not necessary that all available features are of equal importance in the classification context. Possible selection of a useful fraction of the available features may result in huge data compression. In this paper, we propose an algorithmic approach by means of which such an *optimal* set of features could be selected. In short, we reverse the traditional sequential observation selection strategy of SVM to that of sequential feature selection. To achieve this we have modified the solution proposed by Zhu and Hastie in the context of import vector machine (IVM), to select an *optimal* sub-dimensional model to build the final classifier with sufficient accuracy. © 2015 Wiley Periodicals, Inc. Statistical Analysis and Data Mining 8: 49–63, 2015

Keywords: classification; import vector machine; radial basis function; regularization; reproducing kernel Hilbert space; support vector machine

1. INTRODUCTION

Many machine learning as well as nonparametric function estimation methods can be recast into a unified regularization-based modeling framework. Almost all the present kernel machine-based approaches try to do function estimation by representing the original function as a linear combination of the basis functions in the higher dimensional space. The popular support vector machine (SVM) is a member of this framework. SVM achieves huge data compression by selecting only few support points (i.e., observations) that lies near the boundary of the classifier function. However domains such as bioinformatics which generally produce thousands of dimensions/features/covaraites (p) and only few observations (n) are well suited for kernel machine-based function estimation if compression can be achieved in feature space (i.e., in p) rather than in terms of observations (i.e., in n). Another advantage of such feature selection is biomarker discovery, which generally corresponds to only few selected features. In this paper we propose a new kernel machine based on regularization principle which achieves function estimation with sufficient accuracy by selecting only few features in the original input space and thus named feature import vector machine (FIVM). The basic idea ([1,2]) of selecting a sub-model in place of a complicated full model is not new. Recently Zhu and Hastie [3] proposed import vector machine (IVM) as an alternative to the popular SVM, which is built on kernel logistic regression (KLR). The IVM has the following advantages over standard SVM:

- Probabilistic interpretation of the classification result.

[*] *Correspondence to:* Samiran Ghosh (sghos@med.wayne.edu)

- Selection of few observations, hence SVM-like data compression.
- Straightforward generalization of IVM for multiclass classification.

The most innovative feature of IVM is replacement of the SVM loss by the negative log-likelihood (NLL) of the binomial distribution. However in doing so it destroys the sparse observation selection property of SVM. To solve this problem Zhu and Hastie [3] proposed a sequential selection (a variant of greedy search) strategy to select only few *important* observations named as import vectors. We pose the same problem if we exchange the role played by n and p in IVM. Hence the idea is to create a dimension/feature screening/selection methodology via a sequential search strategy over the original feature space. Our proposal has the following features:

- It uses kernel machine for classifier construction.
- It produces a nonlinear classification boundary in the original input space.
- The feature selection is done in the original input space, not in the kernel transformed feature space.
- Unlike SVM (both L_1 and L_2), the result has probabilistic interpretation.

Though the feature selection via SVM in the transformed kernel space or only via a linear kernel (see ref. [4] and references therein) is a well-studied problem, to the best of our knowledge except for the only one article [5], there exist very limited attempt where feature selection is done in SVM in the original space, when a nonlinear kernel function is in use. However as reported in ref. [4] optimization problem presented in ref. [5] is potentially non-convex due to the placement of feature scaling factor inside the kernel (see also ref. [6]), as a result computational and resulting convergence is questionable. Moreover except for empirical observations no theoretical justification is provided in ref. [5], neither about the optimality of the selected features nor about the classification accuracy under any setup. If classification can be done using nonlinear kernel but feature selection in the original space, this leads to tremendous advantage as in one hand we are able to combine the power of nonlinear classification via kernel machine, on the other hand important features are selected in the original input space, thus making their physical interpretation straight forward.

The rest of the paper is organized as follows. A brief introduction of the RKHS is provided in Section 2. We also briefly describe some key properties of SVM and KLR in Section 3. Dimension screening and some key optimality properties are described in Section 4. Here we also formally propose FIVM algorithm. Section 5 illustrates our proposed methodology on some synesthetic as well as real-life classification examples from bioinformatics domain. Multiclass extension of the proposed methodology is briefly discussed in Section 6. We conclude the article with a discussion in Section 7. For brevity all the proofs are provided in the Appendix.

2. REPRODUCING KERNEL HILBERT SPACE AND FUNCTION REPRESENTATION

Formally speaking, let $T = (\mathbf{x}_i, y_i)$, $1 \le i \le n$, are i.i.d. pairs distributed as $(\mathbf{x}, y)$ with probability measure $\mathbb{P}$, where $y_i \in C = \{-1, 1\}$ or $y_i \in I\!R$ and $\mathbf{x}_i = (x_1, \cdots, x_p) \in I\!R^p$. A general class of regularization problems has the form

$$\underset{f \in \mathcal{H}}{\text{argmin}} \left[\frac{1}{n} \sum_{i=1}^{n} L\,(y_i, f(\mathbf{x}_i)) + \lambda J(f) \right], \tag{1}$$

where $L\,(y_i, f(\mathbf{x}_i))$ is a convex loss function, $J(f)$ is a penalty functional, $\lambda > 0$ is the smoothing or regularization parameter, and $\mathcal{H}$ is a space of functions on which $J(f)$ is defined. An important subclass of the form (1) is generated by a positive definite kernel $K(\mathbf{x}, \mathbf{x}')$ and the corresponding space of functions $\mathcal{H}_k$ is called reproducing kernel Hilbert space (RKHS). In this article, we will employ radial basis function (RBF) as kernel which is given by, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\theta^2}\right\}$. For special case of squared norm regularizer, by the important 'Representer Theorem' of Kimeldorf and Wahba [7,8], any solution to the above problem is finite-dimensional, and has the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i), \tag{2}$$

where $J(f) = ||f||^2_{\mathcal{H}_k}$. In the light of Eq. (2), Eq. (1) can be rewritten as

$$\underset{\boldsymbol{\alpha} \in I\!R^p}{\mathit{argmin}} \left[\frac{1}{n} \sum_{i=1}^{n} L\,(y_i, f(\mathbf{x}_i)) + \lambda ||f||^2_{\mathcal{H}_k} \right]. \tag{3}$$

Smoothing spline analysis of variance (ANOVA), penalized polynomial regression, SVM, IVM, KLASSO (kernelized version of LASSO proposed by Roth [9]) are all examples of RKHS-based methodology. For a more rigorous treatment in this matter please refer to refs [8,10,11].

3. SVM, KERNEL LOGISTIC REGRESSION AND LASSO

The standard SVM [12] belongs to a general class of regularization problems described earlier in Eq. (3), where $L(y_i, f(\mathbf{x}_i))$ is chosen as the hinge loss function. The SVM could be cast in the form of minimizing a loss function as a function of the margins $yf(\mathbf{x})$ ([13,14]). Fitting SVM is equivalent to finding

$$\underset{f\in\mathcal{H}_k}{argmin}\ \frac{1}{n}\sum_{i=1}^{n}[1-y_i f(\mathbf{x}_i)]_+ + \lambda||f||^2_{\mathcal{H}_k}. \tag{4}$$

The standard SVM produces a nonlinear classification boundary in the original input space via kernel transformation on the features. The dimension of the transformed feature space is often very high and may be infinite for some cases. However it employs *kernel trick* on a positive definite reproducing kernel to achieve this seemingly impossible computation. The optimal solution of the Eq. (4) is given by Eq. (2). It turns out that for most cases a sizeable number of α_i's are zero, which seems an attractive property for data compression. Nonzero $\mathbf{x}_i$'s are known as support points and Eq. (2) is known as sparse representation of $f(\mathbf{x})$. Recently Lin [15] has shown that the SVM implements Bayes' rule asymptotically.

However SVM has two major drawbacks. First, it works well for two-class problems. Recently different researchers ([16–18]) have proposed methodologies for its possible multiclass extension. However the success of the proposed methodologies is questionable. Second, SVM can only estimate sign$[p(\mathbf{x})-\frac{1}{2}]$, while the quantity $p(\mathbf{x}) = P(Y=1|X=\mathbf{x})$ giving classification probability is often of interest by itself. Noting the similarity of the hinge loss of SVM and that of the NLL of the binomial distribution (plotted in Fig. 1), Zhu and Hastie [3] proposed to replace the hinge loss in Eq. (4). This essentially produces kernel logistic regression (KLR) given by:

$$\underset{f\in\mathcal{H}_k}{\text{argmin}}\ \frac{1}{n}\sum_{i=1}^{n}\ln[1+e^{-y_i f(\mathbf{x}_i)}] + \lambda||f||^2_{\mathcal{H}_k}. \tag{5}$$

The KLR eliminates two drawbacks of SVM stated above. However it no longer enjoys support point property of the SVM, thus making all of the α_i's nonzero in Eq. (2). Zhu and Hastie [3] suggested an idea based on selecting an approximate submodel to overcome this difficulty. Their algorithm termed as IVM uses both y_i and input $\mathbf{x}_i$ to select a subset of T to approximate the full model. However for feature selection we face a problem of different kind, where selection of dimension (p) is of prime interest. For example in large p small n domain, reduction of dimension (p) is more important than reduction of the number of observations (n). Simultaneous screening of p and n is another possibility in the recently popular *Big data* context. However in this article our focus is exclusively on dimension/feature selection. In the variable selection context LASSO proposed by Tibshirani [19] is a very successful method for automatic feature selection. In penalized regression context, L_1 penalized methods offer automatic dimension selection owing to the nature of the penalty function. However though LASSO is successful in many situations, it has limitations. For example in the context of large p small n domain ($p \gg n$), LASSO can select at most n dimensions only. Also in situations where two (or more) dimensions have high correlation, LASSO tends to select only one dimension from the group. Park and Hastie [20] considered NLL of the binomial distribution with L_1-penalty for dimension selection in the classification framework; however, they do not employ any kernel transformation. Feature selection in the transformed feature space is another Possibility; however, owing to the nonlinearity of the kernel it is often impossible to map the direct relationship between transformed and original features. Moreover selection via L_1-penalty will inherit all merits as well as drawbacks of LASSO discussed so far. The FIVM algorithm that we are proposing here does not have these drawbacks as it does not rely on a specific penalty function for feature selection. Depending upon the choice of the convergence parameter (ϵ introduced latter) as many/few dimensions can be selected, as desired.

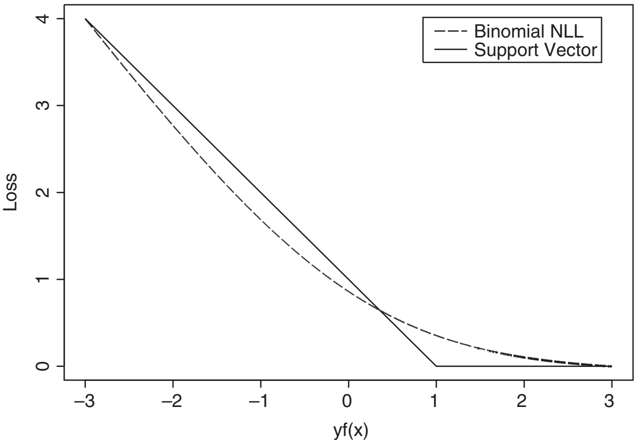

Fig. 1 Hinge loss of SVM and NLL of binomial distribution for two-class classification, $y \in \{-1, 1\}$.

4. FEATURE SELECTION IN KLR FRAMEWORK

Let us denote the index set $\mathcal{L} = \{1, 2, \ldots, p\}$ as the collection of all available features/dimensions in the original input space and $\mathcal{S} \subseteq \mathcal{L}$ is an arbitrary subset of these available features such that $q = card(\mathcal{S}) \leq p$. Given an index set $\mathcal{S}$, it will also induce a corresponding feature space which we denote by $\mathcal{F}_{\mathcal{S}}$. For the RBF, a kernel on any arbitrary collection of dimensions such as $\mathcal{S}$ can be defined as $K_{\mathcal{S}}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{\sum_{t\in\mathcal{S}}(x_{it}-x_{jt})^2}{2\theta^2}\right\}$. We can rephrase the original regularization problem on this arbitrary set $\mathcal{S}$ as to obtain

$$\underset{\boldsymbol{\alpha} \in I\!R^q}{argmin} \left[\frac{1}{n}\sum_{i=1}^{n} L_{\mathcal{S}}\left(y_i, \alpha_i, \mathbf{x}_i\right) + \lambda \sum_{i,j=1}^{n} \alpha_i \alpha_j K_{\mathcal{S}}(\mathbf{x}_i, \mathbf{x}_j)\right]. \tag{6}$$

It is easy to note that when $\mathcal{S} = \mathcal{L}$, problem (6) reduces to (3). Now using Wahba's representer theorem, solution for (6) is given by

$$f_{\mathcal{S}}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K_{\mathcal{S}}(\mathbf{x}, \mathbf{x}_i). \tag{7}$$

We want to study the behavior of the function $f_{\mathcal{S}}(\mathbf{x})$ as $\mathcal{S} \to \mathcal{L}$. More specifically speaking, whether the addition of more dimensions in $\mathcal{S}$ will increase the classifiers accuracy. Next we propose a theorem under the assumption that the transformed feature space is so rich that the training data are completely separable.

THEOREM 1: If training data is separable in $\mathcal{S}$ then it will be separable in any higher dimensional space provided it is a superset of $\mathcal{S}$.

The advantage of selecting a submodel based on few features is multifold. Computational burden is not only greatly reduced but it also obeys the principle of parsimony without jeopardizing classification accuracy. Theorem 1 shows that separating hyperplane in $\mathcal{S}$ is also a separating hyperplane in $\mathcal{L}$. Though it does not guarantee that it is also a margin maximizing hyperplane in $\mathcal{L}$. Under complete separability of the training data set in $\mathcal{L}$, theoretical justification of using a submodel in $\mathcal{S}(\subseteq \mathcal{L})$ is provided next. Suppose the dictionary of the basis functions of the transformed feature space is

$$\{h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_l(\mathbf{x})\},$$

where l denotes the dimension of the transformed feature space. The classification boundary, which is a hyperplane in the transformed feature space, is given by,

$$\{\mathbf{x} : f(\mathbf{x}) = \beta_0 + \mathbf{h}(\mathbf{x})^T \boldsymbol{\beta} = 0\}.$$

Consider the original full feature space $\mathcal{F}_{\mathcal{L}}$ and corresponding index set $\mathcal{L}$. For simplicity and without loss of generality, we assume the first q coordinates are true features (or signals) and $\mathbf{x}^* = (x_1, \cdots, x_q)$, $q < p$, while remaining dimensions $(> q)$ contains just noise. The classification boundary is given by a hyperplane in the true feature space $\mathcal{F}_{\mathcal{S}}$ as

$$\{\mathbf{x}^* : f(\mathbf{x}^*) = \beta_0 + \mathbf{h}(\mathbf{x}^*)^T \boldsymbol{\beta}^* = 0\}.$$

As in Theorem 1 suppose training data $\mathbf{x}^*$ in $\mathcal{F}_{\mathcal{S}}$ are separable, then the margin-maximizing SVM in $\mathcal{S}$ is:

$$Q(q, \mathbf{x}^*, \mathcal{S}) = \max_{\beta_0, \boldsymbol{\beta}^*, \|\boldsymbol{\beta}^*\|=1} D^* \quad \text{subject to}$$
$$y_i\left(\beta_0 + \mathbf{h}(\mathbf{x}_i^*)^T \boldsymbol{\beta}^*\right) \geq D^*, \quad i = 1, \cdots, n$$

where D^* is the shortest distance from the training data to the separating hyperplane in $\mathcal{F}_{\mathcal{S}}$. If training data $\mathbf{x}^*$ in $\mathcal{S}$ are separable, then the whole training data $\mathbf{x}$ in $\mathcal{L}$ are also separable by Theorem 1. The margin-maximizing SVM in $\mathcal{L}$ reduces to:

$$Q(p, \mathbf{x}, \mathcal{L}) = \max_{\beta_0, \boldsymbol{\beta}, \|\boldsymbol{\beta}\|=1} D, \quad \text{subject to}$$
$$y_i\left(\beta_0 + \mathbf{h}(\mathbf{x}_i)^T \boldsymbol{\beta}\right) \geq D, \quad i = 1, \cdots, n$$

where D is the shortest distance from the training data to the separating hyperplane in $\mathcal{F}_{\mathcal{L}}$. A KLR problem in $\mathcal{L}$ can be equivalently stated as obtaining

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + e^{-y_i f(\mathbf{x}_i)}\right) \quad \text{subject to}$$
$$\|\boldsymbol{\beta}\|_2^2 \leq s, \; f(\mathbf{x}_i) = \beta_0 + \mathbf{h}(\mathbf{x}_i)^T \boldsymbol{\beta}, \; i = 1, \cdots, n$$

where β_0 is the bias term and may be omitted for simplification purposes. Let us denote the solution of the above KLR problem as $\tilde{\boldsymbol{\beta}}(s)$. Equivalently for any $\mathcal{S}$ having only first $q(\leq p)$ dimensions of $\mathcal{L}$, the KLR problem in $\mathcal{S}$ is

$$\min_{\beta_0, \boldsymbol{\beta}^*} \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + e^{-y_i f(\mathbf{x}_i^*)}\right) + \lambda \, \|f\|_{\mathcal{H}_k}^2.$$

Above is equivalent to

$$\min_{\beta_0, \boldsymbol{\beta}^*} \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + e^{-y_i f(\mathbf{x}_i^*)}\right) \quad \text{subject to}$$
$$\|\boldsymbol{\beta}^*\|_2^2 \leq s, \; f(\mathbf{x}_i^*) = \beta_0 + \mathbf{h}(\mathbf{x}_i^*)^T \boldsymbol{\beta}^*, \; i = 1, \cdots, n. \tag{8}$$

Let us denote its solution by $\hat{\boldsymbol{\beta}}^*(s)$. Then for the submodel in $\mathcal{S}$ we have two propositions followed by Theorem 2.

PROPOSITION 1: $Q(p, \mathbf{x}, \mathcal{L}) \geq Q(q, \mathbf{x}^*, \mathcal{S})$.

On the other hand, when the true feature space is $\mathcal{F}_{\mathcal{S}}$ and feature information is stored in the first q coordinates of $\mathbf{x}$ and its last $p - q$ coordinates are noises, we can partition $\mathbf{x}_{n\times p} = (\mathbf{x}^*_{n\times q}, \mathbf{x}^{\dagger}_{n\times p-q})$, where $\mathbf{x}^{\dagger}$ is a noise vector. We define a robust version of margin in the whole space $\mathcal{F}_{\mathcal{L}}$ to guard against all possible noises:

$$Q(\mathbf{x}^*, \mathcal{L}) = \max_{\mathbf{x}^{\dagger}} Q(p, (\mathbf{x}^*, \mathbf{x}^{\dagger}), \mathcal{L}),$$

that is, the shortest distance from the training data to the hyperplane separating $\mathbf{x} = (\mathbf{x}^*, \mathbf{x}^{\dagger})$ for all noise $\mathbf{x}^{\dagger}$ cases. Denote the first q coordinates of solution $\boldsymbol{\beta}$ by $\tilde{\boldsymbol{\beta}}^*$.

PROPOSITION 2: $Q(\mathbf{x}^*, \mathcal{L}) = Q(q, \mathbf{x}^*, \mathcal{S})$, and thus $\hat{\boldsymbol{\beta}}^* = \tilde{\boldsymbol{\beta}}^*$.

THEOREM 2: If training data is separable in $\mathcal{S}(\subseteq \mathcal{L})$ and the solution for the equivalent KLR problem in $\mathcal{S}$ and $\mathcal{L}$ are respectively $\hat{\boldsymbol{\beta}}^*(s)$ and $\tilde{\boldsymbol{\beta}}(s)$, then as $s \to \infty$

$$\hat{\boldsymbol{\beta}}^*(s)/s - \tilde{\boldsymbol{\beta}}^*(s)/s \to \hat{\boldsymbol{\beta}}^* - \tilde{\boldsymbol{\beta}}^* = 0.$$

where $\tilde{\boldsymbol{\beta}}^*(s)$ are the first q coordinates of $\tilde{\boldsymbol{\beta}}(s)$.

Theorem 2 shows optimality of the classifier under complete separability assumption in the subspace $\mathcal{F}_{\mathcal{S}}$.

4.1. Feature Import Vector Machine

Since the KLR lies in the heart of the FIVM, first we would like to describe the basic steps of KLR, where we want to minimize

$$H = \frac{1}{n}\sum_{i=1}^{n} \ln\left[1 + e^{-y_i f(\mathbf{x}_i)}\right] + \lambda ||f||^2_{\mathcal{H}_k}. \qquad (9)$$

Let us define,

$$p_i = \frac{1}{1 + e^{-y_i f(\mathbf{x}_i)}}, \text{ for } i = 1, 2, \ldots, n, \qquad (10)$$

$$\mathbf{a} = (a_1, a_2, \ldots, a_n)', \qquad (11)$$

$$\mathbf{p} = (p_1, p_2, \ldots, p_n)', \qquad (12)$$

$$\mathbf{y} = (y_1, y_2, \ldots, y_n)', \qquad (13)$$

$$\mathbf{K} = \left(K(\mathbf{x}_i, \mathbf{x}_j)\right)^n_{i,j=1}, \qquad (14)$$

$$\mathbf{W} = Diag(p_1(1 - p_1), \qquad (15)$$
$$p_2(1 - p_2), \ldots, p_n(1 - p_n)).$$

Following Zhu and Hastie [3] with little abuse of notations, Eq. (9) can be rewritten in matrix notations as

$$H = \frac{1}{n}\mathbf{1}' \ln\left(\mathbf{1} + e^{-\mathbf{y}\cdot\mathbf{Ka}}\right) + \lambda\mathbf{a}'\mathbf{Ka}, \qquad (16)$$

where '·' represents element-wise multiplication. To find $\mathbf{a}$, we set the derivative of H with respect to $\mathbf{a}$ equals to zero and use the Newton-Raphson method to iteratively solve the score equation. With a little bit of algebra, it can be shown that Newton-Raphson step is a weighted least square method

$$\mathbf{a}^{(k)} = \left(\frac{1}{n}\mathbf{K}'\mathbf{WK} + \lambda\mathbf{K}\right)^{-1} \mathbf{K}'\mathbf{Wz},$$

where $\mathbf{a}^{(k)}$ is the value of $\mathbf{a}$ obtained in the kth step and

$$\mathbf{z} = \frac{1}{n}\left(\mathbf{Ka}^{(k-1)} + \mathbf{W}^{-1}(\mathbf{y}\cdot\mathbf{p})\right).$$

Alternatively other minimization method can be used directly to minimize Eq. (9) to obtain $\mathbf{a}$. Quasi-Newton method implemented in IMSL [21] library is an interesting possibility. The advantage of this method is that no differentiation and matrix inversion is necessary. Details of other nonlinear function optimization methods can be found in ref. [22] and reference therein.

4.2. FIVM Algorithm

As mentioned earlier we want to find a subset $\mathcal{S}$ of $\mathcal{L}$ such that it is a good approximation of the full model. For simplicity, we assume that the observations on the $\mathbf{x}$ are normalized and centered, that is $\frac{1}{n}\sum_{i=1}^{n} x_{ij}^2 = 1$ and $\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0$. This also ascertains same scaling factor (θ) across all dimensions. We use a greedy search technique to bypass the huge combinatorial search problem involving too many features. We start with the null model or $\mathcal{S} = \emptyset$, then iteratively build up $\mathcal{S}$ by adding one feature at a time. Our main goal is to add a feature from $\mathcal{L} \setminus \mathcal{S}$ which will produce minimum value of the regularized negative log-likelihood, until it satisfies some stopping criteria.

Algorithm 1: FIVM Algorithm

1. Let $\mathcal{S} = \emptyset$, $\mathcal{L} = \{1, 2, \ldots, p\}$ and $k = 1$.

2. For each $l \in \mathcal{L} \setminus \mathcal{S}$, let

$$f_l(\mathbf{x}) = \sum_{i=1}^{n} a_i K_{\mathcal{S}\cup l}(\mathbf{x}, \mathbf{x}_i), \text{ where}$$

$$K_{\mathcal{S}\cup l}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{\sum_{t\in\mathcal{S}\cup l}(x_{it} - x_{jt})^2}{2\theta^2}\right\}.$$

Define $\mathbf{K_l} = \left(K_{\mathcal{S}\cup l}(\mathbf{x}_i, \mathbf{x}_j)\right)_{i,j=1}^{n}$. Use Newton-Raphson or other function minimization method to find $\mathbf{a}$ which minimizes,

$$H(\mathbf{x}_l) = \frac{1}{n}\mathbf{1}' \log\left(\mathbf{1} + e^{-\mathbf{y}\cdot\mathbf{K_l a}}\right) + \lambda \mathbf{a}'\mathbf{K_l a}. \tag{17}$$

3. Find $l^\star$ such that

$$l^\star = \underset{l\in\mathcal{L}}{\mathrm{argmin}} H(\mathbf{x}_l) \tag{18}$$

Let $\mathcal{S} = \mathcal{S} \cup \{l^\star\}$, $\mathcal{L} = \mathcal{L} \setminus \{l^\star\}$, $H_k = H(\mathbf{x}_{l^\star})$ and $k = k+1$.

4. Repeat Steps 2 and 3 until convergence criteria are satisfied.

The dimensions in $\mathcal{S}$ are called imported features.

4.3. Convergence Criteria

In their original IVM algorithm, Zhu and Hastie [3] compared the quantity H_k in different iterations. At step k, they compare H_k with $H_{k-\Delta k}$, where Δk is a pre-chosen small integer, say $\Delta k = 1$. If the ratio $|\frac{H_k - H_{k-\Delta k}}{H_k}|$ is less than a pre-chosen small number ϵ, say $\epsilon = 0.001$, the algorithm stops adding new observations. This convergence criterion is fine in IVM context as their algorithm compares individual observations without altering dimensions. For FIVM in a specific iteration different $H(\mathbf{x}_l)$'s (Step 2 of the Algorithm 1) are comparable as dimension remains fixed. However, individual H_k's are not comparable across different iterations as each of them becomes trans-dimensional quantity. For this kind of trans-dimensional comparison we need to think of a quantity independent of the dimensional effect. To do so in Step 3 of the Algorithm 1, at the kth iteration after selecting $l^\star$ we compute $p_k = p_{l^\star}$ defined as the proportion of correctly classified training observations with k imported features. If the ratio $|\frac{p_k - p_{k-\Delta k}}{p_k}|$ is less than a pre-chosen small number ϵ, say $\epsilon = 0.001$, the algorithm stops adding new features.

Though it was not discussed in their original paper we would like to mention that the convergence criterion described in ref. [3] has a mild assumption of no repetition of observations to be successfully applicable. In case there are two (or more) identical data points and $\Delta k = 1$ is chosen and if it turns out that in one of those identical points is also an imported observation in any step of the iteration, then the algorithm will stop there as it will produce $|\frac{H_k - H_{k-\Delta k}}{H_k}| = 0 < \epsilon$. For FIVM we make two similar assumptions. First, there are no two features having exactly the same value for all observations in the training set. Second, there exists no feature having a constant value throughout. It is trivial to prove that for these cases $|\frac{p_k - p_{k-\Delta k}}{p_k}| = 0 < \epsilon$, hence the algorithm will not be successful. Note that for very small sample size p_k could be an unstable quantity. While this is true, the way we define relative ratio of p_k and $p_{k-\Delta k}$ (to compare with ϵ) will somewhat normalize this numerical stability issue.

4.4. Choice of the Regularization Parameter λ

Our goal is to construct a classifier that will not only reduce the misclassification error in training set, but also produce good overall generalization capability. To choose optimum value of λ we closely follow the computational consideration of Zhu and Hastie [3]. Following their suggestions we randomly split all the data into a training set and a tuning set. We choose optimal value of λ by decreasing it from a larger value to a smaller value until we hit the optimum misclassification value in the tuning set. Details of these steps could be found in ref. [3] and hence not repeated here.

5. EXPERIMENTAL RESULT

In this section we test the FIVM algorithm on few experimental data sets. Our first data set is a synthetic one, where we deliberately introduce few extra features that essentially contain white noise. Our goal is to see whether the algorithm can filter the informative features from the noisy ones. Second, we consider a real-life genomic breast cancer data studied earlier in classification context through singular value decomposition (SVD) by West *et al.* [23]. Finally we consider Colon Cancer data set of Alon *et al.*[24], who studied and benchmarked extensively in classification and clustering context.

5.1. Exploration with Synthetic Data

The data is generated following mixture of normal distributions. Following the suggestion of Hastie *et al.* [3,25], we first generate 10 means $\boldsymbol{\mu}_k$ from a bivariate normal distribution $N((1.5, 0)', 2\mathbf{I})$ and label this class as $+1$. Similarly we generate another 10 means from $N((0, 1.5)', 2\mathbf{I})$ and label them as -1. Then from each class we generate 100 observations as follows: For each observation a mean μ_k is chosen with probability 1/10, and then generate a $N(\mu_k, \mathbf{I}/5)$, which leads to a mixture of normal clusters for each class. Next for each observations we deliberately add another eight dimensions and fill them with white noise. So our final data set has 200 observations

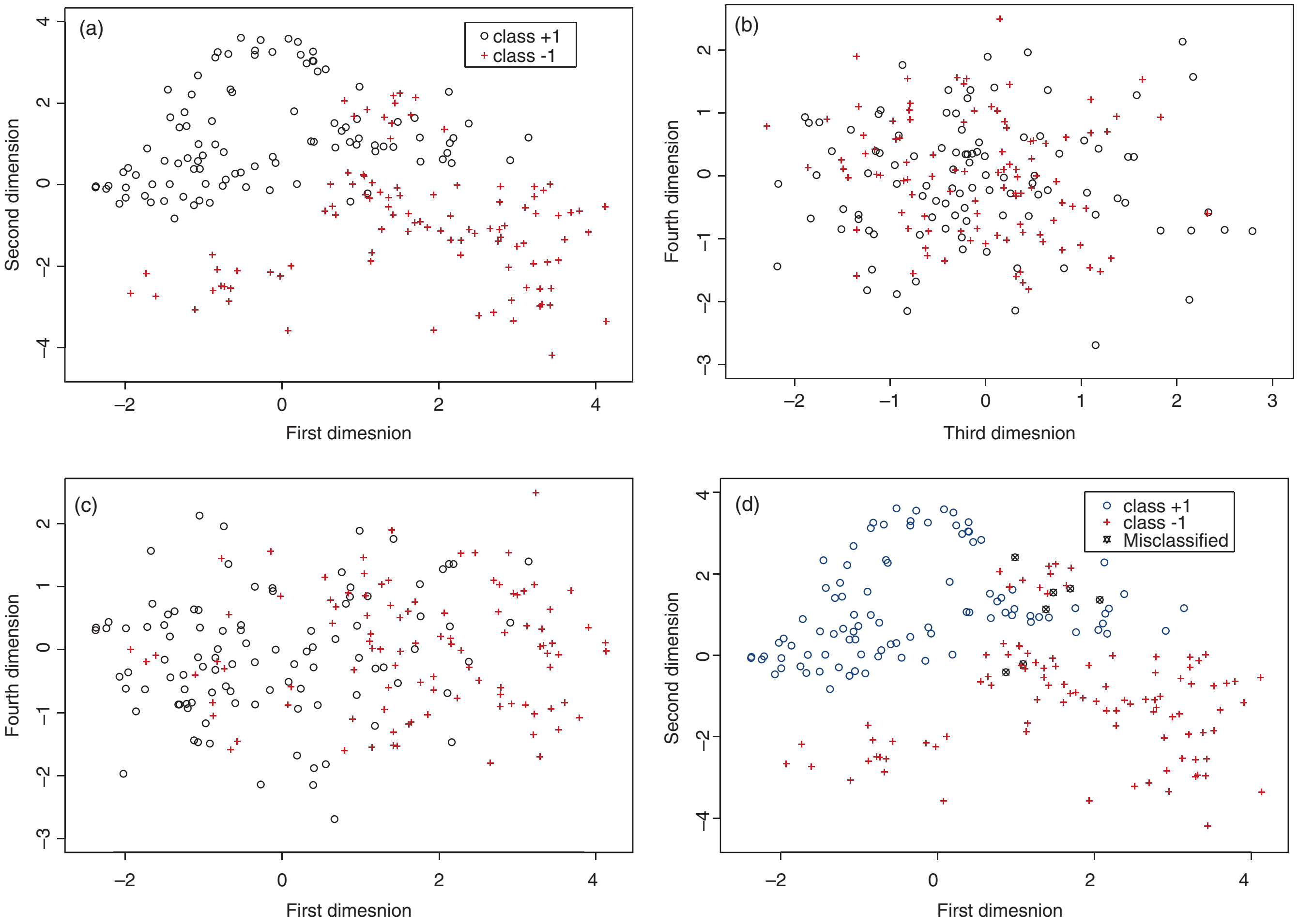

Fig. 2 Scatter plots of the synthetic data over different dimensions. Plot (a) is between first two dimensions. Plot (b) and plot (c) include noisy dimensions. Clearly last two scatter plots do not represent any classification pattern, as expected. Plot (d) represents data points after classification. Only dimensions selected (first and second one) by the FIVM algorithm is used for final classification. Misclassification rate in training set is 0.105. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

and each observation has ten dimensions/features. Scatter plots for these data points are represented in Fig. 2(a)– (c).

We expect FIVM algorithm to filter out informative features from the noisy ones. Figure 3(a) and Table 1 show how the tuning parameter λ is chosen. The optimal λ is found to be 0.5 which corresponds to misclassification rate in training set as 0.105. Notably only first two dimensions are selected and we keep $\epsilon = 0.05$ fixed. Figure 3(b) represents the number of selected dimensions for different values of ϵ. If we choose very small value of $\epsilon (< 0.0001)$ naturally all the dimensions will be selected on the other hand for very large value of ϵ only one dimension will be enough. Clearly from the graph optimal cutoff value will lie between 0.02 and 0.06, which justifies our choice for $\epsilon = 0.05$. Figure 3(c) represents the training error rate if we forcefully add more dimension by lowering ϵ value. There is very minor improvement when more than two dimensions are selected as they contain white noise.

We next present testing data set in Fig. 4(a). Testing data is generated in similar fashion as that of training data. However only those dimensions (namely first two) are used for classification which are selected in the training data by FIVM algorithm. It should be noted that the order of the informative dimensions is a nonissue as FIVM will search and select (see Step 3 of the Algorithm 1) those dimensions sequentially which decreases regularized NLL the most. Figure 4(b) represents plot after classification. In Table 1 we summarize the performance of FIVM, SVM and L_1 penalized logistic regression. The scheme of representing testing/training error is following: number without bracket represents the mean while inside the bracket (if any) is standard error. Mean testing error for FIVM is 0.125. Notably for SVM with Gaussian kernel this is 0.18. For L_1 penalized logistic regression testing error is 0.145. Ordinary SVM will not do any dimension selection and performance of SVM does not improve if we include all dimensions in

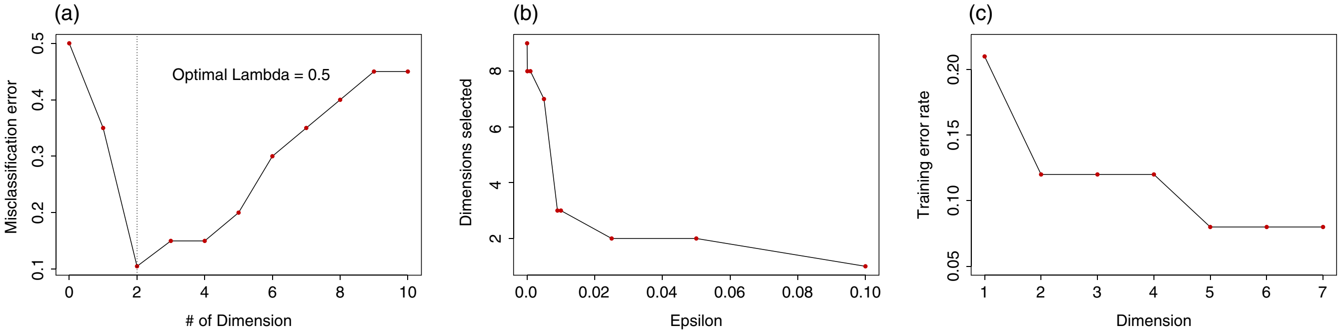

Fig. 3 Radial basis kernel is used throughout, $n = 200$, $\theta = 1$, $\Delta k = 1$. The stopping criterion is satisfied when $|\mathcal{S}| = 2$. For plot (c), ϵ value is adjusted so that desired number of dimensions could be chosen. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

the present setup. We next increase the number of testing sample size to see if the testing error rate gets effected for large number of data points. Figure 4(c) represents a plot for testing error rate for different number of testing data set. As seen in this plot, testing error rate is a decreasing function of sample size. To consider the generalization capability of FIVM further, we also generated ten separate synthetic training and test data sets each with 200 observations with different random seeds as described in Section 5.1. The average training error for FIVM is 0.106(±0.016) while average testing error is 0.175(±0.019). In each case FIVM is able to select two informative dimensions for appropriate choice of λ. We would like to mention in the synthetic data set we have considered $p = 10$, with only two informative dimensions. To consider $n < p$ scenario it is possible to add as many noisy dimensions desired. We would like to report addition of these noisy dimensions will not affect FIVM's feature selection accuracy, thus making it useful for both $n < p$ and $p < n$ scenario. In Table 1 we also report the exact runtime based on a single processor Windows-7-based machine. FIVM is slower than L_1 penalized logistic regression but its performance is comparable to SVM. This is not surprising as both SVM and FIVM are nonlinear Techniques; however, in our experience this computation cost is not too prohibitive.

Table 1. Testing performance of SVM, FIVM and L_1 logistic regression. SVM will not do any dimension selection, so dimensions are selected first by FIVM. L_1 logistic regression is also able to select two informative features for appropriate λ. For all cases tuning parameters are selected through extensive grid search and via tenfold cross-validation over the training set. We searched over the range $[2^{-6}, 2^{6}]$ for θ and $[2^{-10}, 2^{10}]$ for λ.

SVM performance	FIVM performance	L_1 logistic regression
$\theta = 2$, $\lambda = 4$, Train error = 0.125	$\theta = 1$, $\lambda = 0.5$, Train error = 0.105	$\lambda = 0.1$, Train error = 0.12
Test error= 0.18, 36 Support points	Test error= 0.125	Test error= 0.145
Run time = 55 s	Run time = 51 s	Run time = 31 s

5.2. Exploration with Breast Cancer Data

We next test our algorithm with a real-life data set. This data set is studied earlier by West *et al.*[23] in classification context through SVD with stochastic regularization by using Bayesian analysis. Their basic idea was to explore whether gene expression information generated by DNA microarray analysis of human tumors can provide molecular phenotyping that identifies distinct tumor classifications. Tumors were either positive for both the estrogen and progesterone receptors or negative for both receptors. The final collection of tumors consisted of 13 estrogen receptor (ER)+lymph node (LN)+tumors, 12 ER−LN+tumors, 12 ER+LN−tumors and 12 ER− LN−tumors. Other biological details can be found in the original paper of West *et al.* [23]. The initial 49 tumors were classified as ER+ or ER− via immunohistochemistry (IHC) at the time of diagnosis and then later via protein immunoblotting assay for ER to check the IHC results. As reported in their paper in five cases these results are found conflicting. We have selected these five and another nine samples randomly to create our test set. So on the basis of ER status our training set consists of 35 samples while the test set consists of 14 samples. Each sample consists of 7129 gene probes.

A second data set consists of the same set of genetic probes but an outcome variable based on Lymph Node status is also studied. We consider both data sets for determining efficacy of FIVM algorithm. It is customary in many of the classification papers for the large p, small n domain to screen out only a few genes based on some preliminary analysis, thus reducing dimensionality. In their paper West *et al.* [23] described a simple screen to identify the 100 genes maximally correlated with outcome. This choice often made for convenience rather than being

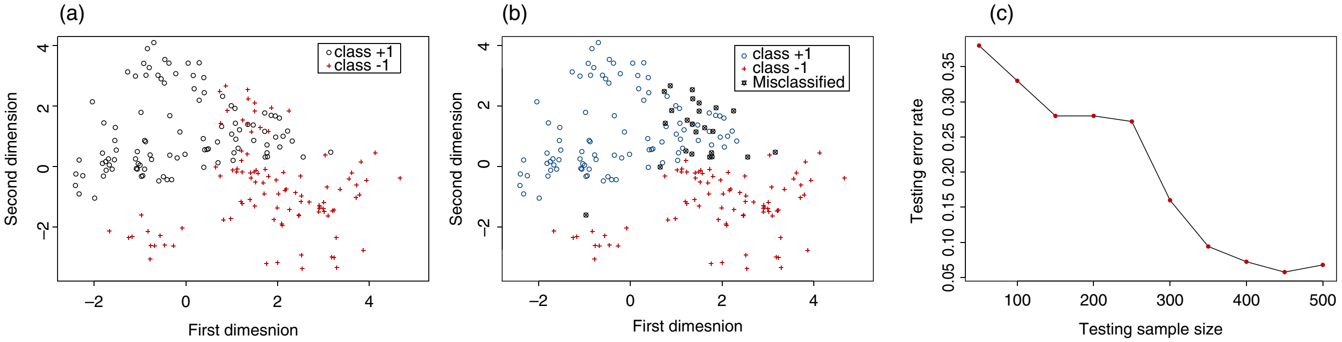

Fig. 4 Plot (a) shows scatter plots of the synthetic testing data over selected dimensions. Plot (b) represents data points after FIVM classification. Plot (c) shows misclassification rate as a function of sample size. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

confirmatory. We consider an alternative approach through FIVM algorithm by means of which we like to select maximally important subset of genes. The normalized data set is divided into training and testing data set as described earlier. Tuning parameters are chosen via tenfold cross-validation and grid search similar to the synthetic data set over the training set and found to be $\lambda = 0.5$ and $\theta = 1$. For each experiment we explore two different values of the convergence parameter ϵ namely, 0.05 and 0.001 with $\Delta k = 1$.

5.2.1. *Classification on the basis of estrogen receptor status*

For $\epsilon = 0.05$ only four genes are selected while for $\epsilon = 0.001$ six genes are selected. Selected genes and details of the results are presented in Table 2. The number outside the bracket indicates mean bootstrap error (over 20 replications) over the test set and the number inside the bracket is standard deviation. Though testing accuracy is not high for FIVM, it consistently outperforms SVM and additionally performs feature selection. We next present four three-dimensional (3D) scatter plots in Fig. 5 based on the selected set of genes. Figure 5(a) represents a scatter plot for training data set based on the first three selected genes which decreases regularized NLL most. Figure 5(b) represents a scatter plot based on second set of three selected genes. Similar plots for test set are drawn in Fig. 5(c) and (d). Notably out of top six genes selected by FIVM, four are found to be common with that of West *et al.* [23].

Table 2. Estrogen receptor (ER) classification result for breast cancer data.

$\epsilon = 0.05$	$\epsilon = 0.001$
Four genes selected by FIVM	Six genes selected by FIVM
Z84721_cds2, Contains alpha and zeta globin genes and ESTs	Z84721_cds2, Contains alpha and zeta globin genes and ESTs
X83425, Homo Sapiens LU gene for Lutheran blood group glycoprotein	X83425, Homo Sapiens LU gene for Lutheran blood group glycoprotein
X55037, Homo Sapiens GATA-3 mRNA	X55037, Homo Sapiens GATA-3 mRNA
U33147, Human mammaglobin	U33147, Human mammaglobin
mRNA, complete cds	mRNA, complete cds
	HG1205-HT1205, Collagen, Type IV, Alpha 2, N-Terminus
	X03635, Human mRNA for oestrogen receptor
Testing error= 0.344(±0.032)	Testing error= 0.416(±0.027)
SVM Testing performance & Tuning parameters	
$\theta = 0.5$, $\lambda = 1$	$\theta = 1$, $\lambda = 1$
Error=0.406(±0.052), 12(±4) support points	Error=0.41(±0.054), 20(±3) support points

5.2.2. *Classification on the basis of lymph node status*

We have performed similar analysis for lymph node (LN) data set. For two different choices of $\epsilon = 0.05$ and $\epsilon = 0.001$ three and five genes are selected respectively. Selected genes and details of the results are presented in Table 3. Notably mean bootstrap testing accuracy for SVM is close to 0.475, implying that it is similar to a naive random classifier. We next present four 3D scatter plots in Fig. 6 based on selected set of genes. Figure 6(a) represents a scatter plot for training data set based on the first three selected genes which decreases regularized NLL most. Figure 6(b) represents a scatter plot based on second set of three selected genes. Similar plots for test set are drawn in Fig. 6(c) and (d).

5.3. Exploration with Colon Cancer Data

Alon *et al.*[24] described a gene expression profile for 40 tumor and 22 normal colon tissue samples, analyzed with an Affymetrix oligonucleotide array which

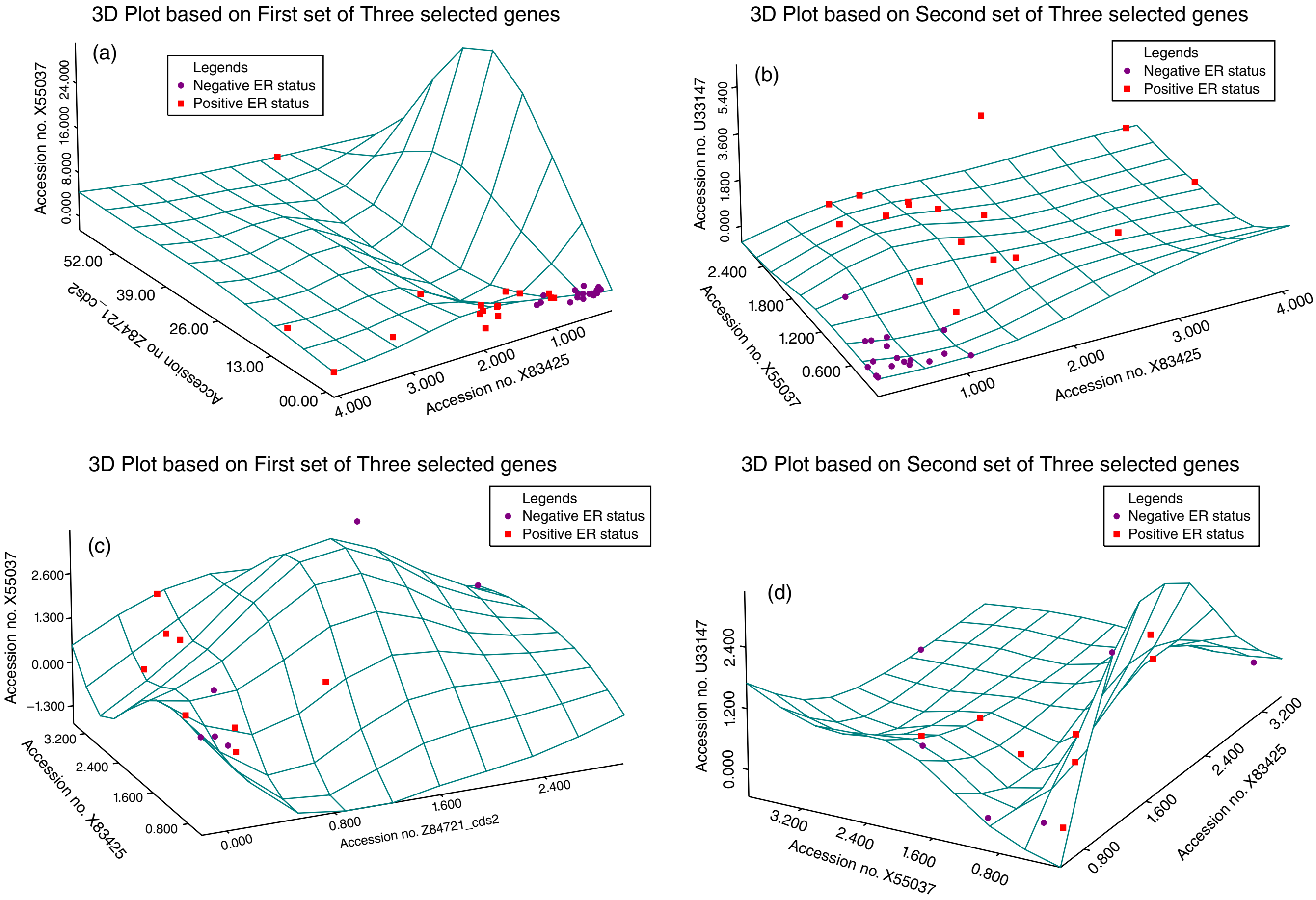

Fig. 5 Scatter plots of the breast cancer data set (ER status). Plot (a) and (c) are drawn among first three selected genes which decreases regularized NLL the most. Plot (b) and (d) are drawn among second set of three selected genes. Each gene is marked by the accession number in each axis. Top two plots (a and b) are for training set and bottom two plots (c and d) are for test set. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

is complementary to more than 6500 human genes and expressed sequence tags (ESTs). Colon adenocarcinoma specimens were collected from patients and mRNA was extracted and hybridized to the array. The Affymetrix Hum6000 array contains about 65 000 features. For details regarding array preparation we would like to refer to original paper of Alon *et al.*[24]. Final data set contains intensities of 2000 genes in 22 normal and 40 tumor colon tissues. The genes chosen are with highest minimal intensity across the samples. Each gene was also normalized based on all observations. The data set can be downloaded with all complementary information from the website, http://microarray.princeton.edu/oncology/affydata/index.html.

This data set is heavily studied and benchmarked [5,26–28] in different context by many researchers extensively. FIVM algorithm is used to find its efficacy in gene screening and classification accuracy. We first divide them into training set and testing set by allocating normal and tumor samples randomly in the test and training set respectively. The resulting training set contains 50 observations while the testing data set has 12 observations. Normalized training data set is then presented to FIVM algorithm. Tuning parameter λ is found to be 0.35, with $\theta = 2$ via cross-validation and we choose $\Delta k = 1$. For this experiment we choose $\epsilon = 0.001$ as the value of the convergence parameter. A lower value of ϵ will of course select more number of genes. Only three genes are selected by the algorithm with mean bootstrap testing error 0.258. The selected genes and details of the result are presented in Table 4. Notably the three selected genes has appeared in many different citations as the top significant genes that can differentiate between normal and cancer tissue samples [29,30]. FIVM again beats SVM in testing accuracy. Interestingly this data set is analyzed using similar training (50 of 62) and testing (12 of 62) split-up with 15 selected genes by Weston *et al.*[5] and they report 12.8% testings error (no variance is reported). While they do not identify which genes are selected, we report similar accuracy with

Table 3. Lymph node (LN) classification result for breast cancer data.

$\epsilon = 0.05$	$\epsilon = 0.001$
Three genes selected by FIVM	Five genes selected by FIVM
HG3638-HT3849_s_at Amyloid Beta (A4) Precursor Protein, Alt. Splice 2,	HG3638-HT3849_s_at Amyloid Beta (A4) Precursor Protein, Alt. Splice 2,
U02493, Human 54 kDa protein mRNA, complete cds	U02493, Human 54 kDa protein mRNA, complete cds
U51678, Human small acidic protein mRNA, complete cds	U51678, Human small acidic protein mRNA, complete cds
	HG3432-HT3620_s_at Fibroblast Growth Receptor K-Sam, Alt. Splice 3
	X80692, Homo Sapiens ERK3 mRNA
Testing error = 0.304(±0.017)	Testing error = 0.411(±0.022)
SVM Testing Performance & Tuning Parameters	
$\theta = 0.25$, $\lambda = 2$	$\theta = 0.5$, $\lambda = 4$
Error = 0.475(±0.012), 25(±4) support points	Error = 0.475(±0.012), 25(±4) support points

only three genes. Selecting additional genes by lowering the values of ϵ will of course improve the accuracy of our method. Next we present two 3D scatter plots in Fig. 7 based on the selected set of genes for both training and testing sets.

6. MULTICLASS EXTENSION OF FIVM

FIVM essentially follows KLR to select a submodel, hence it is readily extendible for multiclass case. We briefly outline the algorithm for multiclass FIVM (MFIVM), which is again adopted for dimension filtering from multiclass IVM described in ref. [3]. Following Section 3, for multiclass case the class label y assumes value from a finite set $C = \{1, \cdots, M\}$. Let us denote the conditional probability of a point being classified in category $c (\in C)$ as $p_c(\mathbf{x}) = P(Y = c|\mathbf{X} = \mathbf{x})$. Then Bayes classification rule is given by

$$c(\mathbf{x}) = \underset{c \in C}{\operatorname{argmax}} \quad p_c(\mathbf{x}) = \underset{c \in C}{\operatorname{argmax}} \quad \frac{e^{f_c(\mathbf{x})}}{\sum_{k=1}^{M} e^{f_k(\mathbf{x})}}, \quad (19)$$

where $f_c(\mathbf{x}) \in \mathcal{H}_k$; $\mathcal{H}_k$ is RKHS generated by the positive definite kernel $K(.,.)$. As noted in ref. [3] above model is not identifiable for a location shift of $f_c(\mathbf{x})$ and we need to put an identifiability constraint

$$\sum_{c=1}^{M} f_c(\mathbf{x}) = 0.$$

Following the similar path of Section 4.1 we first introduce multiclass KLR, in which we minimize regularized NLL of the multinomial likelihood represented as

$$\begin{aligned} H &= -\frac{1}{n} \sum_{i=1}^{n} \ln p_{\mathbf{y}_i}(\mathbf{x}_i) + \lambda ||\mathbf{f}||^2_{\mathcal{H}_k} \\ &= -\frac{1}{n} \sum_{i=1}^{n} \left[\mathbf{y}_i \mathbf{f}(\mathbf{x}_i) - \ln \left(e^{f_1(\mathbf{x}_i)} \ldots + e^{f_M(\mathbf{x}_i)} \right) \right] \\ &\quad + \lambda ||\mathbf{f}||^2_{\mathcal{H}_k}, \end{aligned} \quad (20)$$

where $\mathbf{y_i}$ is a M dimensional vector with all zero elements except a 1 in position c indicating which class the ith observation belongs to and

$$\mathbf{f}(\mathbf{x}_i) = (f_1(\mathbf{x}_i), \ldots, f_C(\mathbf{x}_i))', \quad (21)$$

$$||\mathbf{f}||^2_{\mathcal{H}_k} = \sum_{c=1}^{M} ||f_c||^2_{\mathcal{H}_k}. \quad (22)$$

Again using representer theorem of Kimeldorf and Wahba [7], it can be shown that $f_c(\mathbf{x})$ minimizes (20) and is given by

$$f_c(\mathbf{x}) = \sum_{i=1}^{n} a_{ic} K(\mathbf{x}_i, \mathbf{x}). \quad (23)$$

Following Zhu and Hastie [3] with little abuse of notations, Eq. (20) can be rewritten in matrix notations as

$$\begin{aligned} H &= \frac{1}{n} \sum_{i=1}^{n} \left[-\mathbf{y}_i' (\mathbf{K}(i,)\mathbf{A})' + \log \left(\mathbf{1}' e^{(\mathbf{K}(i,)\mathbf{A})'} \right) \right] \\ &\quad + \lambda \sum_{c=1}^{M} a_c{}' \mathbf{K} a_c, \end{aligned} \quad (24)$$

where $\mathbf{A} = (a_1, \ldots, a_M)$ and $\mathbf{K}$ is defined as earlier in the two-class problem. With the above setup we would like to apply FIVM algorithm described in Section 4.2 for selecting out important features. Computational burden of MFIVM is similar to two-class FIVM and is proportional with the number of class labels. In Fig. 8, we present a simulation result for MFIVM with three classes and with two informative dimensions. The data in each class are generated from a mixture of Gaussians [25].

7. SUMMARY AND CONCLUSIONS

Classification literature is pretty rich both from probabilistic and deterministic domain. In this article we have

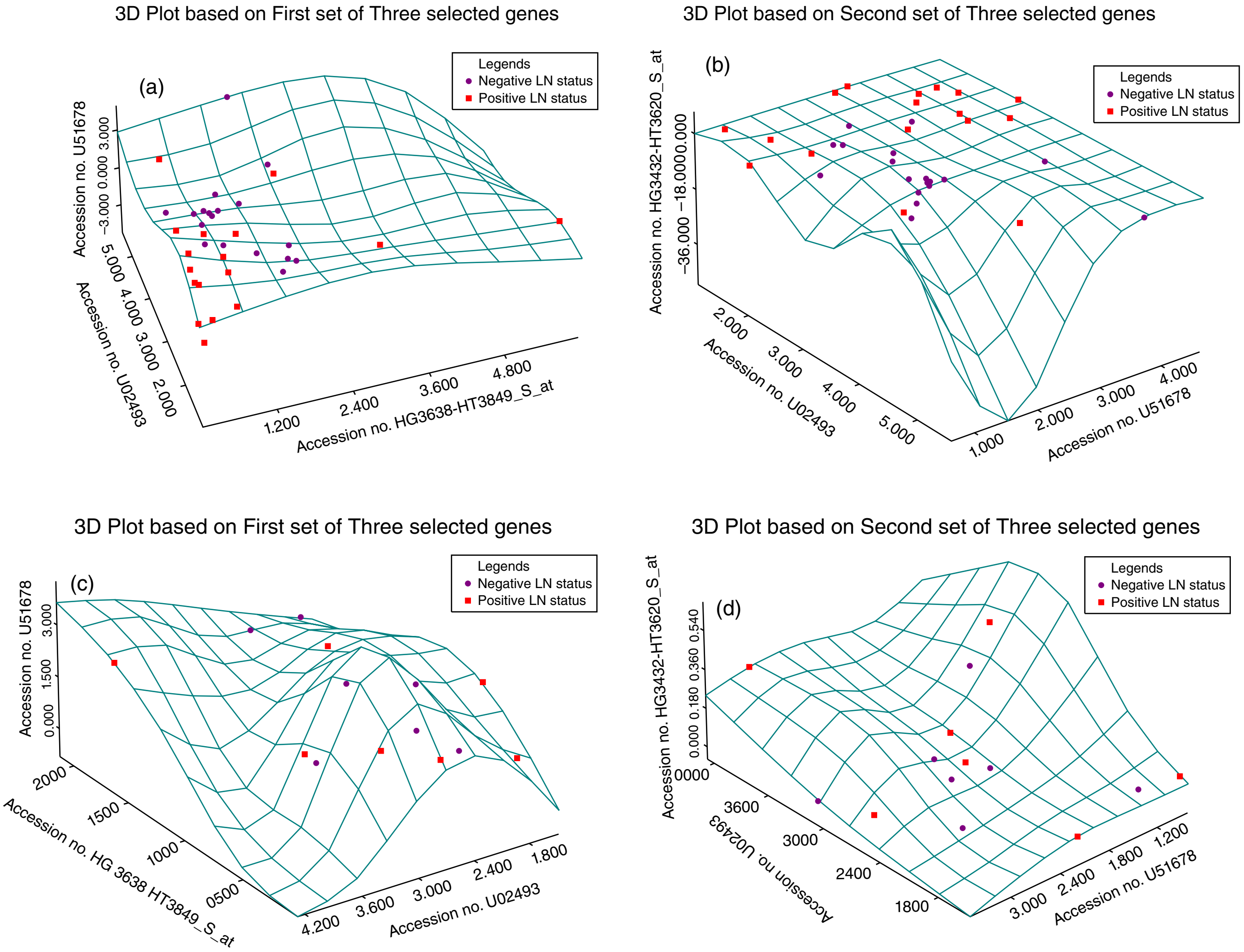

Fig. 6 Scatter plots of the breast cancer data set (LN status). Plot (a) and (c) are drawn among first three selected genes which decreases regularized NLL the most. Plot (b) and (d) are drawn among second set of three selected genes. Each gene is marked by the accession number in each axis. Top two plots (a and b) are for training set and bottom two plots (c and d) are for test set. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

developed a classification method based on RKHS principle. We first select useful features that decrease the regularized NLL most. The final classifier is then constructed on the transformed feature space but by using only the selected features. This leads to great data reduction as well as nonlinear classification. We have tested efficacy of our algorithm for synthetic as well as real-life data examples.

One future problem for classification in high dimension will be when dimensions are having non-ignorable association among each other. Notably this cross-correlation information is not being utilized in the dimension selection algorithm developed in this article. Another exciting future possibility is simultaneous selection of feature and observation especially in the *Big data* context. With appropriate regularization scheme this may lead to huge data compression with minimal loss in classification accuracy.

Table 4. Classification result for colon cancer data Set.

$\epsilon = 0.001$
Three genes selected by FIVM
T71025 : Human (HUMAN)
M76378 : Human cysteine-rich protein (CRP) gene, exons 5 and 6.
M63391 : Human desmin gene, complete cds.
Testing error= 0.153($\pm$0.033)
SVM testing performance & Tuning parameters
$\theta = 0.25$ & $\lambda = 8$
Testing error=0.22($\pm$0.047), 10($\pm$3) support points

8. ACKNOWLEDGMENTS

The research of first author was partly supported by NIH grant number P30 ES020957. The research of the second author was partly supported by NSF grant numbers DMS-10-5635 and DMS-12-65203. Authors would like to thank

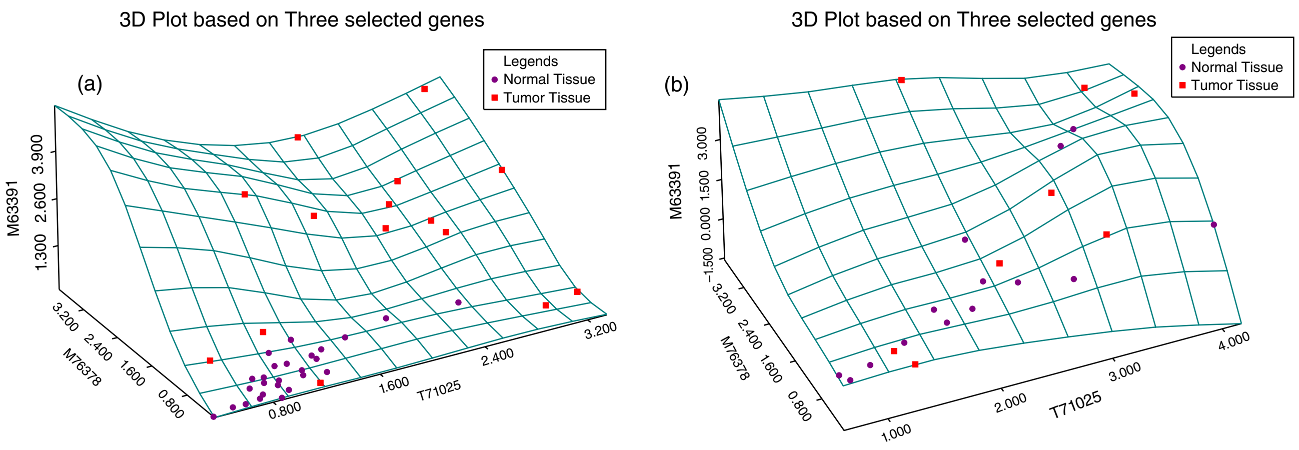

Fig. 7 Scatter plots of the colon cancer data set. Plots are drawn on the basis of three selected genes. Each gene is marked by the gene number in each axis. Plot (a) is for training set and plot (b) is for testing set. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

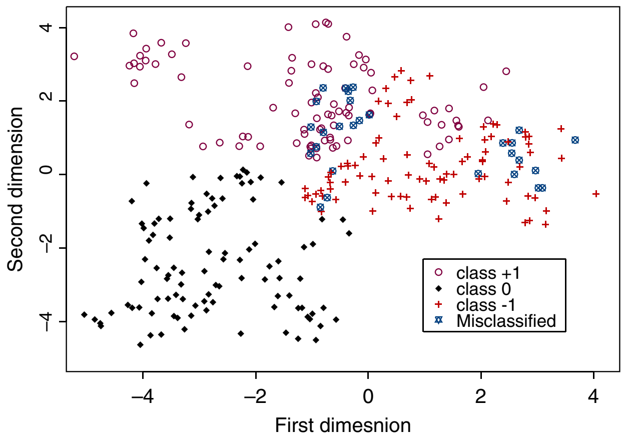

Fig. 8 Radial kernel is used. $C = 3$, $n = 300$, $\lambda = 1.5$, training error $= 0.089$, testing error $= 0.1$. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Dr Dipak Dey for his comments. Authors would also like to thank two anonymous referees and the editorial team, whose comments provided additional insights and have greatly improved the scope and presentation of the article.

APPENDIX

PROOF OF THEOREM 1

Since n training observations are completely separable in $\mathcal{S}$ we have

$$y_i f_{\mathcal{S}}(\mathbf{x}_i) = y_i \sum_{j=1}^{n} \alpha_j K_{\mathcal{S}}(\mathbf{x}_i, \mathbf{x}_j) > 0, \text{ for } i = 1, 2, \ldots, n.$$

Proof: For RBF it is easy to show $K_{\mathcal{L}}(\mathbf{x}_i, \mathbf{x}_j) = K_{\mathcal{S}}(\mathbf{x}_i, \mathbf{x}_j) K_{\mathcal{L}\setminus\mathcal{S}}(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathcal{S} \subseteq \mathcal{L}$ and RBF is a bounded kernel having the property $0 \le K_\Lambda(\mathbf{x}_i, \mathbf{x}_j) \le 1$ for any set Λ and $K_\Lambda(\mathbf{x}_i, \mathbf{x}_j|\theta) = \exp\left\{-\frac{\sum_{t\in\Lambda}(x_{it}-x_{jt})^2}{2\theta^2}\right\}$. Similarly for the set $\mathcal{L}$

$$\begin{aligned} y_i f_{\mathcal{L}}(\mathbf{x}_i) &= y_i \sum_{j=1}^{n} \beta_j K_{\mathcal{L}}(\mathbf{x}_i, \mathbf{x}_j) \\ &= y_i \sum_{j=1}^{n} \beta_j K_{\mathcal{S}}(\mathbf{x}_i, \mathbf{x}_j) K_{\mathcal{L}\setminus\mathcal{S}}(\mathbf{x}_i, \mathbf{x}_j). \end{aligned} \tag{A.1}$$

Since $\mathcal{S} \subseteq \mathcal{L}$, if we take all dimensions in $\mathcal{L} \setminus \mathcal{S}$ as constant (i.e., providing no information) for all observations, then $\sum_{t\in\mathcal{L}\setminus\mathcal{S}}(x_{it} - x_{jt})^2 = 0$. This implies $K_{\mathcal{L}\setminus\mathcal{S}}(\mathbf{x}_i, \mathbf{x}_j) = 1$. Now if we take $\beta_j = \alpha_j$ for $j = 1, 2, \ldots, n$, then $y_i f_{\mathcal{L}}(\mathbf{x}_i) > 0$ for $i = 1, 2, \ldots, n$ and hence separable in $\mathcal{L}$ too. Notably we are not saying in real life $\sum_{t\in\mathcal{L}\setminus\mathcal{S}}(x_{it} - x_{jt})^2 = 0$ need to hold, in fact it will not be true in general. However since we already have complete separability in $\mathcal{S}$, suppressing all $\mathcal{L} \setminus \mathcal{S}$ dimensions by some constant will render $\sum_{t\in\mathcal{L}\setminus\mathcal{S}}(x_{it} - x_{jt})^2 = 0$, hence the lower dimensional classifier in $\mathcal{S}$ will still achieve a complete separation in $\mathcal{L}$. ■

PROOF OF THEOREM 2

We begin with an important theorem of Zhu and Hastie [3] which relates to the solution of KLR and SVM in an unified framework of margin maximizing loss function. We omit the subscript for denoting dimensional set, as in their original theorem dimension is not assumed to be a varying quantity, rather solutions for equivalent SVM and KLR problem considered only.

THEOREM 1: If training data is separable, that is $\exists\beta_0, \beta$, s.t. $y_i(\beta_0 + \sum_{j=1}^{n} \beta_j K(\mathbf{x}_i, \mathbf{x}_j)) > 0$, $\forall i$ and if the solution for the KLR problem stated earlier is denoted as $\tilde{\beta}(s)$, then

$$\frac{\tilde{\beta}(s)}{s} \to \beta^\star \text{ as } s \to \infty,$$

where $\beta^\star$ is the solution of the margin-maximizing SVM, provided $\beta^\star$ is unique. If $\beta^\star$ is not unique, then $\frac{\tilde{\beta}(s)}{s}$ may have multiple convergence points, but they will all represent margin-maximizing separating hyperplanes.

Regarding the proof of the theorem we refer to refs [3,14] . This theorem tells us that margin-maximizing property is shared by KLR and SVM and their asymptotic solution is consistent. Again as stated earlier in Section 4, for any $\mathcal{S}$ having only first $q(\leq p)$ dimensions of $\mathcal{L}$, the KLR problem in $\mathcal{S}$ is given in Eq. 8. Denote its solution by $\hat{\beta}^*(s)$. Then by Theorem 1, as $s \to \infty$,

$$\frac{\hat{\beta}^*(s)}{s} \to \hat{\beta}^\star.$$

Again for the complete space $\mathcal{F}_\mathcal{L}$ having p dimensions, by Theorem 1, we have as $s \to \infty$,

$$\frac{\tilde{\beta}(s)}{s} \to \beta^\star.$$

Let $\tilde{\beta}^*(s)$ be the first q coordinates of $\tilde{\beta}(s)$. Then as $s \to \infty$,

$$\frac{\tilde{\beta}^*(s)}{s} \to \tilde{\beta}^\star.$$

We will give the proof of Theorem 2 through two propositions.

Proof of Proposition 1

Proof: It is a consequence of Theorem 1 and the definitions of $Q(p, \mathbf{x}, \mathcal{L})$ and $Q(q, \mathbf{x}^*, \mathcal{S})$ ■

Proof of Proposition 2

Proof: Suppose training data $\mathbf{x} = (\mathbf{x}^*, \mathbf{x}^\dagger)$ are separable in $\mathcal{L}$. Since $\mathcal{S} \subset \mathcal{L}$. Denote

$$\mathcal{L} = \mathcal{S} + \mathcal{S}^+.$$

We show that a hyperplane in $\mathcal{L}$ separating $\mathbf{x} = (\mathbf{x}^*, \mathbf{x}^\dagger)$ for all possible noise $\mathbf{x}^\dagger$ must perpendicularly pass through a hyperplane in $\mathcal{S}$ that separates $\mathbf{x}^*$.

Because the noise $\mathbf{x}^\dagger$ is random and may take all possible values, it is not possible to separate $\mathbf{x} = (\mathbf{x}^*, \mathbf{x}^\dagger)$ using only $\mathbf{x}^\dagger$, and any hyperplane ℓ in $\mathcal{L}$ separating $\mathbf{x} = (\mathbf{x}^*, \mathbf{x}^\dagger)$ for all possible $\mathbf{x}^\dagger$ must also separate $\mathbf{x}^*$ in $\mathcal{S}$. That is, any hyperplane in $\mathcal{L}$ separating $\mathbf{x} = (\mathbf{x}^*, \mathbf{x}^\dagger)$ for all possible $\mathbf{x}^\dagger$ must pass a separating hyperplane in $\mathcal{S}$.

We next show that if the hyperplane separates $\mathbf{x} = (\mathbf{x}^*, \mathbf{x}^\dagger)$ for all possible noise $\mathbf{x}^\dagger$, it has to be perpendicular to $\mathcal{S}$. Otherwise, suppose a hyperplane ℓ in $\mathcal{L}$ is not perpendicular to $\mathcal{H}$. Then ℓ is tilted to some coordinates of $\mathbf{x}^\dagger$, say these coordinates are x_r, $r > q$. We may select large positive values for these $|x_r|$, such that $\mathbf{x} = (\mathbf{x}^*, \mathbf{x}^\dagger)$ and $\mathbf{x} = (\mathbf{x}^*, -\mathbf{x}^\dagger)$ fall in two sides of the hyperplane ℓ. Thus, the hyperplane ℓ cannot separate $\mathbf{x} = (\mathbf{x}^*, \mathbf{x}^\dagger)$ for both $\mathbf{x}^\dagger$ and $-\mathbf{x}^\dagger$. Therefore, the hyperplane in $\mathcal{L}$ separating $\mathbf{x} = (\mathbf{x}^*, \mathbf{x}^\dagger)$ for all possible $\mathbf{x}^\dagger$ with the largest distance to $\mathbf{x} = (\mathbf{x}^*, \mathbf{x}^\dagger)$ is the same hyperplane in $\mathcal{S}$ separating $\mathbf{x}^*$ with the largest distance to $\mathbf{x}^*$, that is $Q(\mathbf{x}^*, \mathcal{L}) = Q(q, \mathbf{x}^*, \mathcal{S})$. This implies $\hat{\beta}^* = \tilde{\beta}^*$ and hence the proposition is proved. ■

Now by Theorem 1 and above proposition we obtain

$$\hat{\beta}^*(s)/s - \tilde{\beta}^*(s)/s \to \hat{\beta}^* - \tilde{\beta}^* = 0.$$

This completes the proof of Theorem 2.

REFERENCES

[1] Y. Lin, G. Wahba, D. Xiang, F. Gao, and B. Klein, Smoothing spline ANOVA models for large data sets with Bernoulli observations and the randomized GACV, Ann Stat 28 (2000), 1570–1600.

[2] A. Smola and B. Scholkopf, Sparse greedy matrix approximation for machine learning, Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco, CA, Morgan Kaufmann, 2000, 911–918.

[3] J. Zhu and T. Hastie, Kernel logistic regression and the import vector machine, J Comput Graph Stat 14 (2005), 185–205.

[4] M. Nguyen and F. Torre, Optimal feature selection for support vector machines, Pattern Recognit 43 (2010), 584–591.

[5] J. Weston, A. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, Feature selection for SVMs, Adv Neural Inf Process Syst 13 (2000).

[6] S. Canu and Y. Grandvalet, Published in the conference Advances in Neural Information Processing Systems (NIPS 2002), December 9-14, 2002, Vancouver, British Columbia, Canada.

[7] G. Kimeldorf and G. Wahba, Some results on tchebysheffian spline functions, J Math Anal Appl 33 (1971), 82–95.

[8] G. Wahba, Spline Models for Observational Data, Philadelphia, PA, SIAM, 1990.

[9] V. Roth, The generalized LASSO, IEEE Trans Neural Netw 15(1) (2004), 16–28.

[10] E. Parzen, Statistical inference on time series by the RKHS methods, In Proceedings of the 12th Biennial Seminar, Canadain Mathematical Congress, Montreal, 1970, 1–37.

[11] N. Aronszajn, Theory of reproducing kernels, Trans Am Math Soc 68 (1950), 337–404.

[12] V. N. Vapnik, The Nature of Statistical Learning Theory, Berlin, Springer Verlag, 1995.

[13] T. Evgeniou, M. Pontil, and T. Poggio, A Unified Framework for Regularization Networks and Support Vector Machines, A.I. Memo 1654, Boston, MA, Institute of Technology, 1999.

[14] S. Rosset, J. Zhu, and T. Hastie, Margin maximizing loss functions, Neural Information Processing Systems (NIPS 2003) 16 Cambridge, MA, MIT Press, (2004).

[15] Y. Lin, Support vector machines and the Bayes rule in classification, Data Minning Knowel Discov 6 (2002), 259–275.

[16] V. N. Vapnik, Statistical Learning Theory, New York, Wiley, 1998.

[17] W. Weston and C. Watkins, Multi-class Support Vector Machines, Technical Report CSD-TR-98-04, University of London, Royal Holloway, 1999.

[18] Y. Lee, Y. Lin, and G. Wahba, Multicategory support vector machines, theory, and application to the classification and microarray data and satelite rafiance data, J Am Stat Assoc 99 (2004), 67–81.

[19] R. Tibshirani, Regression shrinkage and selection via the laso, J R Stat Soc B 58 (1996), 267–288.

[20] M. Park and T. Hastie, Penalized logistic regression for detecting gene interactions, Biostatistics 9 (2008), 30–50.

[21] V. Numerics, IMSL C Library Users's Manual, vetrsion 5.5, ed., Vol. 2, Houston, TX, IMSL, 1987.

[22] J. E. Dennis and R. B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Englewood Cliffs, NJ, Prentice Hall, 1983.

[23] M. West, C. Blanchette†, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, A. J. Olson, J. J. R. Marks, and J. R. Nevins, Predicting the clinical status of human breast cancer by using gene expression profiles, In Proceedings of the National Academy of Sciences 98(20), (2001), 11462–11467.

[24] 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, In Proceedings of the National Academy of Sciences, 96(12), (1999), 6745–6750.

[25] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, Series in Statsitics, New York, Springer, 2001.

[26] G. Getz, H. Gal, I. Kela, D. A. Notterman, and E. Domany, Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data, Bioinformatics 19 (2003), 1079–1089.

[27] N. Pochet, F. D. Smet, J. A. K. Suykens, and B. L. R. De Moor, Systematic benchmarking of microarray data classification: assessing the role of non-linearity and dimensionality reduction, Bioinformatics 20 (2004), 3185–3195.

[28] X. Li, S. Rao, Y. Wang, and B. Gong, Gene mining: a novel and powerful ensemble decision approach to hunting for disease genes using microarray expression profiling, Nucleic Acids Res 32 (2004), 2685–2694.

[29] S. Yang, T. Murali, and V. Pavlovic, M. Schaffer, and S. Kasif , RankGene: identification of diagnostic genes based on expression data, Bioinformatics 19 (2003), 1578–1579.

[30] S. Yang, Comparison of RankGene Measures for Colon Cancer Data of Alon et al., 2010. Available at http://genomics10.bu.edu/yangsu/rankgene/compare-alon-colon-cancer-top100.html (accessed on 18 September 2014).