Sparse Bayesian Approach for Feature Selection

Chang Li
UBRI, School of Computer Science and Technology
University of Science and Technology of China
Hefei, China 230027
Email: changli@mail.ustc.edu.cn

Huanhuan Chen
UBRI, School of Computer Science and Technology
University of Science and Technology of China
Hefei, China 230027
Email: hchen@ustc.edu.cn

***Abstract*—This paper employs sparse Bayesian approach to enable the Probabilistic Classification Vector Machine (PCVM) to select a relevant subset of features. Because of probabilistic outputs and the ability to automatically optimize the regularization items, the sparse Bayesian framework has shown great advantages in real-world applications. However, the Gaussian priors that introduce the same prior to different classes may lead to instability in the classifications. An improved Gaussian prior, whose sign is determined by the class label, is adopt in PCVM. In this paper, we present a joint classifier and feature learning algorithm: Feature Selection Probabilistic Classification Vector Machine (FPCVM). The improved Gaussian priors, named as truncated Gaussian prior, are introduced into the feature space for feature selection, and into the sample space to generate sparsity to the weight parameters, respectively. The expectation-maximization (EM) algorithm is employed to obtain a maximum a posteriori (MAP) estimation of these parameters. In experiments, both the accuracy of classification and performance of feature selection are evaluated on synthetic datasets, benchmark datasets and high-dimensional gene expression datasets.**

I. Introduction

In binary classification problems, given a dataset $D = \{x^{(i)}, y^{(i)}\}_{i=1}^{N}$ called training set, where $x^{(i)} \in \mathbb{R}^M$ is the feature vector with M dimensions and $y^{(i)} \in \{-1, +1\}$ is the corresponding class label. The goal is to learn a decision function $f(\cdot)$ with some controlling parameters to predict the label of new vector sharing the same distribution. Performance of the new classifier is estimated by the generalization ability, i.e., the accuracy of the prediction of the classifier.

Recently, the kernel based linear discriminant function $f(\cdot)$ has drawn great interest in research [3], [14], which treats prediction function $f(\mathbf{x}; \mathbf{w})$ as a linear combination of kernel functions:

$$f(\mathbf{x}; \mathbf{w}, b) = \sum_{i=1}^{N} w_i \phi_{i,\theta}(\mathbf{x}) + b = \mathbf{\Phi}_\theta(\mathbf{x})\mathbf{w} + b, \qquad (1)$$

where $\mathbf{w} = (w_1, w_2, \cdots, w_N)^T$ is the weight parameter of the model, b is the bias and $\mathbf{\Phi}_\theta(x) = (\phi_{1,\theta}(x), \phi_{2,\theta}, \cdots, \phi_{M,\theta})$ is the kernel function parameterized by θ called kernel parameters.

In terms of joint classifier and feature learning, the kernel parameters $\theta = (\theta_1, \theta_2, \cdots, \theta_M)$ can be extended to M dimension to accommodate feature selection. When an element θ_k equals to 0, the corresponding feature will not contribute for learning results. Therefore, the joint classifier and feature learning is achieved by seeking sparsity in the sample space, i.e. weight parameter $\mathbf{w}$ and in the feature space, i.e. the kernel parameters θ. In this paper, we focus on two widely used kernels:

Gaussian RBF kernel:

$$\phi_\theta(x, x^{(i)}) = \exp\left(-\sum_{k=1}^{M} \theta_k (x_k - x_k^{(i)})^2\right).$$

pth order polynomial kernel: (2)

$$\phi_\theta(x, x^{(i)}) = \left(1 + \sum_{k=1}^{M} \theta_k x_k x_k^{(i)}\right)^p$$

Among the kernel based methods, Relevance Vector Machine (RVM) [14] is one of the famous algorithms based on sparse Bayesian framework. By using Bayesian inference and the sparseness promoting priors, RVM can both output the predict label and the confidence probability. RVM is also a sparse model, and the introduced zero-mean Gaussian prior will lead most of weights to zero.

The success of RVM includes: it optimizes parameters by Bayesian inference, and the probabilistic outputs giving the uncertainty of prediction. This uncertainties allow to obtain full approximation of misclassification costs, which in return benefits for making decisions. But Chen *et al*. argue that the traditional Gaussian prior may bring instability to the results [3]. Based on this finding, Chen *et al*. proposed Probabilistic Classification Vector Machine (PCVM) [3] by using truncated Gaussian prior for different class labels to increase the stability of model.

Feature selection is an important learning task that can remove some redundant and misleading features. In order to address the issue, many filter based feature selection algorithms are employed , such as Max-Dependency Min-Redundancy (mRMR) [12], T-test [16] and Fishers Criterion [7]. But in these algorithms, feature selection is isolated from classifier learning, which might be hard to obtain the optimal dimension subset. Though cross-validation can be used to determine the size of subset, the procedure will take too much extra effort.

However, PCVM and RVM did not perform feature learning and they might suffer from redundance in the dimension space, especially for high-dimensional datasets. In this paper, we propose a joint classifier learning and feature learning algorithm, Feature Selection Probabilistic Classification Vector Machine (FPCVM). There are two tasks in FPCVM: learning a optimized classifier (classification) and selecting the most relevant feature subset with classification (feature selection). To fulfill these tasks, FPCVM employs truncated Gaussian

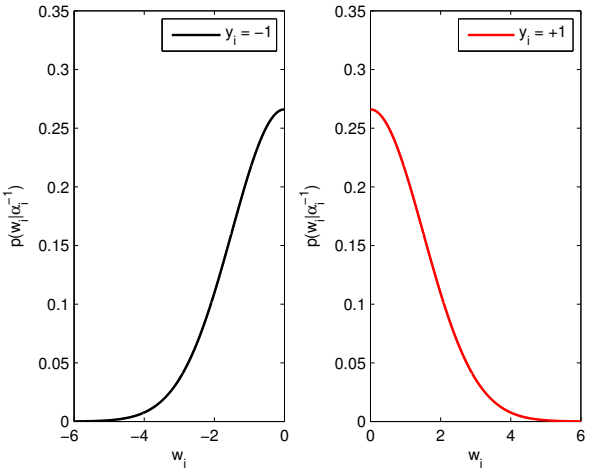

Fig. 1. The truncated Gaussian prior. Sign of each prior is corresponding to the class label, i.e., if $y = +1$ left truncated Gaussian prior is employed; if$y = -1$ right truncated Gaussian prior is employed

priors both in weight parameters and feature parameters, i.e. kernel parameters. These priors run as regularization terms and seek sparsity in both basis functions and feature space.

Compared with PCVM and EPCVM, the advantages of FPCVM are summarised as follow: 1) Being an embedded feature selection model, the approach simultaneously learns the classifier and selects the relevant features; 2) being a sparse model, FPCVM employs sparseness promoting priors in the feature space to seek sparseness.

The paper is organized as follows: Section II introduces the details of sparse Bayesain framework; Section III proposes FPCVM; the experiment study will be reported in Section IV; in Section V the main contribution and future work are summarized.

II. Sparse Bayesian Framework

This section will present sparse Bayesian framework used in classification problems and explain how priors promote sparsity.

A. Model Specification

Consider the binary classification problem, the input samples are $\{x^{(i)}, y^{(i)}\}_{i=1}^{N}$, where $x^{(i)}$ is the feature vector with M dimensions and $y^{(i)} \in \{-1, +1\}$. A link function is employed to smoothly map linear outputs to probabilistic outputs. In PCVM, the following probit link function is utilized:

$$\Psi(x) = \int_{-\infty}^{x} N(t|0,1)dt,$$

where $\Psi(x)$ is the zero-mean Gaussian cumulative function. After linked with probit function, the model becomes:

$$l(y = 1; \mathbf{w}, b) = \Psi(\mathbf{\Phi}_\theta(\mathbf{x})\mathbf{w} + b). \tag{3}$$

B. Truncated Gaussian Priors

In Section I, PCVM uses truncated Gaussian priors as weight priors and Gaussian prior as bias prior to optimize parameters.

$$\begin{aligned} p(\mathbf{w}|\alpha) &= \prod_{i=1}^{N} p(w_i|\alpha_i) = \prod_{i=1}^{N} N_t(w_i|0, \alpha_i^{-1}), \\ p(b|\beta) &= N(b|0, \beta^{-1}), \end{aligned}$$

where $N_t(w_i|0, \alpha_i^{-1})$ is the truncated zero-mean Gaussian prior, shown in Figure 1; α and β are hyperparameters of $\mathbf{w}$ and b, also called precision parameters, which are the inversion of standard derivation of corresponding distributions.

With these priors, PCVM obtains the optimized parameters by using a maximum a posteriori (MAP) estimation on each data point. As it is hard to achieve the complete dataset and model includes lots of latent variables, i.e., the hyperparameters α and β, an expectation-maximization (EM) [5] algorithm is used to efficiently calculate the MAP of training data.

The next section will present a joint classifier and feature learning algorithm and present the derivation of EM to calculate the MAP.

III. Feature Selection Probabilistic Classification Vector Machine

As noticed in Section I, there are two tasks of FPCVM: classifier training and feature selection. Classifier learning has been specified in the former section and for selecting relevant subset, priors must be chosen for feature parameters as well to encourage sparseness in the feature space. In this section, we will present the details of FPCVM and present the specific derivation of EM algorithm.

A. Prior Over Feature Parameters

In the feature space, one reasonable approach is to assign one feature parameter to each feature. Lager feature parameters mean more important features. If a feature parameter is zero, it means that the corresponding feature is removed from model. Based on this intuitive approach, there are no reasonable explanation for negative feature parameters. Therefore, this paper introduces left truncated Gaussian priors, illustrated in Figure 2, to feature parameters. For the kernels noticed in Equation (2), each dimension is attached with a left truncated Gaussian prior:

$$p(\theta_k|\gamma_k) = N_t(\theta_k|0, \gamma_k^{-1}) = \begin{cases} 2N(\theta_k|0, \gamma_k^{-1}) & if\ \theta_k \geq 0 \\ 0 & if\ \theta_k < 0 \end{cases}, \tag{4}$$

where γ are the hyperparemeters which control the precision of kernel parameters. When $\gamma_k \longrightarrow \inf$, it means the kth dimension has little relation with classification task and θ_k is restrict to a small neighborhood of zero, With Equation (3) and the priors of kernel parameters, the expectation-maximization (EM) algorithm will simultaneously calculate both the weight and feature parameters.

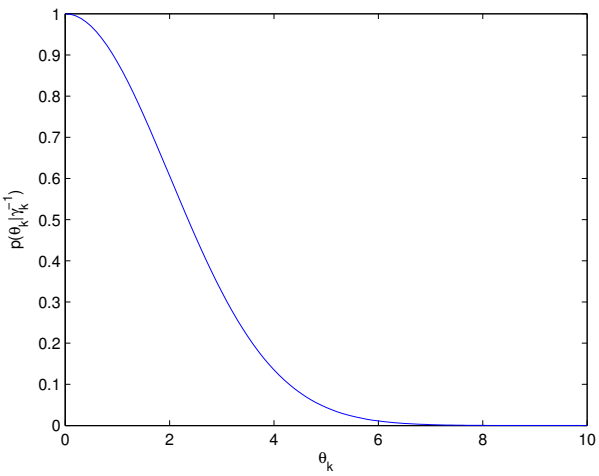

Fig. 2. The truncated Gaussian priors over feature parameters.

B. EM Algorithm

EM algorithm is employed to solve a MAP estimation. Following the standard probabilistic procedure, we assume that model outputs $\mathbf{\Phi}_\theta(\mathbf{x})\mathbf{w}+b$ are disturbed by a random zero-mean Gaussian noise ϵ, where $\epsilon \sim N(0,1)$. Based on the linked model, when $h_\theta(x) = \mathbf{\Phi}\mathbf{w} + b \geqslant 0$, $\hat{y} = 1$ and when $h_\theta(x) = \mathbf{\Phi}\mathbf{w} + b < 0$, $\hat{y} = -1$. The probit model becomes:

$$p(\hat{y} = 1|\mathbf{x}, \mathbf{w}, b, \theta) = \Psi(\mathbf{\Phi}_\theta(\mathbf{x})\mathbf{w}+b). \tag{5}$$

Consider the vector of hidden variables h_θ , α, β and γ. If h_θ , α, β and γ were observable, estimating θ and $\mathbf{w}$ would be simple:

- If h_θ was known, the likelihood of $\mathbf{w}$ would be directly estimated by our assumption.
- If α, β and γ were known, the priors of weight parameters and feature parameters would be obtained respectively by the truncated Gaussian distributions.

Noting the kernel matrix $\mathbf{\Phi}_\theta(\mathbf{x}) = (\mathbf{\Phi}_\theta(\mathbf{x}^{(1)}), \cdots, \mathbf{\Phi}_\theta(\mathbf{x}^{(N)}))$ and $\mathbf{H}_\theta(\mathbf{x}) = (h_\theta(\mathbf{x}^{(1)}), \cdots, h_\theta(\mathbf{x}^{(N)}))^T$, where $\mathbf{\Phi}_\theta(\mathbf{x^{(i)}}) = (\phi_\theta(x^{(1)}, x^{(i)}), \cdots, \phi_\theta(x^{(N)}, x^{(i)}))^T$, the probability of hidden variables are:

$$p(\mathbf{H}_\theta|\mathbf{w}, b) = (2\pi)^{-N/2} \exp\left\{-\frac{1}{2}\|\mathbf{H}_\theta - (\mathbf{\Phi}_\theta\mathbf{w}+bI)\|^2\right\},$$

where $I = (1, \cdots, 1)^T$ is the N-dimension identity matrix. Log-posterior of parameters is:

$$\begin{aligned}
&\log\, p(\mathbf{w}, b, \theta|y, H_\theta, \alpha, \beta, \gamma) \\
\propto &\log\, p(H_\theta|\mathbf{w}, b, \theta) + \log\, p(\mathbf{w}|\alpha) + \log\, p(b|\beta) + \log\, p(\theta|\gamma) \\
\propto &\mathbf{w}^T\Phi_\theta^T(2H_\theta - \Phi_\theta\mathbf{w}) + 2bI^T H_\theta - 2bI^T\Phi_\theta\mathbf{w} - b^2N \\
&- \mathbf{w}^T A\mathbf{w} - \beta b^2 - \theta^T\Gamma\theta,
\end{aligned} \tag{6}$$

where $A = diag(\alpha_1, ..., \alpha_N)$, $\Gamma = diag(\gamma_1, ..., \gamma_M)$.

Expectation-step: In E-step, the expectation of log-posterior, noted as Q, will be calculated:

$$\begin{aligned}
&Q(\mathbf{w}, b, \theta|\mathbf{w}^{old}, b^{old}, \theta^{old}) \\
&= E[\log p(\mathbf{w}, b, \theta|y, H_\theta, \alpha, \beta, \gamma)|y, \mathbf{w}^{old}, b^{old}, \theta^{old}] \\
&= \mathbf{w}^T\Phi_\theta^T(2\bar{H}_\theta - \Phi_\theta w) + 2bI^T\bar{H}_\theta - 2bI^T\Phi_\theta\mathbf{w} \\
&- b^2N - \mathbf{w}^T\bar{A}\mathbf{w} - \bar{\beta}b^2 - \theta^T\bar{\Gamma}\theta,
\end{aligned} \tag{7}$$

where: $\bar{H}_\theta = E[H_\theta|y, \mathbf{w}^{old}, b^{old}, \theta^{old}]$, $\bar{A} = diag(E[\alpha_i|y_i, \mathbf{w}^{old}, b^{old}, \theta^{old}])$, $\bar{\beta} = E[\beta|y_i, \mathbf{w}^{old}, b^{old}, \theta^{old}]$ and $\bar{\Gamma} = diag(E[\gamma_i|y_i, \mathbf{w}^{old}, b^{old}, \theta^{old}])$. All in all, all the variables in E-step can be evaluated analytically.

Maximization-step: In M-step, parameters are updated by:

$$(\mathbf{w}^{new}, b^{new}, \theta^{new}) = \arg\max_{\mathbf{w},b,\theta} Q(\mathbf{w}, b, \theta|\mathbf{w}^{old}, b^{old}, \theta^{old}).$$

The function Q is dominated by parameters $\mathbf{w}, b, \theta$ and the gradients with respect to $\mathbf{w}, b, \theta$ are analytical:

$$\frac{\partial Q}{\partial \mathbf{w}} = -2\Phi_\theta^T\Phi_\theta\mathbf{w} + \Phi_\theta^T\bar{H}_\theta - 2\bar{A}\mathbf{w}, \tag{8}$$

$$\frac{\partial Q}{\partial b} = 2I^T\bar{H}_\theta - 2bN - 2\Phi_\theta\mathbf{w} - 2b\bar{\beta}, \tag{9}$$

$$\frac{\partial Q}{\partial \theta_k} = -2\theta_k\bar{\gamma}_k - 2\sum_{i=1}^{N}\sum_{j=1}^{N}\{(\Phi_\theta\mathbf{w} - \bar{H}_\theta)\mathbf{w}^T \odot \frac{\partial\Phi_\theta}{\partial\theta_k}\}_{(i,j)}, \tag{10}$$

where $\odot$ represents for element-wise Hadamard matrix multiplication.

Generally, it is hardly to analytically acquire the joint maximization of Q respect to $\mathbf{w}, b, \theta$. However, by solving $\frac{\partial Q}{\partial \mathbf{w}} = 0$ and $\frac{\partial Q}{\partial b} = 0$, we get the optimal $\mathbf{w}$ and b:

$$\mathbf{w}^{new} = (\Phi_\theta^T\Phi_\theta + \bar{A})^{-1}(\Phi_\theta^T\bar{H}_\theta - b\Phi_\theta^T I), \tag{11}$$

$$b^{new} = \frac{I^T\bar{H}_\theta - I^T\Phi_\theta\mathbf{w}}{\bar{\beta} + N}. \tag{12}$$

Adding $\mathbf{w}^{new}$ and b^{new} to Q, the maximization with θ could be solved by convex optimization method, in this paper we use a standard conjugate gradient algorithm.

The procedure of solving FPCVM is summarized by the following pseudo code: Algorithm 1.

Algorithm 1 consists of the following major steps:

1) Initialize model parameters (line 1-2).
2) Calculate kernel matrix ϕ (line 4).
3) Update model parameters : $\mathbf{w}$ based on Equation (11) and b based on Equation (12) (line 5).
4) Update feature parameters : θ based on Equation (10) (line 6).
5) Update index vector: used (line 7).
6) Compare the max change of $\mathbf{w}$ with threshold to determine whether algorithm is converged, if so terminate algorithm and output the results, else go back to line 3 and continue the loop (line 9-10).

In case of ill matrix decomposition, we employ an index vector **used** to identify the sample whose weight w is closed to zero and directly remove the corresponding basis function

Algorithm 1: FPCVM

Input: $\{\mathbf{X}, \mathbf{Y}\} = \{x_n, y_n\}_{n=1}^{N}$ is training set; *maxItes* is maximal iteration; *threshold* is a const number to determine whether to converge; *initParameter* are initial parameters of model; and *used* is a vector to index which sample is in use.

Output: weight vectors w, bias b and feature parameters θ

$[\mathbf{w}, \mathbf{b}, \theta]$=*initialize*(initParameter);
vector=*determine_usefull_vector*(w);
for $i = 1$ *to* $maxItes$ **do**
  $\mathbf{\Phi} = Kernel(\mathbf{X}, \mathbf{Y}, \theta)$;
  $[\mathbf{w^{new}}, b^{new}] = weight\_vectors\_update\{\mathbf{\Phi}, \mathbf{w}, \mathbf{b}, \mathbf{Y}, \mathbf{used}\}$;
  $\theta^{\mathbf{new}} = feature\_parameter\_update(\mathbf{X}, \mathbf{Y}, \theta, \mathbf{w^{new}}, \mathbf{b^{new}})$;
  $\mathbf{used^{new}} = update\_usefull\_vector(\mathbf{w^{new}})$;
  if $max(abs(\mathbf{w^{new}} - \mathbf{w})) < threshold$ **then**
    break;
  end
end

from kernel matrix $\mathbf{\Phi}$. Theoretically, $(\mathbf{\Phi^T \Phi} + \bar{\mathbf{A}})$ is positive definite, however, when some diagonal values of $\bar{A}$ tend to be infinity, it may still hard to decomposed analytically, explained by Tipping [14]. To avoid this, we also need to prune the corresponding hyperparamets and the basis functions. The same procedure has been used by Figueiredo in [6]. Furthermore, to increase the stability of calculation inversion, Cholesky decomposition is used instead of directly processing inversion of matrix [13].

In FPCVM, the bottlenecks of computation are the inverse of matrix in Equation (11) and the update of feature parameters θ. Cholesky decomposition is used in calculating the inversion, which has the memory complexity $O(\tilde{N}^2)$ and computational complexity $O(\tilde{N}^3)$, where $\tilde{N} \leq N$ is the number of used basis functions. The rule of updating θ involves recalculating kernel matrix which has the computational complexity $O(\tilde{N}^2\tilde{M})$ and memory complexity $O(\tilde{N}^2)$. Where $\tilde{M}$ is the number of used features and $\tilde{M} \leq M$.

This complexity requires larger memory usage and longer training time. However, because of the sparseness promoting priors, FPCVM can rapidly prune basis functions and features to a small subsets. To obtain a faster approach the online strategies could be employed [4], [15].

IV. Experimental Study

In this section, FPCVM is estimated on several datasets including synthetic data, high-dimensional gene expression data and real-world benchmark datasets. To evaluate the feature selection performance of FPCVM, FPCVM is evaluated on one synthetic dataset. The experiments on gene expression datasets whose dimensions are mostly irrelevant are to evaluate the performance of feature selection. The ability of classification is evaluated on several benchmark datasets. To obtain reasonable comparisons with other state-of-the-art methods, the cross-validation has been used in both FPCVM and other compared algorithms.

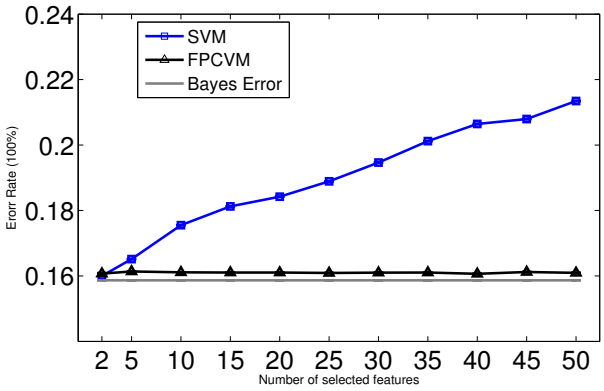

Fig. 3. A set of test on synthetic data. With the increasing irrelevant features, the error rates of SVM increases while FPCVM shows great immunity from irrelevant features.

A. Synthetic Data

The synthetic dataset is generated by two two-dimensional Gaussian distributions. Each Gaussian distribution standing for a class has the unit standard variance and opposed-number means, i.e., $-\mu_1 = \mu_2 = [1/\sqrt{2}, 1/\sqrt{2}]^T$. The dimension is increased by introducing irrelevant features created from a random process ranging from $[-1, +1]$. For the K-dimension dataset, the feature becomes:

$$-\mu_1 = \mu_2 = [1/\sqrt{2}, 1/\sqrt{2}, \underbrace{random_1, \cdots, random_{K-2}}_{(K-2) random\ values}]^T.$$

Intuitively, only the two original features are useful and the optimal decision boundary is linear with Bayes error rate: $\Phi(-1) \simeq 0.1587$. The training set composes of 200 samples generated from each distribution and the independent testing set includes 1000 samples from each distribution. We use the lib-SVM[1] as the compared algorithm.

The tests are repeated twenty times to get more accurate results. Both FPCVM and SVM use linear kernel, and the result is illustrated in Figure 3. With the increasing dimensions, SVM underperforms while FPCVM remains to be accurate.

This experiment demonstrates that FPCVM is a more robust algorithm dealing with many irrelevant features. As FPCVM automatically estimates the feature parameters, F-PCVM always selects the two original features. The degraded performance of SVM with many irrelevant features indicates that the feature selection is crucial in designing classifiers.

B. Benchmark Data

In this set of experiments, we use four real-world datasets to evaluate the performance of FPCVM. These datasets can be obtained from the UCI [11] and DELVE[2] and some

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[2] http://www.cs.toronto.edu/%7edelve/data/datasets.html

TABLE I. COMPARISON OF COMPARED ALGORITHMS ON GENE EXPRESSION DATA (% ACCURATE)

Classifier	AML/ALL	Colon
Adaboosting(Ecision stumps) [2]	95.8	72.6
SVM (Linear kernel) [2]	94.4	77.4
SVM (Quadratic kernel) [2]	95.8	74.2
Logistic regression(No kernel: on feature space) [9]	97.2	71.0
RVM(No kernel: on feature space) [9]	97.2	88.7
Sparse probit regression(No kernel: on feature space) [10]	97.2	88.7
Sparse probit (Linear kernel) [10]	95.8	84.6
JFCO (Quadratic kernel) [10]	98.6	88.7
FPCVM (RBF kernel)	**100**	**95.2**

TABLE II. SUMMARY OF FOUR BENCHMARK DATASETS.

Data	No. Train	No. Test	Positive %	Negative %	Dim
Splice	1000	2175	44.93%	55.07%	60
Thyroid	140	75	30.23%	69.77%	5
Titanic	150	2051	58.33%	41.67%	3
Waveform	400	4600	32.94%	67.06%	21

TABLE III. COMPARISON OF CLASSIFICATION ALGORITHMS ON FOUR BENCHMARK DATASETS, BY % ERROR AND (STANDARD DERIVATION) .

Error	Splice	Thyroid	Titanic	Waveform
*k*NN [3]	24.00(2.35)	4.36(2.21)	23.84(4.94)	10.98(0.67)
QDA [3]	14.77(0.94)	6.85(2.37)	24.30(6.43)	16.65(0.83)
SVM_{soft} [3]	10.70(0.63)	4.79(2.04)	22.69(0.86)	**10.25(0.43)**
SVM_{hard} [3]	10.70(0.63)	5.04(2.14)	**22.30(1.05)**	10.95(0.57)
RVM [3]	12.94(0.71)	5.12(2.62)	23.30(1.50)	10.80(0.64)
PCVM [3]	10.60(0.65)	4.55(2.49)	22.58(1.37)	10.40(0.58)
FPCVM	**7.12(0.68)**	**3.87(2.12)**	22.95(1.37)	11.15(0.51)

have been organized and preprocessed by Rätsch *et al.*[3]. The characteristics of datasets is summarized in Table II.

The performance of these algorithms are reported in Table III and most of experimental results come from [3]. For RVM, the kernel parameter is selected by cross-validation. The similar cross-validation procedure is used to choose the trade-off parameter c in SVM.

In these datasets, FPCVM chooses 26 out of 60 features in splice dataset, 4 out of 5 features in thyroid dataset, and 18 out of 21 features in waveform dataset. In titanic dataset, which has only 3 features, FPCVM only optimizes feature parameters, instead of performing feature selection, i.e. setting the corresponding feature parameters to zero.

In Table III, FPCVM is the best in splice and titanic datasets and very close to the best in the other two datasets. Other state-of-the-art algorithms may have a little advantage in accuracy, but FPCVM performs the joint classifier and feature learning and successfully optimized both weight parameters and feature parameters.

C. High-Dimension Gene Expression Data

Generally speaking, gene expression data consists of many irrelevant and redundant dimensions, which means feature selection is important to obtain robust classifiers [10]. The experiments are performed on two gene expression datasets: leukemia dataset[4] and colon cancer dataset[5]. Leukemia data is organized by Golub *et al.* [8] and consists of 7129 genes (features) getting from 47 acute myeloid leukemia (AML) patients and 25 acute lymphoblastic leukemia (ALL) patients. Colon cancer dataset contains 2000 genes (features) with 22 normal and 40 tumor colon tissues [1]. We will use these two datasets to assess the performance of FPCVM against the high dimensionality. During this experiment, a Gaussian RBF kernel is utilized.

To estimate the performance, the experiment employs a leave-one-out cross-validation procedure shown in [10]. This testing strategy suggests that during training stage, we use $\mathbf{n}-\mathbf{1}$ samples to build the model then test on the left sample and repeat the procedure one-by-one for each sample. The result is shown in Table I and compared with SVM, RVM, Sparse Probit Regression (SPR) [6], Joint Classifier and Feature Optimization (JCFO) [10] and the logistic regression. FPCVM is better than other algorithms on the both datasets.

In the colon cancer experiments, FPCVM averagely selects 53.6 out of 2000 genes to make diagnose during each repeat and in all repeated experiments 584 genes are chosen at least one time. The importance of each kernel parameter during all test is shown in Figure 4. Intuitively, there are 8 genes with outstandingly large coefficients and their gene accessions and names are shown in Table IV. For more in-depth analysis, 200 genes with relative smaller coefficients are only selected once in the leave-one-out cross-validation. These genes may be irrelevant and contribute little in the diagnosis. While only 13 genes, including the former mentioned 8 genes, with relative larger coefficients are chosen during over half of the leave-one-out cross-validation. The properties of these genes are shown in Table IV and most of them are translated in colon cancer, i.e., useful in diagnosing [1].

In the AML/ALL (leukemia) experiments, the importance of θ is shown in the bottom picture of Figure 4. Briefly, there are 218 out of 7129 useful genes shown in the Figure 4 and 8 genes play more important roles (the eight highest bars in the picture) for diagnosing. In the 72 leave-one-out cross-validations, FPCVM totally selects 3819 genes to make predictions. However, most of these genes are used less than 10 times while 56 genes are selected more than 60 times out of 72 cross-validations.

V. CONCLUSION

A. Discussion

We proposed a joint classifier and feature learning algorithm FPCVM. The performance of FPCVM has been evaluated on two criteria: performance of feature selection and accuracy of classification. Different from other feature selection methods, FPCVM adopts sparse Bayesian approach and employs sparseness promoting priors in both weight and

[3]http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm

[4]http://www.broadinstitute.org/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=43

[5]http://microarray.princeton.edu/oncology/affydata/index.html

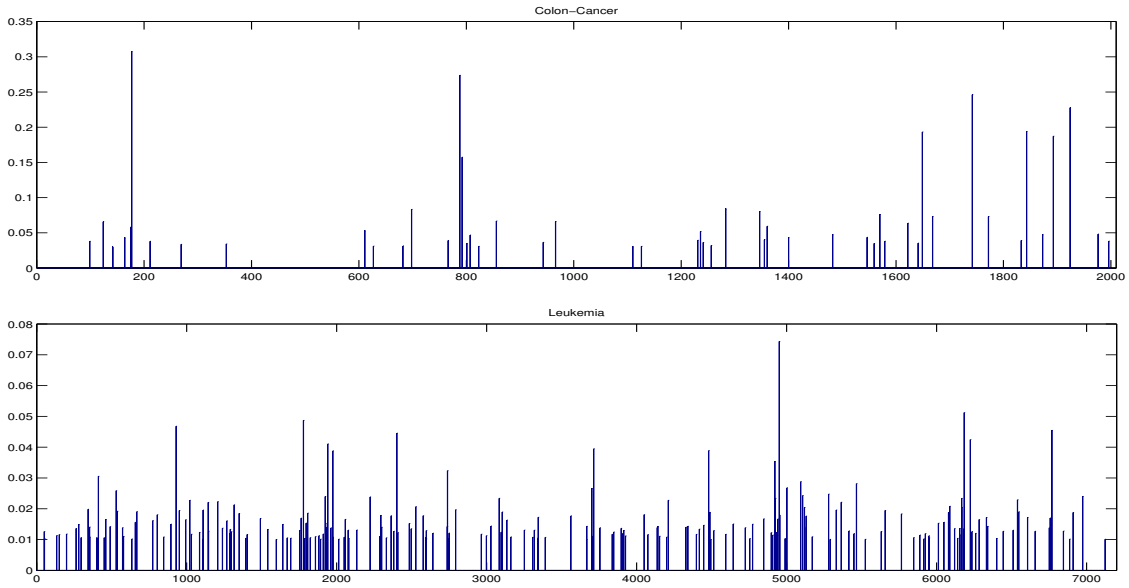

Fig. 4. Illsurlations of feature parameters (θ) for each feature for colon cancer dataset (top) and leukemia dataset (bottom).

TABLE IV. SOME IMPORTANT GENES FOR DIAGNOSIS COLON CANCER. THE DATA OF GENES COME FROM [1].

Coefficient(θ_k)	Index	Accession	Gene name
0.3073	177	T49941	IGF2
0.2737	788	X02492	IFI6
0.2461	1742	X02874	OAS1
0.2277	1924	H64807	SLC19A3
0.1937	1843	H06524	GSN
0.1932	1649	M31994	ALDH1A1
0.1873	1893	L34657	PECAM1
0.1574	792	R88740	ATP5J
0.0833	698	T51261	SERPINE2
0.0802	1346	T62947	RSL24D1
0.0762	1570	T54303	KRT8
0.0733	1668	M82919	GABRB3
0.0669	856	X66924	ID3

feature parameters to seek sparsity in both basis functions and feature space. The experiments demonstrate that the accuracy of FPCVM in benchmark datasets are either the best or very close to the best. On high-dimensional gene expression datasets, FPCVM performs remarkable accuracy compared with other state-of-the-art approaches. Moreover, FPCVM automatically outputs the relevant feature subset.

B. Future Work

Due to the disadvantages of EM algorithm, FPCVM might converge to local minima. Other approaches can be employed to solve FPCVM, e.g. Laplace approximation or expectation propagation [4]. The incremental basis updating strategy could lead to more efficient solutions of FPCVM [15].

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grants 61203292, 61311130140 and the One Thousand Young Talents Program.

REFERENCES

[1] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences*, vol. 96, no. 12, pp. 6745–6750, 1999.

[2] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, "Tissue classification with gene expression profiles," *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 559–583, 2000.

[3] H. Chen, P. Tino, and X. Yao, "Probabilistic classification vector machines," *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 901–914, 2009.

[4] H. Chen, P. Tino, and X. Yao, "Efficient probabilistic classification vector machine with incremental basis function selection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 356 – 369, 2014.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[6] M. A. Figueiredo, "Adaptive sparseness for supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1150–1159, 2003.

[7] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.

[8] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri *et al.*, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[9] B. Krishnapuram, A. Hartemink, and L. Carin, "Logistic regression and rvm for cancer diagnosis from gene expression signatures," in *Proc. 2002 Workshop Genomic Signal Processing and Statistics (GENSIPS)*, 2002.

[10] B. Krishnapuram, A. Harternink, L. Carin, and M. A. Figueiredo, "A bayesian approach to joint feature selection and classifier design," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1105–1111, 2004.

[11] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz, "UCI repository of machine learning databases," 1998. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[12] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[13] W. H. Press, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[14] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[15] M. E. Tipping, A. C. Faul *et al.*, "Fast marginal likelihood maximisation for sparse bayesian models," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, vol. 1, no. 3, 2003.

[16] N. Zhou and L. Wang, "A modified t-test feature selection method and its application on the hapmap genotype data," *Genomics, Proteomics & Bioinformatics*, vol. 5, no. 3, pp. 242–249, 2007.