The Relevance Sample-Feature Machine: A Sparse Bayesian Learning Approach to Joint Feature-Sample Selection

Yalda Mohsenzadeh, Hamid Sheikhzadeh, *Senior Member, IEEE,* Ali M. Reza, *Senior Member, IEEE,* Najmehsadat Bathaee, and Mahdi M. Kalayeh

***Abstract*—This paper introduces a novel sparse Bayesian machine-learning algorithm for embedded feature selection in classification tasks. Our proposed algorithm, called the relevance sample feature machine (RSFM), is able to simultaneously choose the relevance samples and also the relevance features for regression or classification problems. We propose a separable model in feature and sample domains. Adopting a Bayesian approach and using Gaussian priors, the learned model by RSFM is sparse in both sample and feature domains. The proposed algorithm is an extension of the standard RVM algorithm, which only opts for sparsity in the sample domain. Experimental comparisons on synthetic as well as benchmark data sets show that RSFM is successful in both feature selection (eliminating the irrelevant features) and accurate classification. The main advantages of our proposed algorithm are: less system complexity, better generalization and avoiding overfitting, and less computational cost during the testing stage.**

***Index Terms*—Joint feature selection and classifier design, relevance features, relevance sample feature machine (RSFM), relevance samples, sparse Bayesian learning.**

I. Introduction

In RECENT years, sparse kernel-based learning methods have been widely used for supervised learning applied to the regression or the classification problems. In supervised learning, given a training set of input and output pairs $\{\mathbf{x}_n, t_n\}_{n=1}^N$, in which $\mathbf{x}_n \in R^M$ and $t_n \in R$ for the case of regression or $t_n \in \{0, 1\}$ for the case of binary classification, the objective is to predict the output t^* for an arbitrary new test input $\mathbf{x}^*$. The kernel-based methods predict the output based on a function $y(\mathbf{x})$ that is a linearly weighted combination of some kernel functions centered on each training point, i.e.:

$$y(\mathbf{x}|\mathbf{w}) = w_0 + \sum_{n=1}^{N} w_n K(\mathbf{x}, \mathbf{x}_n) \tag{1}$$

in which $\mathbf{w} = (w_0, w_1, \ldots, w_N)^T$ is the weight vector of the linear model and $K(\mathbf{x}, \mathbf{x}_n)$ is a generally nonlinear and predetermined kernel function. The number of parameters (weights) and the number of training vectors in this model are the same. However, the sparse kernel-based learning methods avoid overfitting by imposing some additional constraints on the parameters. Therefore, the learned parameters result in a sparse model which is dependent only on a subset of kernel functions and their corresponding training vectors.

The support vector machine (SVM) [1] and the relevance vector machine (RVM) [2] are two principal cases of the sparse kernel-based learning methods. The SVM results in a sparse model by solving an optimization problem that attempts to minimize a measure of error on the training set while simultaneously maximizing the margin between the two classes. The selected training vectors in the SVM paradigm are called support vectors. The RVM results in a sparse model by assuming a zero-mean Gaussian prior with a different variance for each weight and a flat Gamma prior on the hyperparameters (variance of the weights) that results into an implicit (marginalized) prior over the parameters with a Student-t distribution. The selected training vectors in the RVM paradigm are called relevance vectors. As mentioned in [2], the RVM offers some advantages over the SVM. Since the RVM offers a Bayesian framework, the prediction of the RVM for regression or classification is probabilistic. Moreover, the RVM results in a sparser model compared to the SVM, and facilitates utilizing arbitrary kernel functions.

In the RVM-based regression, the output is modeled with $t_n = y(\mathbf{x}_n) + \epsilon_n$, in which ϵ_n is a zero-mean Gaussian noise with precision (inverse of variance) β. Therefore, the probability density function (PDF) of the output corresponding to a test input $\mathbf{x}$ given $\mathbf{w}$ and β will be normal: $p(t|\mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}|\mathbf{w}), \beta)$. In the RVM-based classifier, the probability that a test input $\mathbf{x}$ belongs to a specific class (e.g., $t = 1$) given $\mathbf{w}$ will be modeled with $p(t = 1|\mathbf{w}) = \sigma(y(\mathbf{x}|\mathbf{w}))$, in which $\sigma(y)$ is a sigmoid logistic function. The RVM assumes zero-mean

Manuscript received March 26, 2012; revised January 9, 2013; accepted April 19, 2013. Date of publication June 13, 2013; date of current version November 18, 2013. This paper was recommended by Associate Editor P. S. Sastry

Y. Mohsenzadeh, H. Sheikhzadeh, and N. Bathaee are with the Department of Electrical Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran 15914, Iran (e-mail: y.mohsenzadeh@aut.ac.ir; hsheikh@aut.ac.ir; nbathaee@aut.ac.ir).

A. M. Reza is with the Department of Electrical Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran 15914, Iran and also with the Department of Electrical Engineering and Computer Science, University of Wisconsin-Milwaukee, Milwaukee, WI 53201 USA (e-mail: moghaddamjoo@aut.ac.ir).

M. M. Kalayeh is with the Center for Research in Computer Vision (CRCV), University of Central Florida (e-mail: mahdi@eecs.ucf.edu).

Digital Object Identifier 10.1109/TCYB.2013.2260736

independent Gaussian priors for the weights, and introduces hyperparameters $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_N)^T$ as precisions of priors for the weights. In order to obtain the final sparse model, the RVM estimates the hyperparameters $\boldsymbol{\alpha}$ (and β in the regression case) by maximizing the marginal likelihood $p(\{t_n\}_{n=1}^N|\boldsymbol{\alpha})$ (or $p(\{t_n\}_{n=1}^N|\boldsymbol{\alpha}, \beta)$ in the regression case). During the learning procedure many of α_is tend to infinity, and thus the corresponding weights highly packed around zero. The corresponding kernel functions of these weights can then be pruned from (1) and sparsity is realized. Success of RVM in regression and classification has been proven in many applications; for example, in hand motion recognition [3], pose estimation and tracking [4], scaling text classification[5], 3-D human pose recovery from silhouettes [6], detection of clustered microclassifications for mammography [7], classification of gene expression data [8], [9], detection of activations for neuroimaging [10], real-time tracking [11], object detection in scenes [12], and mimicking human visual understanding in the cardiac defect detection task [13], [14], [15].

The classic RVM is a top-down approach. In other words, it begins with the full model including all of the training samples and prunes the irrelevant samples in each iteration. Fast RVM introduced in [16] is an incremental version of the RVM that starts with an empty model and in each iteration based on a criterion adds the relevant and informative samples or omits the irrelevant ones to finally achieve the learned model. In the fast RVM, there is a concern of convergence to a suboptimal model. Therefore, the available computational resources and the size of the training data set determine which version (classic RVM or fast RVM) to be used. Multiclass RVM presented in [17] is another extension to the classic RVM (which is a binary classifier) and addresses the multiclass recognition problems. In [17], two multiclass classification algorithms $mRVM_1$ and $mRVM_2$ were introduced. While the $mRVM_1$ is an incremental algorithm based on the fast RVM, the $mRVM_2$ employs the top-down approach.

In supervised learning, sparsity is an important and desirable property because of two reasons. First, sparsity controls the complexity of the model and avoids overfitting. Second, predicting with sparse model is computationally highly effective. The RVM finds a sparse subset of training samples, relevance samples, and employs a linear combination of kernels centered at these relevance samples for the regression or classification task. Each input sample in supervised learning consists of several features. However, the RVM is neither designed nor capable of identifying a subset of features (as sparse as possible) that is most informative for the regression or classification task. In this paper, we present the relevance sample feature machine (RSFM) that simultaneously finds the relevance samples and the relevance features. In other words, RSFM performs joint feature selection and classifier design simultaneously. Therefore, our proposed method, RSFM, benefits from sparsity in two domains, the samples and the features.

The idea of simultaneously finding the sparse set of samples and the sparse set of features is inspired by a theory in psychology of the human mind in recognizing visual objects. This theory states that in the process of learning a new visual object, the human mind does not memorize the whole object but only recalls some important discriminative and informative features of that object [18], [19], [20]. Rephrasing this theory for the binary classification task, we can say that the human mind keeps some informative features or perhaps views of some samples of two visual objects to perform classification. This simple theory serves as an inspiration for updating the RVM to simultaneously select the relevance samples and the relevance features (informative views) in the classification task.

Simultaneous selection of relevance features and classifier design during training process leads to what is known as embedded methods in feature selection [21]. Very few embedded feature selection methods for kernel-based classifiers exist in the literature [22], [23], [24], [25]. In [22], a joint classifier and feature optimization (JCFO) technique is introduced in which feature selection is performed by defining a kernel function with a separate scaling factor for each feature and applying a sparsity prior to the corresponding scaling factor. The JCFO model is

$$p(y_* = 1|\mathbf{x}_*) = \phi\left(w_0 + \sum_{i=1}^{N} w_i K_\sigma(\mathbf{x}_*, \mathbf{x}_i)\right) \tag{2}$$

where $\phi(.)$ is the Gaussian cumulative distribution function and $K_\sigma(\mathbf{x}, \mathbf{x}_i)$ is a kernel function with scaling parameter σ. Two examples for these kernels are *rth order polynomial*:

$$K_\sigma(\mathbf{x}_*, \mathbf{x}_i) = \left(1 + \sum_{k=1}^{L} \sigma_k x_{*k} x_{ik}\right)^r \tag{3}$$

and *Gaussian radial basis function (RBF)*:

$$K_\sigma(\mathbf{x}_*, \mathbf{x}_i) = exp\left(-\sum_{k=1}^{L} \sigma_k (x_{*k} - x_{ik})^2\right). \tag{4}$$

Recently, a joint feature selection and classifier learning (JFSCL) is introduced in [23]. The JFSCL is also an embedded method for feature selection and classification that performs this task by optimizing a global loss function that includes a term associated with the empirical loss and another term corresponding to feature selection and regularization constraint on the parameters. This loss function is an extended loss function for RVM

$$\Gamma(\Omega) = L(D, \mathbf{w}, \mathbf{v}) + \lambda(\|\mathbf{x}\|_1 + \|\sigma_k^d(\mathbf{v})\|_1) \tag{5}$$

where $\sigma_k^d(\mathbf{v})$ controls activation of features. The JFSCL uses Boosted Lasso (BLasso) algorithm [26] for optimization of this loss function.

In our proposed RSFM learning method, each input $\mathbf{X}_i$ in the training set is considered as a set of several features $\mathbf{X}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{iL}\}$. The prediction is based on a function $y(\mathbf{X}_i)$ that is a linear combination of some kernel functions centered at features of training point.

$$y(\mathbf{X}_*|\mathbf{w}, \boldsymbol{\lambda}) = \mathbf{w}^T \begin{pmatrix} K(\mathbf{x}_{*1}, \mathbf{x}_{11}) & \cdots & K(\mathbf{x}_{*L}, \mathbf{x}_{1L}) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_{*1}, \mathbf{x}_{N1}) & \cdots & K(\mathbf{x}_{*L}, \mathbf{x}_{NL}) \end{pmatrix} \boldsymbol{\lambda} \tag{6}$$

where $\mathbf{w}$ and $\boldsymbol{\lambda}$ are the weights of the linear model. The weights for samples are incorporated in $\mathbf{w}$, and the weights

for features are included in $\boldsymbol{\lambda}$. We place independent zero-mean Gaussian priors on both $\mathbf{w}$ and $\boldsymbol{\lambda}$ to promote sparsity and also derive the inference of the newly defined Bayesian model.

As illustrated in (2)–(4), JCFO achieves feature selection through adjusting the kernel parameter σ, while JFSCL achieves this by adding a term to the RVM cost function as shown in (5). In contrast, our proposed method introduces a new separable model illustrated in (6) with two sets of parameters for feature selection ($\boldsymbol{\lambda}$) and classifier design ($\mathbf{w}$). RSFM is more capable in comparison with the JCFO since RSFM has a kernel scaling parameter (σ for Gaussian kernels) which is not used for feature selection and so can be employed for improving the performance. Furthermore, RSFM is capable of employing multikernel methods that have already shown their usefulness in improving the classifier performance, for example, in SVM [27], [28] or RVM [29], [12]. In addition, JCFO uses a conjugate gradient algorithm for optimization of the scaling vector $\boldsymbol{\sigma}$ that results in a high computational cost and possibly convergence problems. The model and approach of the JFSCL is completely different from our proposed method in the sense that it just adds a new term to the cost function of RVM and optimize it using BLasso. In contrast, we propose a new separable kernel-based model and also present its Bayesian inference. As reported by the authors in [22] and [23], both JCFO and JFSCL suffers from computational complexity, which makes them impractical for large training data sets, while the computational complexity of RSFM, which uses EM-based optimization (expectation maximization), as evaluated by our extensive experiments is almost the same as RVM.

In [24], a feature selection method for SVM is presented. This method uses gradient descent algorithm for minimization of error bounds in the SVM. Another embedded feature selection approach is presented in [25] for the SVM (weighted SVM) using a convex energy-based framework for joint feature selection and classification. The weighted SVM exhibits promising results both for feature selection and classification performance. In addition to JCFO and JFSCL, RSFM is also compared with these two extensions on the SVM. Our performance evaluations demonstrate that the RSFM performs very closed to the mentioned works on the SVM but it has the advantages of probabilistic prediction and sparser trained model both in feature and sample domains.

The main advantages of the proposed RSFM method in comparison to the RVM can be summarized as: 1) simultaneous determination of relevance features and relevance samples (joint feature selection and classifier learning and sparsity in two domains), 2) lower system complexity, 3) better generalization and thus avoidance of overfitting, and 4) lower computational cost during normal operation of the algorithm (testing stage).

The rest of this paper is organized as follows. In Section II, we present our proposed model, the RSFM, and our inference algorithm for both the regression and classification tasks. Section III includes evaluation results and comparison of the RSFM with other state-of-the-art methods on two synthetic data sets, three small benchmark data sets, a large benchmark data set for handwritten digit recognition, two high dimensional gene expression data sets and also our generated data set for object recognition. Finally, our conclusion is presented in Section IV.

II. Relevance Sample Feature Machine

In this section, we will detail the inference procedure of the proposed RSFM for regression and classification tasks.

A. Model Specifications

Let $\{\mathbf{X}_n, t_n\}_{n=1}^{N}$ denote the set of input-output pairs, where $\mathbf{X}_n$ is a set of vectors representing the n^{th} input sample and t_n is the corresponding output value of the n^{th} sample. We assume that $\mathbf{X}_n$ includes L features $\mathbf{X}_n = \{\mathbf{x}_{n1}, \mathbf{x}_{n2}, \ldots, \mathbf{x}_{nL}\}$. Note that each $\{\mathbf{x}_{nj}\}_{j=1}^{L}$ is a $M \times 1$ vector. We incorporate the RVM model to realize our goal of determining the most relevance samples and the most relevance features. To that end, we consider target values as the samples of the model

$$t_n = y(\mathbf{X}_n; \mathbf{w}, \boldsymbol{\lambda}) + \epsilon_n \tag{7}$$

where ϵ_n's are i.i.d. samples of a zero-mean Gaussian noise with variance $\sigma^2 = \beta^{-1}$. Vectors $\mathbf{w} = [w_1, w_2, \ldots, w_N]^T$ and $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_L]^T$ are model adjustable parameters. Our aim is to find the optimum values of these parameters. Therefore, a mixture kernel-based model is proposed as

$$y_n = y(\mathbf{X}_n; \mathbf{w}, \boldsymbol{\lambda}) = \sum_{i=1}^{N} \sum_{j=1}^{L} w_i k(\mathbf{x}_{nj}, \mathbf{x}_{ij}) \lambda_j = \mathbf{w}^T \boldsymbol{\Phi}_n \boldsymbol{\lambda} \tag{8}$$

where $k(\mathbf{x}_{nj}, \mathbf{x}_{ij})$ is a kernel function applying over the j^{th} feature of the n^{th} and i^{th} input samples of the data set and $\boldsymbol{\Phi}_n$ is a $N \times L$ matrix defined as

$$\boldsymbol{\Phi}_n = \begin{array}{c} \\ sample1 \\ sample2 \\ \vdots \\ sampleN \end{array} \begin{array}{c} \begin{array}{cccc} feature1 & feature2 & \cdots & featureL \end{array} \\ \begin{pmatrix} k(\mathbf{x}_{n1},\mathbf{x}_{11}) & k(\mathbf{x}_{n2},\mathbf{x}_{12}) & \cdots & k(\mathbf{x}_{nL},\mathbf{x}_{1L}) \\ k(\mathbf{x}_{n1},\mathbf{x}_{21}) & k(\mathbf{x}_{n2},\mathbf{x}_{22}) & \cdots & k(\mathbf{x}_{nL},\mathbf{x}_{2L}) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_{n1},\mathbf{x}_{N1}) & k(\mathbf{x}_{n2},\mathbf{x}_{N2}) & \cdots & k(\mathbf{x}_{nL},\mathbf{x}_{NL}) \end{pmatrix} \end{array}. \tag{9}$$

Equation (9) shows a matrix where each row is corresponding to an input sample of the data set and each column is associated with one of the L features extracted from that particular sample. Substituting (9) in (8) would clearly illustrate the role of the adjustable vector parameters of $\mathbf{w}$ and $\boldsymbol{\lambda}$. This formulation is shown in the following:

$$y_n = [w_1, w_2, \ldots, w_N] \begin{bmatrix} k(\mathbf{x}_{n1},\mathbf{x}_{11}) & \cdots & k(\mathbf{x}_{nL},\mathbf{x}_{1L}) \\ k(\mathbf{x}_{n1},\mathbf{x}_{21}) & \cdots & k(\mathbf{x}_{nL},\mathbf{x}_{2L}) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_{n1},\mathbf{x}_{N1}) & \cdots & k(\mathbf{x}_{nL},\mathbf{x}_{NL}) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_L \end{bmatrix}. \tag{10}$$

As (10) shows, the proposed model is designed to be separable with respect to $\mathbf{w}$ and $\boldsymbol{\lambda}$. $\mathbf{w}$ determines the relevance samples in the data set, while $\boldsymbol{\lambda}$ determines the relevance features of those samples. We assume that the priors of the adjustable parameters $\mathbf{w}$ and $\boldsymbol{\lambda}$ are statistically independent.

Low complexity is modeled with sparsity of the adjustable parameter vectors $\mathbf{w}$ and $\boldsymbol{\lambda}$. This sparsity property, in addition to reducing model complexity has other advantages such as decreasing computational cost in the testing phase and preventing overfitting during the training step.

Conditional distribution of each output sample, assuming Gaussian distribution for noise, is written as

$$p(t_n|\mathbf{X}_n) = \mathcal{N}(t_n|\mathbf{w}^T\boldsymbol{\Phi}_n\boldsymbol{\lambda}, \sigma^2). \tag{11}$$

Due to the assumption of independent t_n's, the likelihood of the entire data set can be written as:

$$p(\mathbf{t}|\mathbf{w}, \boldsymbol{\lambda}, \sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\left\{\frac{\|\mathbf{t}-\mathbf{Y}\|^2}{2\sigma^2}\right\} \tag{12}$$

where

$$\mathbf{Y} = [\mathbf{w}^T\boldsymbol{\Phi}_1\boldsymbol{\lambda}, \mathbf{w}^T\boldsymbol{\Phi}_2\boldsymbol{\lambda}, \ldots, \mathbf{w}^T\boldsymbol{\Phi}_N\boldsymbol{\lambda}]^T. \tag{13}$$

Note that direct maximization of (12) will lead to overfitting that does not yield the desired sparse solution for $\mathbf{w}$ and $\boldsymbol{\lambda}$. To solve this problem, we take a Bayesian perspective approach and define prior distributions with hyperparameters on $\mathbf{w}$ and $\boldsymbol{\lambda}$. We consider a zero-mean Gaussian distribution with a corresponding variance for each parameter vector of $\mathbf{w}$ and $\boldsymbol{\lambda}$. Fig. 1 illustrates a graphical representation of the proposed model. Therefore,

$$\begin{aligned} p(\mathbf{w}|\boldsymbol{\alpha}_w) &= \prod_{i=1}^{N} \mathcal{N}(w_i|0, \alpha_{w_i}^{-1}) \\ &= \prod_{i=1}^{N} (2\pi\alpha_{w_i}^{-1})^{-\frac{1}{2}} \exp\{-\frac{\alpha_{w_i}}{2}(w_i-0)^2\} \\ &= (2\pi)^{-\frac{N}{2}} \left\{(\prod_{i=1}^{N} \alpha_{w_i}^{\frac{1}{2}}) \exp\{-\frac{1}{2}\mathbf{w}^T\mathbf{A}_w\mathbf{w}\}\right\} \end{aligned} \tag{14}$$

and

$$\begin{aligned} p(\boldsymbol{\lambda}|\boldsymbol{\alpha}_\lambda) &= \prod_{i=1}^{L} \mathcal{N}(\lambda_i|0, \alpha_{\lambda_i}^{-1}) \\ &= \prod_{i=1}^{L} (2\pi\alpha_{\lambda_i}^{-1})^{-\frac{1}{2}} \exp\{-\frac{\alpha_{\lambda_i}}{2}(\lambda_i-0)^2\} \\ &= (2\pi)^{-\frac{L}{2}} \left\{(\prod_{i=1}^{L} \alpha_{\lambda_i}^{\frac{1}{2}}) \exp\{-\frac{1}{2}\boldsymbol{\lambda}^T\mathbf{A}_\lambda\boldsymbol{\lambda}\}\right\} \end{aligned} \tag{15}$$

where $\boldsymbol{\alpha}_w = [\alpha_{w_1}, \alpha_{w_2}, \ldots, \alpha_{w_N}]^T$ and $\boldsymbol{\alpha}_\lambda = [\alpha_{\lambda_1}, \alpha_{\lambda_2}, \ldots, \alpha_{\lambda_L}]^T$ are hyperparameter vectors in which an individual hyperparameter α_{w_i} (precision of w_i) is associated independently to the weight w_i and, in the same way, an individual hyperparameter α_{λ_j} (precision of λ_j) is associated independently to the weight λ_j. Also $\mathbf{A}_w = diag(\alpha_{w_1}, \alpha_{w_2}, \ldots, \alpha_{w_N})$ and $\mathbf{A}_\lambda = diag(\alpha_{\lambda_1}, \alpha_{\lambda_2}, \ldots, \alpha_{\lambda_L})$ denote the diagonal matrices of the hyperparameters of $\mathbf{w}$ and $\boldsymbol{\lambda}$, respectively.

For the hierarchical prior of the model, we place Gamma distributions over hyperparameters $\boldsymbol{\alpha}_w$, $\boldsymbol{\alpha}_\lambda$ and $\beta^{-1} = \sigma^2$. Similar to the approach in [2], in order to achieve sparsity in $\mathbf{w}$ and $\boldsymbol{\lambda}$, we place the parameters of Gamma distributions to zero (i.e., a flat Gamma) that consequently results into Student-t marginalized priors over $\mathbf{w}$ and $\boldsymbol{\lambda}$.

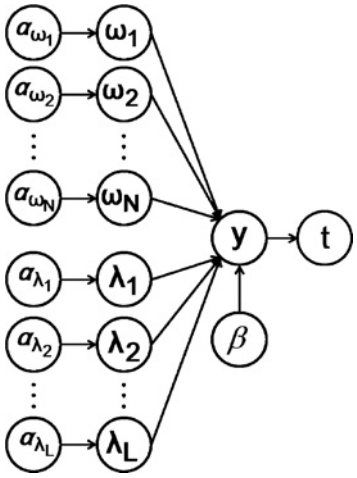

Fig. 1. Graphical model of RSFM.

B. Inference

The posterior distribution of unknown parameters of the proposed model is described as

$$p(\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda, \sigma^2|\mathbf{t}) = p(\mathbf{w}, \boldsymbol{\lambda}|\mathbf{t}, \boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda, \sigma^2) p(\boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda, \sigma^2|\mathbf{t}). \tag{16}$$

The first term on the right hand side is further extended as

$$\begin{aligned} p(\mathbf{w}, \boldsymbol{\lambda}|\mathbf{t}, \boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda, \sigma^2) &= \frac{p(\mathbf{t}|\mathbf{w}, \boldsymbol{\lambda}, \sigma^2) p(\mathbf{w}, \boldsymbol{\lambda}|\boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda)}{p(\mathbf{t}|\boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda, \sigma^2)} \\ &= \frac{p(\mathbf{t}|\mathbf{w}, \boldsymbol{\lambda}, \sigma^2) p(\mathbf{w}|\boldsymbol{\alpha}_w) p(\boldsymbol{\lambda}|\boldsymbol{\alpha}_\lambda)}{p(\mathbf{t}|\boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda, \sigma^2)} \end{aligned} \tag{17}$$

by using Bayes' rule and benefiting from the assumption on the independency of $\mathbf{w}$ and $\boldsymbol{\lambda}$ parameters. The numerator of (17) can be calculated by using (12), (14), and (15). The denominator of (17) is the integral of the numerator with respect to variables $\mathbf{w}$ and $\boldsymbol{\lambda}$. Unfortunately, this integral cannot be calculated analytically. The following equation shows a relation that we will later use to substitute the second term on the right hand side of (16).

$$p(\boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda, \sigma^2|\mathbf{t}) \propto p(\mathbf{t}|\boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda, \sigma^2) p(\boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda) p(\sigma^2). \tag{18}$$

Reliability and validity of such replacement is comprehensively described in [2].

With the assumption of uniform distribution for $p(\boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda)p(\sigma^2)$, we use type II maximum likelihood method [2] to determine the most probable values for $\boldsymbol{\alpha}_w$, $\boldsymbol{\alpha}_\lambda$, and σ^2 denoted as $\boldsymbol{\alpha}_{w_{MP}}$, $\boldsymbol{\alpha}_{\lambda_{MP}}$ and $\sigma^2{}_{MP}$, respectively. Such task requires calculation of $p(\mathbf{t}|\boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda, \sigma^2)$ that is analytically incomputable. In the RVM, the posterior over weights is Gaussian and available in closed form, however, introducing $\boldsymbol{\lambda}$ in the proposed model breaks that up and the posterior over $\mathbf{w}$ and $\boldsymbol{\lambda}$ is not analytically available anymore. Therefore, we used Laplace method [30] to approximate the posterior over $\mathbf{w}$ and $\boldsymbol{\lambda}$ as a production of two Gaussian distributions

over $\mathbf{w}$ and $\boldsymbol{\lambda}$ as explained in details in the following. To present calculations in a compact form, (7) and (8) are used to represent the formulation in a way that incorporates all the N training samples in one equation as

$$\mathbf{t}_{N\times 1} = [\mathbf{w}^T\boldsymbol{\Phi}_1\boldsymbol{\lambda} \quad \mathbf{w}^T\boldsymbol{\Phi}_2\boldsymbol{\lambda} \quad \dots \quad \mathbf{w}^T\boldsymbol{\Phi}_N\boldsymbol{\lambda}]^T + \boldsymbol{\epsilon}_{N\times 1}. \tag{19}$$

Representation of (19) in the following will help us to demonstrate later analysis in a simpler manner.

$$\boldsymbol{\Phi}_{wN\times L} = [\boldsymbol{\Phi}_1^T\mathbf{w} \quad \boldsymbol{\Phi}_2^T\mathbf{w} \quad \dots \quad \boldsymbol{\Phi}_N^T\mathbf{w}]^T \implies \mathbf{t} = \boldsymbol{\Phi}_w\boldsymbol{\lambda} + \boldsymbol{\epsilon}. \tag{20}$$

$$\boldsymbol{\Phi}_{\lambda N\times N} = [\boldsymbol{\Phi}_1\boldsymbol{\lambda} \quad \boldsymbol{\Phi}_2\boldsymbol{\lambda} \quad \dots \quad \boldsymbol{\Phi}_N\boldsymbol{\lambda}]^T \implies \mathbf{t} = \boldsymbol{\Phi}_\lambda\mathbf{w} + \boldsymbol{\epsilon}. \tag{21}$$

Considering the independency of $\mathbf{w}$ and $\boldsymbol{\lambda}$ and substituting (12), (14), and (16) in (17) results in

$$\begin{aligned} &p(\mathbf{w},\boldsymbol{\lambda}|\mathbf{t},\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_\lambda,\beta)p(\mathbf{t}|\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_\lambda,\beta) \\ &\quad = p(\mathbf{t}|\mathbf{w},\boldsymbol{\lambda},\beta)p(\mathbf{w},\boldsymbol{\lambda}|\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_\lambda) \\ &\quad = \frac{\beta^{\frac{N}{2}}}{(2\pi)^{\frac{2N+L}{2}}}|\mathbf{A}_w|^{\frac{1}{2}}|\mathbf{A}_\lambda|^{\frac{1}{2}} \\ &\qquad .\exp\{-\frac{\beta}{2}(\mathbf{t}-\boldsymbol{\Phi}_w\boldsymbol{\lambda})^T(\mathbf{t}-\boldsymbol{\Phi}_w\boldsymbol{\lambda}) \\ &\qquad\qquad -\tfrac{1}{2}\mathbf{w}^T\mathbf{A}_w\mathbf{w} - \tfrac{1}{2}\boldsymbol{\lambda}^T\mathbf{A}_\lambda\boldsymbol{\lambda}\} \end{aligned} \tag{22}$$

in which $\beta = \sigma^{-2}$ is the noise precision (inverse of variance).

Using the Laplace method [30], we approximate the posterior $p(\mathbf{w},\boldsymbol{\lambda}|\mathbf{t},\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_\lambda,\beta)$ on the left side of (22) with quadratic terms with respect to $\mathbf{w}$ and $\boldsymbol{\lambda}$. Notice that the mean vector $\boldsymbol{\mu}_w$ and the covariance matrix $\boldsymbol{\Sigma}_w$ of $\mathbf{w}$ will depend on $\boldsymbol{\lambda}$, and conversely, the mean vector $\boldsymbol{\mu}_\lambda$ and the covariance matrix $\boldsymbol{\Sigma}_\lambda$ of $\boldsymbol{\lambda}$ will depend on $\mathbf{w}$. To solve this problem, we use an iterative Expectation Maximization (EM) algorithm so that in each step one of the variables $\mathbf{w}$ or $\boldsymbol{\lambda}$ is considered to be constant and the covariance matrix and mean vector of the other variable is computed. Comprehensive derivations of $(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$ and $(\boldsymbol{\mu}_\lambda, \boldsymbol{\Sigma}_\lambda)$ will be presented in the following.

The power term of the right side of (22) can be expressed in two ways.

$$\begin{aligned} -E\{\mathbf{w},\boldsymbol{\lambda}\} \triangleq\ & -\tfrac{\beta}{2}(\mathbf{t}^T\mathbf{t} - 2\mathbf{t}^T\boldsymbol{\Phi}_w\boldsymbol{\lambda} + \boldsymbol{\lambda}^T\boldsymbol{\Phi}_w^T\boldsymbol{\Phi}_w\boldsymbol{\lambda}) \\ & -\tfrac{1}{2}\boldsymbol{\lambda}^T\mathbf{A}_\lambda\boldsymbol{\lambda} - \tfrac{1}{2}\mathbf{w}^T\mathbf{A}_w\mathbf{w} \end{aligned} \tag{23}$$

$$\begin{aligned} -E\{\mathbf{w},\boldsymbol{\lambda}\} \triangleq\ & -\tfrac{\beta}{2}(\mathbf{t}^T\mathbf{t} - 2\mathbf{t}^T\boldsymbol{\Phi}_\lambda\mathbf{w} + \mathbf{w}^T\boldsymbol{\Phi}_\lambda^T\boldsymbol{\Phi}_\lambda\mathbf{w}) \\ & -\tfrac{1}{2}\boldsymbol{\lambda}^T\mathbf{A}_\lambda\boldsymbol{\lambda} - \tfrac{1}{2}\mathbf{w}^T\mathbf{A}_w\mathbf{w} \end{aligned} \tag{24}$$

where (23) assumes a fixed $\mathbf{w}$, and, in the same manner, (24) assumes a fixed $\boldsymbol{\lambda}$.

In order to determine $\boldsymbol{\mu}_\lambda$, (23) is differentiated with respect to $\boldsymbol{\lambda}$ and the result is set to zero. To obtain $\boldsymbol{\Sigma}_\lambda$ (23) is differentiated twice with respect to $\boldsymbol{\lambda}$, and then negated and inverted to yield the covariance matrix of $\boldsymbol{\lambda}$.

$$\boldsymbol{\mu}_\lambda = \frac{\beta}{2}(\frac{\beta}{2}\boldsymbol{\Phi}_w^T\boldsymbol{\Phi}_w + \mathbf{A}_\lambda)^{-1}\boldsymbol{\Phi}_w^T\mathbf{t} \tag{25}$$

$$\boldsymbol{\Sigma}_\lambda = (\frac{\beta}{2}\boldsymbol{\Phi}_w^T\boldsymbol{\Phi}_w + \mathbf{A}_\lambda)^{-1} \tag{26}$$

Using (26), we can simplify (25) as

$$\boldsymbol{\mu}_\lambda = \frac{\beta}{2}\boldsymbol{\Sigma}_\lambda\boldsymbol{\Phi}_w^T\mathbf{t}. \tag{27}$$

In the same way, we can calculate $\boldsymbol{\mu}_w$ and $\boldsymbol{\Sigma}_w$ by differentiating (24) with respect to $\mathbf{w}$ assuming $\boldsymbol{\lambda}$ to be fixed.

$$\boldsymbol{\Sigma}_w = (\frac{\beta}{2}\boldsymbol{\Phi}_\lambda^T\boldsymbol{\Phi}_\lambda + \mathbf{A}_w)^{-1} \tag{28}$$

$$\boldsymbol{\mu}_w = \frac{\beta}{2}\boldsymbol{\Sigma}_w\boldsymbol{\Phi}_\lambda^T\mathbf{t}. \tag{29}$$

Employing (25)–(28), the power term in (23) can be written as

$$\begin{aligned} -E\{\mathbf{w},\boldsymbol{\lambda}\} &= -\frac{1}{2}(\mathbf{w}-\boldsymbol{\mu}_w)^T\boldsymbol{\Sigma}_w^{-1}(\mathbf{w}-\boldsymbol{\mu}_w) \\ &\quad -\frac{1}{2}(\boldsymbol{\lambda}-\boldsymbol{\mu}_\lambda)^T\boldsymbol{\Sigma}_\lambda^{-1}(\boldsymbol{\lambda}-\boldsymbol{\mu}_\lambda) \\ &\quad -\frac{\beta}{2}\mathbf{t}^T\mathbf{t} + \frac{1}{8}\beta^2\mathbf{t}^T\boldsymbol{\Phi}_\lambda\boldsymbol{\Sigma}_w\boldsymbol{\Phi}_\lambda^T\mathbf{t} \\ &\quad +\frac{1}{8}\beta^2\mathbf{t}^T\boldsymbol{\Phi}_w\boldsymbol{\Sigma}_\lambda\boldsymbol{\Phi}_w^T\mathbf{t} \\ &= -\frac{1}{2}(\mathbf{w}-\boldsymbol{\mu}_w)^T\boldsymbol{\Sigma}_w^{-1}(\mathbf{w}-\boldsymbol{\mu}_w) \\ &\quad -\frac{1}{2}(\boldsymbol{\lambda}-\boldsymbol{\mu}_\lambda)^T\boldsymbol{\Sigma}_\lambda^{-1}(\boldsymbol{\lambda}-\boldsymbol{\mu}_\lambda) \\ &\quad -\frac{1}{2}[\mathbf{t}^T(\frac{2}{\beta}I_N + \boldsymbol{\Phi}_\lambda\mathbf{A}_w^{-1}\boldsymbol{\Phi}_\lambda^T)^{-1}\mathbf{t}] \\ &\quad -\frac{1}{2}[\mathbf{t}^T(\frac{2}{\beta}I_N + \boldsymbol{\Phi}_w\mathbf{A}_\lambda^{-1}\boldsymbol{\Phi}_w^T)^{-1}\mathbf{t}]. \end{aligned} \tag{30}$$

Using determinant identity[1] we can write the left side of (22) as

$$\begin{aligned} &p(\mathbf{w},\boldsymbol{\lambda}|\mathbf{t},\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_\lambda,\beta)p(\mathbf{t}|\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_\lambda,\beta) \\ &\quad = \mathcal{N}_w(\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)\mathcal{N}_\lambda(\boldsymbol{\mu}_\lambda,\boldsymbol{\Sigma}_\lambda) \\ &\qquad .\frac{2^N}{(2\pi)^{\frac{N}{2}}\beta^{\frac{N}{2}}}|\frac{2}{\beta}I_N + \boldsymbol{\Phi}_\lambda\mathbf{A}_w^{-1}\boldsymbol{\Phi}_\lambda^T|^{-\frac{1}{2}} \\ &\qquad .|\frac{2}{\beta}\mathbf{I}_N + \boldsymbol{\Phi}_w\mathbf{A}_\lambda^{-1}\boldsymbol{\Phi}_w^T|^{-\frac{1}{2}} \\ &\qquad .\exp\{-\frac{1}{2}[\mathbf{t}^T(\frac{2}{\beta}\mathbf{I}_N + \boldsymbol{\Phi}_\lambda\mathbf{A}_w^{-1}\boldsymbol{\Phi}_\lambda^T)^{-1}\mathbf{t}] \\ &\qquad\qquad -\frac{1}{2}[\mathbf{t}^T(\frac{2}{\beta}\mathbf{I}_N + \boldsymbol{\Phi}_w\mathbf{A}_\lambda^{-1}\boldsymbol{\Phi}_w^T)^{-1}\mathbf{t}]\} \end{aligned} \tag{31}$$

where $\mathcal{N}_w(\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)$ denotes a normal distribution on $\mathbf{w}$ with mean of $\boldsymbol{\mu}_w$ and covariance of $\boldsymbol{\Sigma}_w$ and $\mathcal{N}_\lambda(\boldsymbol{\mu}_\lambda,\boldsymbol{\Sigma}_\lambda)$ denotes a normal distribution on $\boldsymbol{\lambda}$ with mean of $\boldsymbol{\mu}_\lambda$ and covariance of $\boldsymbol{\Sigma}_\lambda$.

C. Hyperparameters Optimization

The type II maximum likelihood method for hyperparameters optimization can be written as

$$\{\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_\lambda,\beta\}_{MP} = \text{argmax}_{\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_\lambda,\beta}L \tag{32}$$

where

$$\begin{aligned} L = \log\{&\frac{2^N}{(2\pi)^{\frac{N}{2}}\beta^{\frac{N}{2}}}|\frac{2}{\beta}I_N + \boldsymbol{\Phi}_\lambda\mathbf{A}_w^{-1}\boldsymbol{\Phi}_\lambda^T|^{-\frac{1}{2}} \\ &.|\frac{2}{\beta}\mathbf{I}_N + \boldsymbol{\Phi}_w\mathbf{A}_\lambda^{-1}\boldsymbol{\Phi}_w^T|^{-\frac{1}{2}} \\ &.\exp\{-\frac{1}{2}[\mathbf{t}^T(\frac{2}{\beta}\mathbf{I}_N + \boldsymbol{\Phi}_\lambda\mathbf{A}_w^{-1}\boldsymbol{\Phi}_\lambda^T)^{-1}\mathbf{t}] \\ &\qquad -\frac{1}{2}[\mathbf{t}^T(\frac{2}{\beta}\mathbf{I}_N + \boldsymbol{\Phi}_w\mathbf{A}_\lambda^{-1}\boldsymbol{\Phi}_w^T)^{-1}\mathbf{t}]\}\}. \end{aligned} \tag{33}$$

[1] $|\mathbf{A}||\beta^{-1}\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T| = |\beta^{-1}\mathbf{I}||\mathbf{A} + \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi}|$

Algorithm 1 Relevance Sample-Feature Machine

Input: $D = \{\mathbf{X}_n, t_n\}_{n=1}^{N}$ where $\mathbf{X}_n = \{\mathbf{x}_{n1}, \mathbf{x}_{n2}, \ldots, \mathbf{x}_{nL}\}$: training data set; N = the number of samples, L = the number of features for each sample.
Output: model parameters: relevance samples, relevance features and their corresponding weights ($\mathbf{w}$, $\boldsymbol{\lambda}$); variance of noise (σ^2).
Initialization: Initialize $\boldsymbol{\alpha}_w$ and $\boldsymbol{\alpha}_\lambda$.
Iterate
Calculate $\boldsymbol{\Sigma}_w$, $\boldsymbol{\mu}_w$, $\boldsymbol{\Sigma}_\lambda$ and $\boldsymbol{\mu}_\lambda$ based on equations (26)-(29).
Update $\boldsymbol{\alpha}_w$, $\boldsymbol{\alpha}_\lambda$, β based on equations (34)-(36).
if $\alpha_{w_k} >$ Certain Big Value
    Omit k^{th} sample from relevance samples.
if $\alpha_{\lambda_k} >$ Certain Big Value
    Omit k^{th} feature from relevance features.
Terminate
when maximum changes in $\boldsymbol{\alpha}_w$ and $\boldsymbol{\alpha}_\lambda$ in two consecutive iterations are less than a certain small value.

Since there is no closed-form solution for this problem, an iterative EM method is utilized [2]. Equation (33) is differentiated with respect to $log\alpha_{w_i}$ and the result is set to zero. By using the method in [2] we obtain the following update equations:

$$\alpha_{w_i}^{new} = \frac{\gamma_{w_i}}{\mu_{w_i}^2}, \qquad \gamma_{w_i} = 1 - \alpha_{w_i}\Sigma_{w_{ii}}. \tag{34}$$

In the same way, we obtain

$$\alpha_{\lambda_i}^{new} = \frac{\gamma_{\lambda_i}}{\mu_{\lambda_i}^2}, \qquad \gamma_{\lambda_i} = 1 - \alpha_{\lambda_i}\Sigma_{\lambda_{ii}}. \tag{35}$$

Finally, differentiating L with respect to $log\beta$ and equating the result to zero yields

$$\beta^{new} = \frac{\frac{N}{2} - \sum_{i=1}^{N}\gamma_{w_i} - \sum_{j=1}^{L}\gamma_{\lambda_j}}{\frac{1}{4}\|\mathbf{t} - \boldsymbol{\Phi}_\lambda\boldsymbol{\mu}_w\|^2 + \frac{1}{4}\|\mathbf{t} - \boldsymbol{\Phi}_w\boldsymbol{\mu}_\lambda\|^2}. \tag{36}$$

Therefore, the learning algorithm consists of iterating (34), (35), and (36) in addition to updating (26), (27), (28), and (29) as described in Algorithm 1. Note that use of the EM method guarantees convergence only to a local optimum.

D. Prediction

After training of the parameters, the optimum values for hyperparameters are employed to obtain the posterior distribution conditioned with $\boldsymbol{\alpha}_{w_{MP}}$, $\boldsymbol{\alpha}_{\lambda_{MP}}$ and β_{MP}. Therefore, the output distribution of t^* corresponding to the test data $\mathbf{X}^*$ can now be obtained as

$$\begin{aligned} p(t^*|\mathbf{t}) &= \int p(t^*|\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda, \sigma^2) \\ &\quad .p(\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda, \sigma^2|\mathbf{t})d\mathbf{w}d\boldsymbol{\lambda}d\boldsymbol{\alpha}_w d\boldsymbol{\alpha}_\lambda d\sigma^2 \\ &= \int p(t^*|\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda, \sigma^2)p(\mathbf{w}, \boldsymbol{\lambda}|\boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda, \sigma^2, \mathbf{t}) \\ &\quad .p(\boldsymbol{\alpha}_w, \boldsymbol{\alpha}_\lambda, \sigma^2|\mathbf{t})d\mathbf{w}d\boldsymbol{\lambda}d\boldsymbol{\alpha}_w d\boldsymbol{\alpha}_\lambda d\sigma^2, \end{aligned} \tag{37}$$

and by determination of $\boldsymbol{\alpha}_{w_{MP}}$, $\boldsymbol{\alpha}_{\lambda_{MP}}$ and β_{MP}

$$\begin{aligned} &p(t^*|\mathbf{t}, \boldsymbol{\alpha}_{w_{MP}}, \boldsymbol{\alpha}_{\lambda_{MP}}, \sigma^2_{MP}) \\ &\quad = \int p(t^*|\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha}_{w_{MP}}, \boldsymbol{\alpha}_{\lambda_{MP}}, \sigma^2_{MP}) \\ &\qquad .p(\mathbf{w}, \boldsymbol{\lambda}, |\boldsymbol{\alpha}_{w_{MP}}, \boldsymbol{\alpha}_{\lambda_{MP}}, \sigma^2_{MP}, \mathbf{t})d\mathbf{w}d\boldsymbol{\lambda}. \end{aligned} \tag{38}$$

Since the terms in the integral are Gaussian, the integral can be calculated simply as

$$p(t^*|\mathbf{t}, \boldsymbol{\alpha}_{w_{MP}}, \boldsymbol{\alpha}_{\lambda_{MP}}, \sigma^2_{MP}) = N_{t^*}(y^*, (\sigma^2)^*) \tag{39}$$

where

$$y^* = \boldsymbol{\mu}_w^T\boldsymbol{\Phi}^*\boldsymbol{\mu}_\lambda \tag{40}$$

and

$$(\sigma^2)^* = trace\{\boldsymbol{\Gamma}\boldsymbol{\Sigma}_w\boldsymbol{\Gamma}^T\} + \boldsymbol{\mu}_w^T\boldsymbol{\Phi}^*\boldsymbol{\Sigma}_\lambda\boldsymbol{\Phi}^*\boldsymbol{\mu}_w \tag{41}$$

where $\boldsymbol{\Gamma}_{N\times L}$ is obtained from Cholesky decomposition of matrix $\boldsymbol{\Phi}^*(\boldsymbol{\Sigma}_\lambda + \boldsymbol{\mu}_\lambda\boldsymbol{\mu}_\lambda^T)\boldsymbol{\Phi}^{*T}$, i.e.,

$$\boldsymbol{\Phi}^*(\boldsymbol{\Sigma}_\lambda + \boldsymbol{\mu}_\lambda\boldsymbol{\mu}_\lambda^T)\boldsymbol{\Phi}^{*T} = \boldsymbol{\Gamma}^T\boldsymbol{\Gamma}. \tag{42}$$

E. RSFM for Classification

It is straightforward to derive the RSFM for the task of classification from the regression described in the previous section.

In the case of binary classification, the posterior probability of membership in one of the classes should be predicted for a new input data. Following the method described by Tipping [2], a sigmoid function $\sigma(y) = \frac{1}{1+e^{-y}}$ is used to generalize the linear model (1) as

$$p(\mathbf{t}|\mathbf{w}, \boldsymbol{\lambda}) = \prod_{n=1}^{N}[\sigma(y(\mathbf{x}_n, \mathbf{w}, \boldsymbol{\lambda}))]^{t_n}[1 - \sigma(y(\mathbf{x}_n, \mathbf{w}, \boldsymbol{\lambda}))]^{1-t_n} \tag{43}$$

where $t_n \in \{0, 1\}$. We use the model $y = \boldsymbol{\Phi}_\lambda\mathbf{w}$ or $y = \boldsymbol{\Phi}_w\boldsymbol{\lambda}$, with two sets of unknown parameters ($\mathbf{w}$ and $\boldsymbol{\lambda}$) that is an extended version of what Tipping has used in [2] with only a single set of parameters. Therefore, our update equations become

$$\boldsymbol{\Sigma}_w = (\boldsymbol{\Phi}_\lambda^T\mathbf{B}\boldsymbol{\Phi}_\lambda + \mathbf{A}_w)^{-1} \tag{44}$$

$$\boldsymbol{\mu}_w = \boldsymbol{\Sigma}_w\boldsymbol{\Phi}_\lambda^T\mathbf{B}\mathbf{t} \tag{45}$$

$$\boldsymbol{\Sigma}_\lambda = (\boldsymbol{\Phi}_w^T\mathbf{B}\boldsymbol{\Phi}_w + \mathbf{A}_\lambda)^{-1} \tag{46}$$

$$\boldsymbol{\mu}_\lambda = \boldsymbol{\Sigma}_\lambda\boldsymbol{\Phi}_w^T\mathbf{B}\mathbf{t} \tag{47}$$

where $\mathbf{B} = diag[\beta_1, \beta_2, \ldots, \beta_N]$ and $\beta_n = \sigma\{y(\mathbf{x}_n)\}[1 - \sigma\{y(\mathbf{x}_n)\}]$.

We are currently extending the proposed method to the multiclass case using the standard multinomial form for the likelihood as

$$p(\mathbf{t}|\mathbf{w}, \boldsymbol{\lambda}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\sigma(y_k(\mathbf{x}_n, \mathbf{w}_k, \boldsymbol{\lambda}_k))^{t_{nk}} \tag{48}$$

where K denotes the number of classes and the output y_k corresponds to the k^{th} class and has its own coefficients $\mathbf{w_k}$ and $\boldsymbol{\lambda_k}$ and associated hyperparameters $\boldsymbol{\alpha}_{w_k}$ and $\boldsymbol{\alpha}_{\lambda_k}$.

III. Evaluation Results

In this section, the proposed method is evaluated on five types of data sets: 1) two synthetic data sets to show the ability of RSFM against the curse of dimensionality and elimination of irrelevant features, 2) three real benchmark data sets from UCI machine learning repository[2] to compare the performance of RSFM with other existing algorithms, 3) a large publicly available data set[3], 4) two high dimensional gene expression data sets and 5) our generated data set to test RSFM on human cognitive based object recognition.

A. Results on Synthetic Data

In the first set of experiments, the performance of RSFM in presence of irrelevant features is evaluated in two scenarios by testing on two sets of synthetic data. The first scenario evaluates the robustness of RSFM against increasing number of irrelevant features on a linear classification problem. The second one is an evaluation in the presence of noisy features on a nonlinear classification problem.

1) *Linear problem:* In the first scenario, the robustness of RSFM is evaluated against increasing the number of irrelevant features. To illustrate this, a set of synthetic data is generated. This data set consists of two Gaussian classes with unit variance and means

$$\boldsymbol{\mu}_1 = -\boldsymbol{\mu}_2 = [\frac{1}{\sqrt{2}} \ \frac{1}{\sqrt{2}} \ \underbrace{0 \ 0 \ \ldots \ 0}_{\text{p-2 zeros}}]^T. \tag{49}$$

Independently of feature dimensionality p, the optimal classifier for this data set is linear and the optimal Bayes error rate is given by $\phi(-1|0, 1) = 0.1587$. This optimal classifier only uses the first two dimensions of the data.

Fig. 2 compares the performance of RSFM with three other existing algorithms on this data set. All of these algorithms are trained using 200 samples (100 samples from each class) and are tested on 500 samples from each class. The average error rates over 20 random samples for each data dimension are reported in Fig. 2. The curves of JCFO and JFSCL in Fig. 2 are reproduced using the results reported in [22] and [23]. As Fig. 2 illustrates the influence of adding irrelevant features to RVM confuses the classifier, while the proposed method keeps almost a constant error rate that is very close to the Bayes error rate. Performances of feature selection methods JCFO, JFSCL, and RSFM are almost the same against adding irrelevant features. The minimum error rate belongs to JCFO in this experiment. This result is predictable because this set of synthetic data is well designed for the JCFO with linear kernel as reported in [22].

2) *Nonlinear problem:* In the second scenario, a nonlinear synthetic data set is generated similar to [24]. This data set consists of 52-dimensional samples in which two dimensions of 52 are relevant features and the rest of the features are noisy and irrelevant. The data is produced as follows: for the first class, the first two features $\{x_1, x_2\}$ are drawn from two distributions $N(\mu_{11}, \Sigma)$ and $N(\mu_{12}, \Sigma)$ with equal probabilities

[2]Available at www.stats.ox.ac.uk/pub/PRNN.

[3]Available at www.cs.toronto.edu/ roweis/data.html

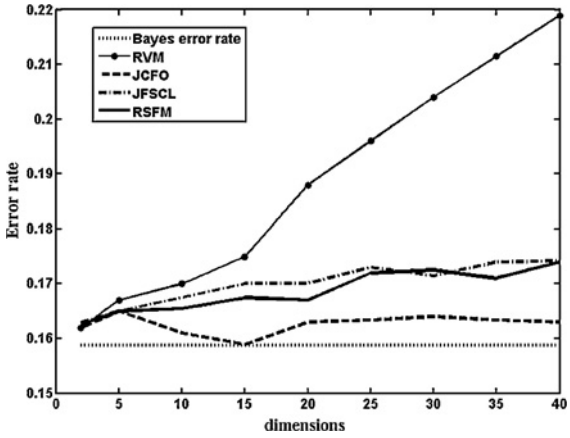

Fig. 2. Comparison of classification performance between JFSCL, JCFO, RSFM, and RVM using the synthetic data.

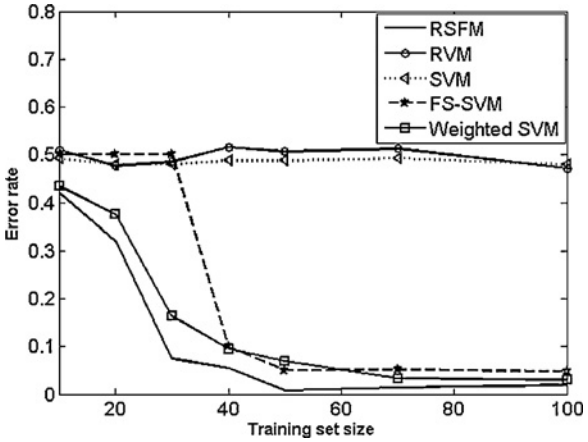

Fig. 3. Comparison of classification performance between RSFM, RVM, SVM, FS-SVM, and weighted SVM using the synthetic data.

where $\mu_{11} = \{-\frac{3}{4}, -3\}$, $\mu_{12} = \{\frac{3}{4}, 3\}$, and $\Sigma = I$. Similarly, for the second class, the first two features are generated from two normal distributions with equal probabilities and means of $\mu_{21} = \{3, -3\}$ and $\mu_{22} = \{-3, 3\}$. The rest of the features x_i, $i = 3, ..., 52$ are noisy and drawn from a normal distribution $N(0, 20)$. Classes 1 and 2 are equi-probable.

Fig. 3 shows the performance results with RSFM and four other related algorithms, SVM, RVM, feature selection for SVM (FS-SVM) [24] and weighted SVM [25]. The curves in Fig.3 are the average test error rate of 30 runs over a testing set of 500 samples for different training set sizes. The results show that RVM and SVM are highly misled with 50 noisy features so that they cannot classify the samples at all and the error rate is around 50%. In contrast, the embedded feature selection methods, including RSFM, FS-SVM [24], and weighted SVM [25] overcome the problem by eliminating the noisy features performing the classification task properly. As depicted in Fig.3 RSFM outperforms the method presented in [24] in classification performance for even small training sizes. The performance of RSFM is very close to weighted SVM, however, RSFM has the advantages of sparser trained model and probabilistic prediction.

TABLE I
ACCURACY OF DIAGNOSTIC CLASSIFICATION WITH BENCHMARK DATA SETS IN TERMS OF NUMBER OF ERRORS

Classifier	Pima	Crabs	WBC
Linear discriminant [32]	67	3	19
Neural network [33]	75	3	N/A
Gaussian process [33]	67	3	8
SVM [32]	64	4	9
Logistic regression [33]	66	4	N/A
RVM [2]	65	0	9
Sparse probit regression [31]	62	0	9
JCFO [22]	64	0	8
RSFM	66	0	9

B. Results on Benchmark Data Sets

In the second set of experiments, the performance of RSFM is evaluated using a set of benchmark data sets for nonlinear classifier. These three data sets are the Pima Indians diabetes, the Leptograpsus crabs, and the Wisconsin breast cancer (WBC). The details of these data sets can be found in [31]. The Pima data set consists of 532 samples with seven measured features for each sample. Similar to the reported experiments in the literature [22], [23], [31], we used 200 predefined samples for training and 332 samples for testing. In the crabs data set, there are 200 samples and five geometric measurements for each sample. In this case, we used 80 predefined samples for training and 120 remaining samples for testing. In the WBC case, there are 569 samples in the data set and 30 numerical measurements for each sample. In this problem, 300 predefined samples are used for training and 269 samples for testing. To apply RSFM to these problems, we normalized the input data to zero-mean and unit variance as in [22]. The number of errors for the proposed method and other state-of-the-art algorithms in this field is reported in Table I. The performance of the proposed method is the best or very close to the best in each column of Table I. Although RSFM performance in terms of error rate is very close to the other methods reported in Table I, there is a significant difference. RSFM is able to select the features and samples simultaneously just like JCFO. The RSFM outperforms JCFO in the two aspects: 1) in RSFM the kernel width parameter is not used for feature selection and so can be employed for improving the performance. For example, we can use multikernel methods in RSFM, 2) the computational complexity of RSFM is at the same order of RVM and so it is better than JCFO (as reported in [22] JCFO suffers from slow speed and high complexity).

In this experiment, the kernel width for the proposed method is chosen based on using cross-validation. RSFM selected 2 out of 80 samples and 3 out of 5 features in the crabs data set, 5 out of 200 samples and 5 out of 7 features in the Pima data set, and 6 out of 300 samples and 7 out of 30 features in the WBC data set. Therefore, RSFM can find a sparse classifier for these data sets both in sample and feature domains. The RSFM classifier sparsity is better or at least equal to JCFO (which is reported in [22]).

C. Results on Handwritten Digit Recognition

In the third set of experiments, RSFM is evaluated on MNIST, a large publicly available data set. MNIST consists of 28×28 handwritten digit images. Each class of a digit includes about 6000 samples for training and 1000 samples for testing. To evaluate RSFM as a binary classifier, we focused the experiments on discriminating the two digits 0 and 8 and also the most challenging two digits 9 and 4. First each image in the data set is vectorized to a 784×1 vector, then the values of vectors are normalized by 255. Finally, in order to investigate the ability of RSFM in eliminating the noisy features, 200 random features drawn from the standard normal distribution are padded to each sample.

The MNIST is a large database containing enough instances for training and testing. Accordingly, we used three disjoint sets for training, validation, and testing. In our setup, the validation set is used to adjust the kernel width of the RSFM. This set has the same size as the training set. The results of the experiments with the testing set are presented in Table II. For each training sample size (N) of this table, N random samples of the database were chosen as the training set and another N random samples as the validation set. After tuning the kernel width parameter on the validation set for the best accuracy, we tested the trained model on the predefined testing set including about 2000 samples. These stages were repeated for 10 runs in each case. The average results of these tests are reported in Table II. The same procedure was performed for the competing method Weighted SVM where the validation set was used for tuning its error-margin trade-off parameter C.

Table II summarizes the tests with the RSFM and the weighted SVM on all of the 1991 test samples after training them with different training sample sizes in a digit 4 versus digit 9 classification problem. The results in Table II show that the error rate of the RSFM in distinguishing 4 against 9 is very close to the weighted SVM. However, the number of relevance samples in the RSFM is smaller than the weighted SVM. Moreover, the training times of both algorithms presented in this table demonstrate that the training time with RSFM is smaller in comparison with the weighted SVM, the advantage increasing for larger training sizes. The same conclusions can be drawn from the digit 8 versus digit 0 recognition test also included in the table.

D. Results on Gene Expression Data Sets

In this section, the RSFM is further evaluated on two high-dimensional data sets. AML/ALL[4] and Colon[5] are two well-known gene expression data sets [22] that are employed in a set of experiments described here. The data sets include samples with a high number of features. AML/ALL consists of 72 samples each with 7129 features and Colon includes 62 instances each with 2000 attributes. In these two data sets most of the features are irrelevant [22]. Therefore, applying a classification method with feature selection capability is desirable. Table III illustrates the results of evaluating RSFM

[4]acute myeloid leukemia/acute lymphoblastic leukemia: http://www.genome.wi.mit.edu/mpr/table-AML-ALL-samples.rtf
[5]http://microarray.princeton.edu/oncology/affydata/index.html

TABLE II
PERFORMANCE OF RSFM AND WEIGHTED SVM ON MNIST DATA SET FOR 4 VERSUS 9 AND 8 VERSUS 0 RECOGNITION

# training Samples	9 versus 4								8 versus 0							
	Weighted SVM				RSFM				Weighted SVM				RSFM			
	# SV	# SF	Acc. (%)	time (S)	# RV	# RF	Acc. (%)	time (S)	# SV	# SF	Acc. (%)	time (S)	# RV	# RF	Acc. (%)	time (S)
100	55	60	89.50	2.38	9	42	88.25	0.72	21	24	97.75	2.76	5	56	97.34	0.36
200	72	76	89.95	6.83	18	84	88.80	1.42	23	25	97.29	6.38	6	70	97.13	0.59
300	56	108	89.85	8.22	20	112	91.46	2.31	26	29	98.47	10.72	5	42	97.65	1.18
400	48	94	90.96	13.21	20	84	90.06	2.34	26	56	98.16	17.30	9	42	98.41	1.42
600	71	127	90.10	25.36	31	84	90.39	3.49	26	57	98.87	32.43	5	28	97.96	3.20
1000	81	97	93.92	68.65	33	98	91.62	5.77	41	60	97.29	86.41	6	28	97.75	5.22

TABLE III
ACCURACY OF DIAGNOSTIC CLASSIFICATION WITH GENE EXPERSSION DATA SETS

Classifier	AML/ALL	Colon
Adaboosting (Decision stumps) [34]	95.8	72.6
SVM (Linear Kernel) [34]	94.4	77.4
SVM (Quadratic Kernel) [34]	95.8	74.2
Logistic regression [22]	97.2	71.0
RVM [22]	97.2	88.7
Sparse probit regression [22]	97.2	88.7
Sparse probit regression (Linear Kernel) [22]	97.2	91.9
Sparse probit regression (Quadratic Kernel) [22]	95.8	84.6
JCFO (Linear Kernel) [22]	100.0	96.8
JCFO (Quadratic Kernel) [22]	98.6	88.7
RSFM	97.2	88.7

as well as other competing methods on these data sets. The accuracy is computed using a leave-one-out cross-validation procedure exactly as reported in [22]. As shown, the results are the same as RVM and very close to the JCFO.

E. Results on Human-Based Object Recognition

1) *Cognitive Views:* A theory in neuroscience states that if one's standpoint is appropriate in observing an object partially, his/her mind would virtually reconstruct the rest of the 3-D object if he/she has already known that object [24]. From this theory, we infer that a human being gets visual cognition by observing enough samples of an object class from all possible view angles and obtaining a sparse representation of views from a sparse representation of samples. We will show in this section that this behavior is properly modeled by the proposed RSFM in the binary classification. Views of an object can be regarded as features of that object. In this section, we will use feature and view interchangeably. The RSFM finds the sparse representation of the class members (relevance samples or R.S.), and the sparse representation of the view angles (relevance features (views) or R.F.) that are sufficient for recognizing a particular set of objects. These selected view angles are what we refer to as cognitive views.

Applying RSFM for 3-D objects is too complex. A simplified environment for 2-D object classification via object boundary curves is presented in this section. A single view in the 2-D case is a segmented part of the boundary curve that has much lower complexity than a single view in a 3-D case which in itself is a 2-D picture.

2) *Feature Extraction:* The previously explained 2-D environment includes different types of boundary curves as objects for which feature extraction will be realized as follows.

1) *Step1:* Obtain a silhouette image and extract 1D representation of the silhouette image using centroidal distance profile (CDP) [35], [36].
2) *Step2:* Divide the CDP into overlapping intervals that are unwrapped segments of the boundary curve.

Output of the second step is the unwrapped version of the views. In fact, views are the boundary segments that could be seen when an imaginary observer with a limited view angle stands at the center-of-mass of the object. This imaginary observer spans the 360° with several views. Clearly, the number of views depends on the observer angle-of-view.

Since the observer stands at the object center of mass, the views are robust against translation. As the observer's angle-of-view is fixed, for the same but scaled objects this method yields the same views but only with different mean values. This issue is compensated by making the mean value of the resultant views equal to zero. Rotation could also be compensated by applying a chain code to the boundary curves.

Therefore, we will have a number of segments from a zero mean CDP, called views, which are invariant with respect to rotation, scaling, and translation. Cognitive views are those that RSFM will choose as the sparse set of views during its training stage.

3) *Generated Data Set:* The performance of the proposed algorithm is evaluated on a 2-D data set. The data set consists of five classes of 2-D objects: Apple brand logos, mugs, directions, bottles, and cars. We generated this data set by presenting a sample for each of these objects to 28 people and asking them to draw four samples of each exemplary object. In this way, every person intuitively drew four objects based on the presented sample picture. As a result, a rich data set with high diversity that is necessary for algorithm evaluation was obtained. Some samples of this data set are presented in Fig. 4.

For any particular object, several views were extracted using the method explained in the previous subsection. Initially, the center of polar coordinates is placed at the center-of-mass of the object. Then, the 360° of the angular coordinate is

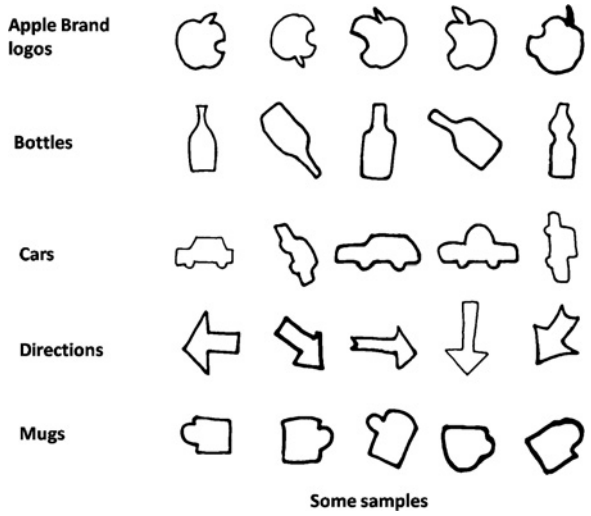

Fig. 4. Some samples of our data set.

quantized into 100 points. Each object of the data set would be presented by a 100 point-centroidal distance vector. This vector is divided into equally segmented intervals. Different number of views with corresponding dimensions could be obtained for an object by changing the length of its segments.

4) *Experiment Setup:* In this experiment, the performance of the proposed RSFM method is evaluated by comparing it with the regular RVM. Although, the idea of cognitive views proposed in this paper is embedded only in RSFM, comparison of RSFM and RVM can demonstrate the efficiency of RSFM in classification task.

For the case of RSFM, we partition the 100-point vector assigned to each object into 10 views. In contrast, there are no views in the case of RVM and the full 100-point vector would be directly used by the RVM.

Since there are five classes of objects in the data set, there are ten possible experiments of binary classification. In the upcoming subsections, the accuracy of the proposed model will be evaluated by the misclassification error rate. In addition, the number of employed relevance samples and relevance views provide a measure of the system complexity.

5) *Misclassification Error:* Fig. 5 summarizes the performance comparison of RSFM and RVM on binary classification experiments using our data set. Since the number of samples in each experiment is small (224 samples), we implemented an eight-fold cross validation in each case and reported the average accuracy along with the error bars in Fig. 5. This figure shows the average object classification accuracy for RSFM and RVM in ten different experiments. As depicted in Fig. 5, the performances of RSFM and RVM are almost the same on an average. However, RSFM uses only a few numbers of views (CDP segments) while RVM requires the full length CDP. It is worth mentioning that the number of misclassified objects using the RSFM does not exceed 2 out of 28 on average where the worst-case scenario is attributed to the classification experiment of Apple brand logos versus Bottles.

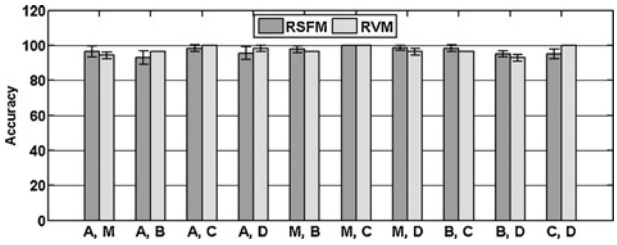

Fig. 5. Comparison of RSFM and RVM on binary classification experiments: Classification accuracy on 10 cases: Apple brand logos versus Mugs (A, M), Apple brand logos versus Bottles (A, B), Apple brand logos versus Directions (A, D), Mugs versus Bottles (M, B), Mugs versus Cars (M, C), Mugs versus Directions (M, D) Bottles versus Cars (B, C), Bottles versus Directions (B, D), Cars versus Directions (C, D).

6) *System Complexity:* System complexity is correlated with the model sparsity. RSFM model is designed to be sparse in two domains, samples and features. As a result, the system complexity of the RSFM is less than the RVM, which is sparse only in sample domain. Table II summarizes the performance comparison of RSFM and RVM. On average, RSFM uses 3.3 relevance samples for a classification task while this value is 3.5 when using RVM. Thus, RSFM achieves almost the same level of sparsity in the sample domain as RVM. The advantage of our proposed method can be verified by considering the number of relevance views in RSFM learned model. On average, only 2.5 views out of 10 are required by RSFM to reach almost the same classification error when compared to RVM that uses all available visual information. While RVM employs the full length CDP vector, RSFM requires only a fraction of the CDP called views. To quantitatively compare the complexities, we define the visual points used in the learned model (VPULM) to determine the amount of required visual information for achieving a specified accuracy.

$$VPULM = \begin{cases} \#ofR.S. \times \#ofR.V. \times 10 & \text{for RSFM} \\ \#ofR.S. \times 100 & \text{for RVM} \end{cases} \tag{50}$$

As Table IV illustrates, the error rates of RSFM and RVM are very close but RSFM benefits from much smaller VPULM values. This reduces the system complexity of our proposed method when compared to RVM.

As evaluated by our extensive experiments the computational cost of RSFM is very close to the RVM. The training time of RSFM is very close to that of RVM and the testing time is less than RVM due to feature pruning.

7) *Discussion on Cognitive Views:* As described earlier, our proposed algorithm is inspired by human visual object recognition and is capable of finding a few number of views that are sufficient for recognition. This stands against exploiting the complete length of an object boundary for performing such task. In this section, we describe the idea of cognitive views in the light of the observations obtained from our experiments.

As Table IV shows, RSFM completes its training stage for the experiment of Bottles versus Cars (B versus C) classification by designating two objects as the relevance samples. These objects are illustrated in Fig. 6. In spite

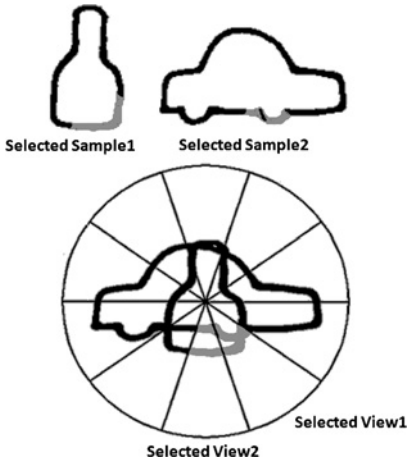

Fig. 6. Relevance samples and relevance views in Bottles versus Cars classification experiment.

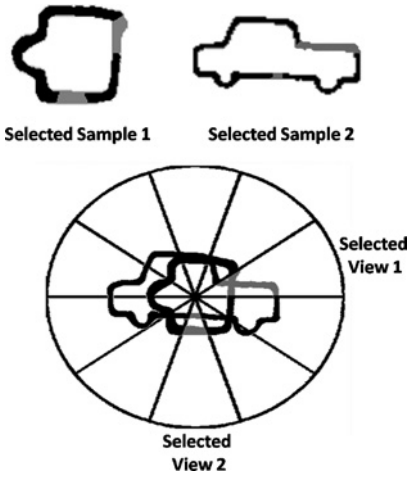

Fig. 7. Relevance samples and relevance views in Mugs versus Cars classification experiment.

of a considerable within-class variation in the data set, the RSFM learning algorithm is capable of performing an accurate generalization by selecting only one sample from each class as their corresponding prototypes. Both RSFM and RVM carry B versus C classification with 100% accuracy. However, while RVM uses the complete CDP of the silhouettes of two relevance samples (200 visual points), RSFM finds only two views of the two samples that are sufficient to differentiate between the two classes. These two views each with a length of 10 points result in only 40 visual points versus the 200 ones that RVM uses for performing the same task. The two relevance views chosen by RSFM are depicted in Fig. 6.

Fig. 7 shows relevance samples and their highlighted relevance views in the experiment of M versus C classification. It is interesting how the learning machine would choose a different car object from the class of cars when it is going to be differentiated from the mug objects rather than bottles.

Comparison of Figs. 6 and 7 shows that RSFM has selected neither the same car object as its relevance sample nor the same views. This is due to the similarity of the view sampling the car's hood and the corresponding view in many of the bottle objects in our data set. So, using this view in the

Fig. 8. Few bottles with the same behavior at the view corresponding to the car's hood.

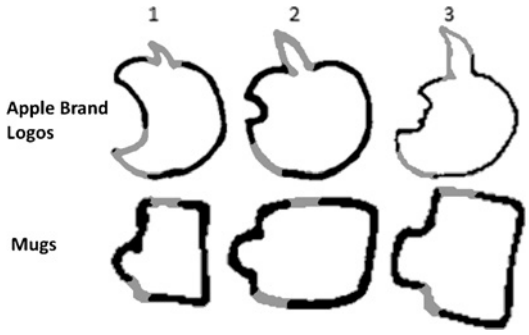

Fig. 9. Relevance samples and relevance views in Apple brand logos versus Mugs classification experiment.

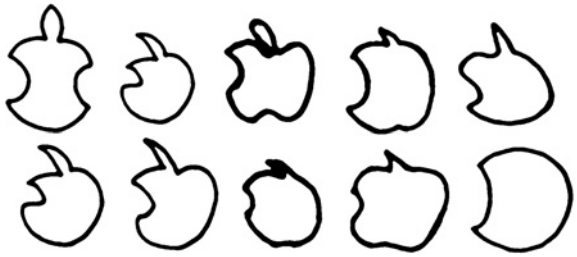

Fig. 10. Within-class variation in Apple brand logo class. Varied bitten parts in the top row and varied stalk parts in the down row.

classification of bottles versus cars will most probably result in several misclassified objects. Fig. 8 shows several of these bottle objects. In contrast, the view of the car's hood is a good discriminative one in Mugs versus Cars classification as can be seen in Fig. 7.

Fig. 9 shows six relevance samples (3 Apple brand logos and 3 mugs) that RSFM has selected for the classification experiment of Apple brand logos versus Mugs. As depicted, the selected view on the top is very discriminative. However, some Apple brand logo objects (illustrated in Fig. 10) do not have a distinctive stalk. Therefore, RSFM selects another cognitive view to discriminate those controversial objects from mugs. This view is on the bottom-left corner. Because of the morphologic diversity among mugs from horizontal expansion to vertical expansion, the proposed algorithm selects 3 mug objects as the relevance samples instead of only one. Variation among Apple brand logo objects especially in the stalk and bitten parts, shown in Fig. 10, forces RSFM to select three relevance samples from Apple brand logo class.

Two examples of misclassified objects are shown in Fig. 11. The object on the left is misclassified because the corresponding view to stalk part is relatively flat in this object and more similar to the associated view in mugs rather than Apple brand logos. For the misclassified object on the right, despite well discriminative view of the stalk, the other view is more similar

TABLE IV
SYSTEM COMPLEXITY AND PERFORMANCE COMPARISONS OF RSFM AND RVM. ABBREVIATIONS ARE THE SAME AS THOSE IN FIG. 5

Classification Experiment	# of Relevance Samples		# of Relevance Views for RSFM	# of Visual points used in Learned Model (out of 11200)		Error rate (%)		Training Time (seconds)	
	RSFM	RVM		RSFM	RVM	RSFM	RVM	RSFM	RVM
A versus. M	6	3	2	120	300	1.786	0.893	0.5587	0.2421
A versus B	5	11	2	100	1100	2.679	3.571	0.5598	0.3176
A versus C	1	2	1	10	200	0	0.893	0.8705	0.6063
A versus D	2	2	3	60	200	0	0.893	0.5606	0.7838
M versus B	7	5	3	210	500	1.786	2.679	0.5234	0.3548
M versus C	2	2	2	40	200	0	1.786	0.4583	0.7040
M versus D	2	4	3	60	400	0	0.893	0.5368	0.4125
B versus C	2	2	2	40	200	0	0	0.4846	0.5275
B versus D	1	2	3	30	200	2.679	0.893	0.8478	0.6146
C versus D	5	2	4	200	200	8.036	4.465	0.5710	0.4572

Fig. 11. Misclassified objects in Apple brand logos versus Mugs classification experiment.

to the bottom-left views of the first and third R.S. mugs in Fig. 9.

F. Computational Complexity

The RSFM algorithm is an iterative calculation of (34), (35), and (36) alongside the equations (26), (27), (28), and (29) as described in Algorithm 1 in Section II-C. Computation of these equations is obtained via Cholesky decomposition of terms $\frac{\beta}{2}\mathbf{\Phi}_{\lambda}^{T}\mathbf{\Phi}_{\lambda} + \mathbf{A}_{w}$ and $\frac{\beta}{2}\mathbf{\Phi}_{w}^{T}\mathbf{\Phi}_{w} + \mathbf{A}_{\lambda}$ which are of the orders $O(N^3)$ and $O(L^3)$, respectively. In practical applications $N >> L$, thus the computational cost is comparable to that of the classical RVM ($O(N^3)$).

IV. CONCLUSION

A. Discussion

In this paper, we presented a novel machine learning algorithm for joint feature and sample selection. To this end we extended the classical RVM formulation and introduced a novel separable model in sample and feature domains. We also presented the Bayesian inference of the proposed model and the learning algorithm based on marginal likelihood maximization using an EM procedure. Briefly, the main offerings of the RSFM can be summarized as

1) determination of relevant samples and at the same time relevant features by introducing a model including two separate parameter sets for sample and feature domains and without employing the kernel scaling parameter for feature selection;
2) fast convergence due to the use of EM-based optimization in comparison to the JCFO that uses conjugate gradient;
3) providing probabilistic predictions in contrast to embedded feature selection methods on SVM presented in [24] and [25] that only produce a point estimation;
4) robustness against irrelevant and noisy features.

The proposed algorithm was tested on two synthetic data sets, three benchmark data sets, a large handwritten digit data set, two high-dimensional gene expression data sets and our generated data set. Experimental results on the synthetic data sets indicate that our proposed method (RSFM) is robust against added irrelevant features and successful in elimination of noisy features and its performance is almost the same as other embedded methods (JCFO, JFSCL, and two extensions on SVM) and superior to RVM and SVM. We also compared our proposed method with other state-of-the-art algorithms (e.g., Gaussian Process, Sparse probit regression, and JCFO) on three benchmark data sets. This set of experiments shows that the classification accuracy of RSFM is superior or very close to the best algorithm.

While JCFO achieves feature selection through adjusting the kernel scaling parameter, JFSCL achieves this by adding a term to the RVM cost function. In contrast, simultaneous feature and sample selection is obtained in our proposed RSFM by introducing a model including two separate parameter sets. RSFM is potentially more capable than JCFO since in RSFM the kernel width parameter is not used for feature selection and so can be employed for improving the performance. Furthermore, multikernel methods [15] are shown to improve the classification performance for example in RVM. RSFM is capable of employing multikernel methods and we are currently extending the proposed RSFM to a multikernel one. Finally, both JCFO and JFSCL suffer from heavy computational complexities as the authors reported in [22] and [23]. Our proposed RSFM, however, enjoys a low computational complexity (which is almost the same as RVM) due to the use of EM-based optimization. Therefore, the training time of the RSFM is much less than JCFO (which uses conjugate gradient for its optimization) and very close to RVM as the presented evaluation results in this paper show.

B. Future Work

As our extensive experimental results on various data sets show, the proposed RSFM algorithm is successful in simulta-

neous feature selection and classification. However, there are some drawbacks for the RSFM alongside all its benefits. First, using the EM-based maximization guarantees convergence only to a local optimum. However, in many experiments we have performed, we have not encountered any noticeable problem. Second, the matrix inversion in calculation of Σ_λ and Σ_w is a limitation in implementing RSFM for large data sets. Of course this is a common problem for all kernel-based methods [22]. We are currently solving this limitation for large data sets by extending the RSFM to an incremental one based on the fast marginal likelihood maximization [16]. Extending the RSFM to a multiclass one is another future plan based on the multiclass model presented in [17].

ACKNOWLEDGMENT

The authors would like to thank the authors of the papers [22], [24], and [25] for helping by providing their codes.

REFERENCES

[1] V. Vapnik, S. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," in *Proc. Annu. Conf. Neural Inform. Process. Systems*, vol. 9. Dec. 1997, pp. 281–287.

[2] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Jun. 2001.

[3] S. Wong and R. Cipolla, "Real-time adaptive hand motion recognition using a sparse Bayesian classifier," in *Proc. Comput. Vision Human-Computer Interaction*, vol. 3766. Oct. 2005, pp. 170–179.

[4] A. Thayananthan, R. Navaratnam, B. Stenger, P. Torr, and R. Cipolla, "Multivariate relevance vector machines for tracking," in *Proc. Eur. Conf. Comput. Vision*, 2006, pp. 124–138.

[5] C. Silva and B. Ribeiro, "Scaling text classification with relevance vector machines," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, vol. 5. Oct. 2006, pp. 4186–4191.

[6] A. Agarwal and B. Triggs, "3-D human pose from silhouettes by relevance vector regression," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, vol. 2. Jun.–Jul. 2004, pp. II-882–II-888.

[7] L. Wei, Y. Yang, R. M. Nishikawa, M. N. Wernick, and A. Edwards, "Relevance vector machine for automatic detection of clustered microcalcifications," *IEEE Trans. Med. Imaging*, vol. 24, no. 10, pp. 1278–1285, Oct. 2005.

[8] Y. Li, C. Campbell, and M. Tipping, "Bayesian automatic relevance determination algorithms for classifying gene expression data," *Bioinformatics*, vol. 18, no. 10, p. 1332, 2002.

[9] D. Yang, S. Zakharkin, G. Page, J. Brand, J. Edwards, A. Bartolucci, and D. Allison, "Applications of Bayesian statistical methods in microarray data analysis," *Amer. J. PharmacoGenomics*, vol. 4, no. 1, pp. 53–62, 2004.

[10] A. Lukic, M. Wernick, D. Tzikas, X. Chen, A. Likas, N. Galatsanos, Y. Yang, F. Zhao, and S. Strother, "Bayesian kernel methods for analysis of functional neuroimages," *IEEE Trans. Med. Imag.*, vol. 26, no. 12, pp. 1613–1624, Dec. 2007.

[11] O. Williams, A. Blake, and R. Cipolla, "Sparse Bayesian learning for efficient visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1292–1304, Aug. 2005.

[12] D. Tzikas, A. Likas, and N. Galatsanos, "Large scale multikernel relevance vector machine for object detection," *Int. J. Artif. Intell. Tools*, vol. 16, no. 6, p. 967, 2007.

[13] M. M. Kalayeh, T. Marin, P. H. Pretorius, M. N. Wernick, Y. Yang, and J. G. Brankov, "Channelized relevance vector machine as a numerical observer for cardiac perfusion defect detection task," in *Proc. SPIE*, vol. 7966. 2011, p. 79660E.

[14] T. Marin, M. M. Kalayeh, P. H. Pretorius, M. N. Wernick, Y. Yang, and J. G. Brankov, "Numerical observer for cardiac motion assessment using machine learning," in *Proc. SPIE*, vol. 7966, 2011, p. 79660G.

[15] M. M. Kalayeh, T. Marin, and J. G. Brankov, "Generalization evaluation of machine learning numerical observers for image quality assessment," *IEEE Trans. Nuclear Sci.*, vol. PP, no. 99, pp. 1–10, May 2013 (early access).

[16] M. Tipping and A. Faul, "Fast marginal likelihood maximization for sparse Bayesian models," in *Proc. 9th Int. Workshop Artif. Intell. Stat.*, vol. 1, no. 3. Jan. 2003.

[17] I. Psorakis, T. Damoulas, and M. Girolami, "Multiclass relevance vector machines: Sparsity and accuracy," *IEEE Trans. Neural Netw.*, vol. 21, no. 10, pp. 1588–1598, Oct. 2010.

[18] G. Humphreys, C. Price, and M. Riddoch, "From objects to names: A cognitive neuroscience approach," *Psychol. Res.*, vol. 62, no. 2, pp. 118–130, 1999.

[19] M. Riddoch and G. Humphreys, *Object Recognition*, B. Rapp, Ed. New York, NY, USA: Psychology Press, 2001.

[20] J. Ward, *The Student's Guide to Cognitive Neuroscience*, New York, NY, USA: Psychology Press, 2006.

[21] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[22] B. Krishnapuram, A. Harternink, L. Carin, and M. Figueiredo, "A Bayesian approach to joint feature selection and classifier design," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1105–1111, Sep. 2004.

[23] Á. Lapedriza, S. Seguí, D. Masip, and J. Vitrià, "A sparse Bayesian approach for joint feature selection and classifier learning," *Pattern Anal. Applicat.*, vol. 11, no. 3, pp. 299–308, 2008.

[24] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for SVMs," in *Proc. Annu. Conf. Advances Neural Information Processing Systems*, Apr. 2001, pp. 668–674.

[25] M. Nguyen and F. De la Torre, "Optimal feature selection for support vector machines," *Pattern Recognit.*, vol. 43, no. 3, pp. 584–591, 2010.

[26] P. Zhao and B. Yu, "Stagewise lasso," *J. Mach. Learn. Res.*, vol. 8, pp. 2701–2726, Dec. 2007.

[27] F. Bach, G. Lanckriet, and M. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 6.

[28] Y. Han and G. Liu, "Probability-confidence-kernel-based localized multiple kernel learning with norm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 827–837, Jun. 2012.

[29] R. Close, J. Wilson, and P. Gader, "A Bayesian approach to localized multikernel learning using the relevance vector machine," in *Proc. IEEE Int. Geoscience Remote Sens. Symp.*, Jul. 2011, pp. 1103–1106.

[30] D. MacKay, "The evidence framework applied to classification networks," *Neural Comput.*, vol. 4, no. 5, pp. 720–736, 1992.

[31] M. Figueiredo, "Adaptive sparseness for supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1150–1159, Sep. 2003.

[32] M. Seeger, "Bayesian model selection for support vector machines, Gaussian processes, and other kernel classifiers," in *Proc. Annu. Conf. Adv. Neural Inform. Processing Systems*, vol. 12. Jun. 2000, pp. 603–609.

[33] C. Williams and D. Barber, "Bayesian classification with gaussian processes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1342–1351, Dec. 1998.

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

[35] S. Dubois and F. Glanz, "An autoregressive model approach to two-dimensional shape classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 1, pp. 55–66, Jan. 1986.

[36] S. Jaggi, W. Karl, S. Mallat, and A. Willsky, "Silhouette recognition using high-resolution pursuit," *Pattern Recognit.*, vol. 32, no. 5, pp. 753–772, 1999.

Yalda Mohsenzadeh received the B.Sc. degree in electrical engineering from Ferdowsi University of Mashhad, Mashhad, Iran, in 2007 and the M.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2009. She is currently pursuing the Ph.D. degree in electrical engineering and is a member of Multimedia Signal Processing Research Lab, Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran.

Her current research interests include machine learning, pattern recognition, Bayesian learning, statistical signal processing, sparse representation, and compressive sensing.

Hamid Sheikhzadeh (M'03–SM'04) received the B.S. and M.S. degrees in electrical engineering from Amirkabir University of Technology, Tehran, Iran, in 1986 and 1989, respectively, and the Ph.D. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 1994.

He was a Faculty Member at the Electrical Engineering Department, Amirkabir University of Technology, until September 2000. From 2000 to 2008, he was a Principle Researcher with ON Semiconductor, Waterloo, ON, Canada. During this period, he developed signal processing algorithms for ultra-low-power and implantable devices leading to many international patents. Currently, he is a Faculty Member in the Electrical Engineering Department of Amirkabir University of Technology. His current research interests include signal processing, machine learning, biomedical signal processing and speech processing, with particular emphasis on speech recognition, speech enhancement, auditory modeling, adaptive signal processing, subband-based approaches, and algorithms for low-power DSP and implantable devices.

A. M. Reza (S'85–M'86–SM'06) received the B.S. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 1975, the M.S. degree in nuclear engineering from Massachusetts Institute of Technology, Cambridge, MA, USA, in 1978, and the Ph.D. degree in electrical engineering from the University of Wyoming, Laramie, WY, USA, in 1986.

Since 1986, he has been a Professor at the Department of Electrical Engineering and Computer Science, University of Wisconsin-Milwaukee, Milwaukee, WI, USA, where he is currently a Professor Emeritus. From July 2008 to June 2011, he was a visiting Professor at the Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran. His current research interest include particle filters, sensor networks, and compressive sensing.

Dr. Reza has an extensive publication record based on the results of his research with his colleagues and graduate students. His research activities are mainly in the areas of signal and image processing, hardware implementation, and pattern recognition. He has applied his research to target tracking, radar, sonar, seismic, and biomedical signals, as well as in image processing.

Najmehsadat Bathaee is currently pusuing the Ph.D. degree from the Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran. She received the B.Sc. and M.Sc. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2005 and 2007, respectively.

She is a member of the Multimedia Signal Processing Research Laboratory, Department of Electrical Engineering, Amirkabir University of Technology. In past years, she focused on wireless networks, large scale network system, and peer-to-peer networks. Her current research interests include non-parametric Bayesian modeling of speech and audio signals.

Mahdi M. Kalayeh received the B.Sc. and M.Sc. degrees in electrical engineering from the Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 2009, and Illinois Institute of Technology (IIT), Chicago, IL, USA, in 2010, respectively.

From 2009 to 2010, he was a Graduate Researcher at the Medical Imaging Research Center at IIT Tech Park. In 2011, he joined the Multimedia Signal Processing Research Laboratory, Tehran Polytechnic as a Visiting Researcher. Currently, he is a Research Assistant at the Center for Research in Computer Vision, University of Central Florida, Orlando, FL, USA, where he is pursuing the Ph.D. degree in computer science. His current research interests include computer vision, machine learning, and data mining.