Neurocomputing 000 (2017) 1–10

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Variational Relevant Sample-Feature Machine: A fully Bayesian approach for embedded feature selection

Ali Mirzaei[a,*], Yalda Mohsenzadeh[b], Hamid Sheikhzadeh[a]

[a] Amirkabir University of Technology, Tehran, Iran
[b] Massachusetts Institute of Technology (MIT), Cambridge, MA, USA

ARTICLE INFO

Article history:
Received 26 November 2015
Revised 27 January 2017
Accepted 12 February 2017
Available online xxx

Communicated by Shiliang Sun

Keywords:
Sparse Bayesian learning
Relevance Vector Machine
Feature selection
Classification
Regression

ABSTRACT

This paper presents a Bayesian learning approach for embedded feature selection. This approach employs a fully Bayesian framework to achieve a model which is sparse in both sample and feature domains. We introduce a novel multi-step algorithm based on Variational Approximation to efficiently compute all model parameters in order to optimize the *maximum a posteriori probability* (MAP) measure. Experiments on both synthetic and real datasets verify that the proposed method is successful in feature selection while achieving high accuracy in both regression and classification tasks. Compared to the existing methods, especially its non-fully Bayesian counterpart, the proposed algorithm results in much higher accuracies when the size of learning data is small. Moreover, the proposed method is more reliable (evident by less variance in accuracy) than other competing algorithms.

1. Introduction

Feature selection is one of the most important techniques which aim to improve the performance of learning methods. Feature selection and dimension reduction provide several advantages: first of all feature selection results in a low dimensional model which reduces the risk of *overfitting* and increases the generalization ability of the model. Moreover, it reduces the model complexity and leads to less computational burden [1] and it also provides insight into the underlying process that generated data [2]. One can classify the existing feature selection methods into three major categories: filtering, wrapper and embedded methods [2,3]. In filtering methods [4–7] the feature selection is performed independently before learning of the model and the features are ranked according to a criterion such as mutual information [5]. Wrapper methods [8–10] use a search algorithm on the feature space and evaluate all possible subsets of features on a specific model and select the best set of features based on that model performance. Typically, these methods perform better compared to filtering methods because they select relevant features according to the modeling accuracy. However, these methods require a much higher computational demand compared to filtering methods. The third group of feature selection algorithms is known as embedded methods [11–14] in which selection of relevant features and learning of the model are performed simultaneously. In other words, these methods find the most relevant features as a part of the model during the training stage. Moreover, the computational cost of embedded methods is much lower than wrapper methods which necessitate searching the whole feature space.

In this paper, we propose an embedded feature selection method based on a fully Bayesian framework for both regression and classification tasks. This method can be considered as a fully Bayesian version of the Relevant Sample-Feature Machine (RSFM) algorithm introduced in [11]. The RSFM includes a set of parameters and another set of hyper-parameters in its model. Then, maximizing the posterior probability and pruning small-valued parameters, the method obtains a model which is sparse both in sample and feature domains. In RSFM, the type-II maximum likelihood method is used to maximize the posterior probability. In this paper, we offer a new formulation and solution for the RSFM, based on a completely Bayesian framework through the use of Variational Byes (VB) inference method, thereby giving a posterior distribution over both parameters and hyper-parameters.

The novelties of the proposed method, called Variational RSFM (VRSFM) include: (i) presenting a fully Bayesian paradigm for RSFM, (ii) employing variational inference, (iii) applying VRSFM to both regression and classification tasks and (iv) obtaining a lower bound for sigmoid function using local variational approximation in the classification case.

* Corresponding author.
E-mail addresses: ali_mirzaei@aut.ac.ir (A. Mirzaei), yalda@mit.edu (Y. Mohsenzadeh), hsheikh@aut.ac.ir (H. Sheikhzadeh).

http://dx.doi.org/10.1016/j.neucom.2017.02.057
0925-2312/

The rest of this paper is structured as follows: in Section 2 basics of the RSFM are presented. Sections 3 and 4 introduce Variational Approximation and its application to the RSFM case. Section 5 includes performance evaluations and Section 6 concludes the paper.

2. Relevance Sample-Feature Machine

Our proposed model is based on the recently presented Relevance Sample Feature Machine (RSFM) which is an embedded feature selection approach [11]. Accordingly, in this section we briefly explain the model and fundamental concepts of the RSFM.

Let $\{X_n, t_n\}_{n=1}^{n=N}$ be the input-output pairs of a given dataset, where X_n is a $1 \times L$ vector that denotes the nth sample of input vectors and t_n is the corresponding output value of the n^{th} sample. t_n can either hold a real (in regression case) or an integer (in classification) value. The regression and classification problems are illustrated separately as follows.

2.1. RSFM model for regression

Following the standard probabilistic formulation [15], the RSFM considers outputs as a noisy function of inputs:

$$t_n = y_n(X_n) + \epsilon_n,$$

where $\{\epsilon_n\}_{n=1}^{n=N}$ are i.i.d. random variables with zero-mean Gaussian distribution with a variance of $\sigma^2 = \beta^{-1}$. In RSFM, the function y_n is defined as:

$$y_n = \boldsymbol{w}^T \Phi_n \boldsymbol{\lambda} \rightarrow p(t_n|\boldsymbol{w}, \boldsymbol{\lambda}, \beta) = \mathcal{N}(\boldsymbol{w}^T \Phi_n \boldsymbol{\lambda}, \beta^{-1}), \tag{1}$$

where

$$\boldsymbol{\Phi_n} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & K(X_{n1}, X_{11}) & \cdots & K(X_{nL}, X_{1L}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K(X_{n1}, X_{N1}) & \cdots & K(X_{nL}, X_{NL}) \end{pmatrix}, \tag{2}$$

and $\boldsymbol{w}$ and $\boldsymbol{\lambda}$ are model parameters which are $(N+1) \times 1$ and $(L+1) \times 1$ vectors, respectively. The vector $\boldsymbol{w}$ includes sample weights and the vector $\boldsymbol{\lambda}$ consists of features weights. X_{nj} denotes the jth feature of the nth sample in the dataset. $K(X_{nj}, X_{ij})$ is a predefined kernel function which is commonly assumed as a radial basis function, $K(x, y) = \exp(-\gamma(x-y)^2)$, where γ is known as kernel width parameter. The final goal in this model is to obtain optimized values for $\boldsymbol{w}$ and $\boldsymbol{\lambda}$. The RSFM uses a Bayesian approach to find these values. In order to avoid overfitting, a prior distribution is assumed for each parameter. More specifically, a zero-mean Gaussian distribution is assumed for each coefficient as:

$$p(\boldsymbol{w}|\boldsymbol{\alpha}_w) = \prod_{i=0}^{N} \mathcal{N}(0, \alpha_{w_i}^{-1}) \ , \ p(\boldsymbol{\lambda}|\boldsymbol{\alpha}_\lambda) = \prod_{j=0}^{L} \mathcal{N}(0, \alpha_{\lambda_j}^{-1}), \tag{3}$$

where α_{w_i} denotes the precision of the prior distribution on w_i and α_{λ_j} is the precision of the prior distribution on λ_j. Then by maximizing the posterior probability on weight parameters ($\boldsymbol{w}$ and $\boldsymbol{\lambda}$) and their corresponding hyper-parameters ($\boldsymbol{\alpha_w} = [\alpha_{w_0}, \ldots, \alpha_{w_N}]$ and $\boldsymbol{\alpha_\lambda} = [\alpha_{\lambda_0}, \ldots, \alpha_{\lambda_L}]$) and pruning the small coefficients, a model is obtained which is sparse both in sample and feature domains. To present future equations in a more compact form, the following parameters are defined:

$$\boldsymbol{\Phi_w}_{N\times(L+1)} = [\boldsymbol{\Phi_1^T w} \ \boldsymbol{\Phi_2^T w} \ \ldots \ \boldsymbol{\Phi_N^T w}]^T \rightarrow \boldsymbol{t} = \boldsymbol{\Phi_w \lambda} + \boldsymbol{\epsilon}, \tag{4}$$

$$\boldsymbol{\Phi_\lambda}_{N\times(N+1)} = [\boldsymbol{\Phi_1 \lambda} \ \boldsymbol{\Phi_2 \lambda} \ \ldots \ \boldsymbol{\Phi_N \lambda}]^T \rightarrow \boldsymbol{t} = \boldsymbol{\Phi_\lambda w} + \boldsymbol{\epsilon}, \tag{5}$$

where $\mathbf{t} = [t_1, t_2, \ldots, t_N]^T$ and $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \ldots, \epsilon_N]^T$.

2.2. RSFM model for classification

For binary classification, following the same procedure described in [15] and [11], the linear model (2) is generalized as

$$p(\boldsymbol{t}|\boldsymbol{w}, \boldsymbol{\lambda}) = \prod_{n=1}^{N} [\sigma(y(X_n; \boldsymbol{w}, \boldsymbol{\lambda})]^{t_n} [1 - \sigma(y(X_n; \boldsymbol{w}, \boldsymbol{\lambda}))]^{1-t_n}, \tag{6}$$

where σ is sigmoid function $\sigma(x) = 1/(1+\exp(-x))$ and $\{t_n\}_{n=1}^{N} \in \{0, 1\}$.

In the next section, a brief review of the Variational Approximation method is presented since this method is employed in developing our proposed learning method in this paper.

3. Variational Approximation

A probabilistic model usually consists of some latent and observed variables denoted by θ and D, respectively. The goal in training such models is to find the posterior distribution of latent variables given observed data, $p(\theta|D)$. Then using this posterior probability function, the expectation of the latent variables is calculated. However, practical calculation of this probability function is complex and analytically intractable. Moreover, even if one finds this posterior probability function, computing the expectations of latent variables, using this distribution is another complicated and intractable problem. This intractability in computing expectations of latent variables stems mainly from of two issues: first, for continuous variables, it is usually infeasible to find a closed-form for the integration; second, in practice latent variables are high dimensional which makes the integration over all latent variables numerically difficult and intractable [16].

One approach to resolve this analytical problem is to approximate the posterior probability function with a tractable one using Variational Approximation method [16]. This method has been widely used for inference of probabilistic models. Basically, this method finds $q(\theta)$ as an approximation of $p(\theta|D)$ such that the KL-divergence between $p(\theta|D)$ and $q(\theta)$ is minimized. If this minimization problem is unconstrained, then the trivial solution for this problem is $q(\theta) = p(\theta|D)$, an undesired return to the intractable integration. In Variational Approximation method, some constraints on $q(\theta)$ are imposed such that a tractable approximate function is obtained. To elaborate more on this method and the mentioned constraints, we briefly explain the Variational Approximation method in the following.

Using the chain rule we obtain the joint distribution $p(D, \theta)$. Therefore, the objective is to find an approximation for posterior distribution $p(\theta|D)$ and also model evidence $p(D)$. To this end, we decompose log of the model evidence as:

$$\ln p(D) = \mathcal{L}(q) + KL(q||p) \tag{7}$$

where

$$\begin{aligned} \mathcal{L}(q) &= \int q(\theta) \ln\{\frac{p(D, \theta)}{q(\theta)}\} d\theta \ , \ KL(q||p) \\ &= -\int q(\theta) \ln\{\frac{p(\theta|D)}{q(\theta)}\} d\theta. \end{aligned} \tag{8}$$

In Eq. (7), $KL(q||p)$ is the KL-divergence distance between two the functions of $q(\theta)$ and $p(\theta|D)$, and is always positive; therefore, $\mathcal{L}(q)$ can be considered as a lower bound for model evidence $p(D)$. The left hand side of Eq. (7) is independent of the function q; as a result, maximizing $\mathcal{L}(q)$ with respect to function $q(\theta)$ will minimize the KL-divergence between $q(\theta)$ and $p(\theta|D)$.

Assume the latent variables of model consist of M disjoint group $\theta = \{\theta_1, \theta_2, \ldots, \theta_M\}$. For obtaining a tractable approximate function ($q(\theta)$), a factorization constraint on $q(\theta)$ is imposed: $q(\theta) = \prod_{i=1}^{M} q_i(\theta_i)$. Please note that no more assumptions on $q(\theta)$

Please cite this article as: A. Mirzaei et al., Variational Relevant Sample-Feature Machine: A fully Bayesian approach for embedded feature selection, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.02.057

are made; in particular, no special form for individual factors $q_i(\theta)$ is considered. Typically, after substituting the factorized $q(\theta)$ into the lower bound ($\mathcal{L}(q)$), we can maximize the lower bound with respect to all functions $q_i(\theta_i)$. Using variational calculus [16] yields:

$$\ln q_j(\theta_j) = E_{i\neq j}(\ln p(D,\theta)) + const. \tag{9}$$

The above equation shows that the log of optimal solution is obtained by taking the expectation of log of joint distribution $P(D, \theta)$ over all latent variables except the variable that we are calculating its factor. The constant value allows to scale the probability function $q_j(\theta_j)$ such that its integration would be equal to one. In special probability functions like Gaussian and Gamma, this constant value is known as a function of parameters of that distribution [16,17].

4. Variational RSFM (VRSFM)

The RSFM [11] uses the Laplace method [16] to approximate the posterior distribution over the model parameters. Then it employs the type-II maximum likelihood method and Expectation-Maximization (EM) algorithm to find the optimum parameters and hyper-parameters that maximize the logarithm of the marginal likelihood. In this paper, we present a fully Bayesian framework for inference of the posterior distribution in the RSFM model. We use a Variational Approximation method [16] to find an approximate function for the posterior distribution.

4.1. Regression case

The goal of the proposed VRSFM is to model RSFM in a fully Bayesian framework; so we assume prior distributions on not only the parameters but also on the hyper-parameters of the model. The prior distributions for parameters are the same as RSFM which are presented in Eq. (3). However, additionally the following prior Gamma distributions are assumed for hyper-parameters of the model:

$$p(\beta) = \Gamma(a_\beta^0, b_\beta^0)\ ,\ \ p(\boldsymbol{\alpha_w}) = \prod_{i=0}^{N} \Gamma(a_{\alpha_{w_i}}^0, b_{\alpha_{w_i}}^0)\ ,$$

$$p(\boldsymbol{\alpha_\lambda}) = \prod_{j=0}^{L} \Gamma(a_{\alpha_{\lambda_j}}^0, b_{\alpha_{\lambda_j}}^0) \tag{10}$$

Note that in the original RSFM, the hyper-parameters were following a flat Gamma distribution (very close to uniform distribution). In contrast, to obtain a fully Bayesian model we assume a prior probability function on all hyper-parameters of the model. Fig. 1 shows the Probabilistic Graphical Model (PGM) for the VRSFM. In this model, we aim to find the posterior probability of all parameters and hyper-parameters after observing the training samples $\boldsymbol{t}$:

$$p(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha_w}, \boldsymbol{\alpha_\lambda}, \beta | \boldsymbol{t}) \tag{11}$$

Unfortunately, the above posterior distribution cannot be calculated analytically. For this reason, we approximate this function using the variational approximation method to achieve a tractable function. More specifically, we approximate the posterior distribution in Eq. (11) with a distribution q which can be factorized as:

$$\begin{aligned} p(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha_w}, \boldsymbol{\alpha_\lambda}, \beta | \boldsymbol{t}) &\approx q(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha_w}, \boldsymbol{\alpha_\lambda}, \beta) \\ &= q_w(\boldsymbol{w}) q_\lambda(\boldsymbol{\lambda}) q_{\alpha_w}(\boldsymbol{\alpha_w}) q_{\alpha_\lambda}(\boldsymbol{\alpha_\lambda}) q_\beta(\beta) \end{aligned} \tag{12}$$

Then we find q such that the KL-divergence between the two distributions (p and q) is minimized.

In order to calculate each factor in Eq. (12) using Eq. (9), we need the joint distribution of random variables. This joint distribution can be obtained from Fig. 1 and chain rule as:

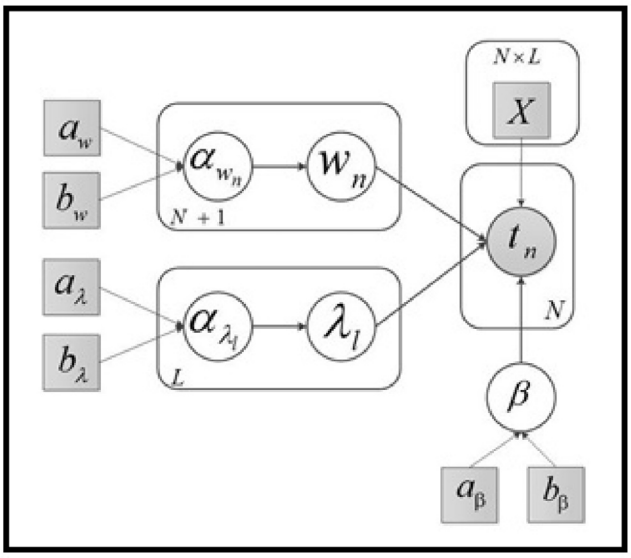

Fig. 1. Probability Graphical Model for VRSFM. The circles are random variable, the squares are parameters and the shaded shapes are observed variables.

$$\begin{aligned} &p(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha_w}, \boldsymbol{\alpha_\lambda}, \beta, \boldsymbol{t}) \\ &= p(\boldsymbol{t}|\boldsymbol{w}, \boldsymbol{\lambda}, \beta) p(\boldsymbol{w}|\boldsymbol{\alpha_w}) p(\boldsymbol{\lambda}|\boldsymbol{\alpha_\lambda}) p(\boldsymbol{\alpha_w}) p(\boldsymbol{\alpha_\lambda}) p(\beta). \end{aligned} \tag{13}$$

Therefore, using Eqs. (1), (3), (9), (10) and (13) we can calculate $q_w(\boldsymbol{w})$ as:

$$\begin{aligned} \ln q_w(\boldsymbol{w}) &= E_{\lambda,\alpha_w,\alpha_\lambda,\beta}(\ln p(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha_w}, \boldsymbol{\alpha_\lambda}, \beta, \boldsymbol{t})) \\ &= E_{\lambda,\alpha_w,\alpha_\lambda,\beta} \ln p(\boldsymbol{t}|\boldsymbol{w}, \boldsymbol{\lambda}, \beta) p(\boldsymbol{w}|\boldsymbol{\alpha_w}) p(\boldsymbol{\lambda}|\boldsymbol{\alpha_\lambda}) p(\boldsymbol{\alpha_w}) p(\boldsymbol{\alpha_\lambda}) p(\beta) \\ &= E_{\lambda,\beta} \ln p(\boldsymbol{t}|\boldsymbol{w}, \boldsymbol{\lambda}, \beta) + E_{\alpha_w} \ln p(\boldsymbol{w}|\boldsymbol{\alpha_w}) + const \\ &= E_{\lambda,\beta}(-\beta(\boldsymbol{t}-\boldsymbol{\Phi_\lambda w})^T(\boldsymbol{t}-\boldsymbol{\Phi_\lambda w})/2) - E_{\alpha_w}(\boldsymbol{w}^T \boldsymbol{A_w} \boldsymbol{w}/2) + const, \end{aligned}$$

where $\boldsymbol{A_w}$ is a $(N+1)\times(N+1)$ matrix and is defined as $\boldsymbol{A_w} = diag(\boldsymbol{\alpha_w})$. The log of $q_w(w)$ has a quadratic form with respect to $\boldsymbol{w}$ and therefore it is Gaussian. We calculate the mean and covariance of this distribution as:

$$\boldsymbol{\Sigma_w} = (E(\beta)E(\boldsymbol{\Phi_\lambda^T \Phi_\lambda}) + E(\boldsymbol{A_w}))^{-1}\ ,\ \boldsymbol{\mu_w} = \boldsymbol{\Sigma_w} E(\beta) E(\boldsymbol{\Phi_\lambda^T})\mathbf{t} \tag{14}$$

In a similar way, we can find $q_\lambda(\boldsymbol{\lambda})$ as a Gaussian distribution with the following mean vector and covariance matrix:

$$\boldsymbol{\Sigma_\lambda} = (E(\beta)E(\boldsymbol{\Phi_w^T \Phi_w}) + E(\boldsymbol{A_\lambda}))^{-1}\ ,\ \ \mu_\lambda = \Sigma_\lambda E(\beta) E(\Phi_w^t)\boldsymbol{t} \tag{15}$$

where $\boldsymbol{A_\lambda}$ is a $(L+1)\times(L+1)$ matrix which is defined as $\boldsymbol{A_\lambda} = diag(\boldsymbol{\alpha_\lambda})$. In the above equations we can obtain $E(\boldsymbol{\Phi_w^T \Phi_w})$ and $E(\boldsymbol{\Phi_w})$ as:

$$E(\boldsymbol{\Phi_w^T \Phi_w}) = \sum_{i=1}^{N} \boldsymbol{\Phi_i^T}(\boldsymbol{\Sigma_w} + \boldsymbol{\mu_w}\boldsymbol{\mu_w^T})\boldsymbol{\Phi_i},$$

$$E(\boldsymbol{\Phi_w^T}) = [\boldsymbol{\Phi_1^T \mu_w \Phi_2^T \mu_w} \cdots \boldsymbol{\Phi_N^T \mu_w}],$$

and similarly for $E(\Phi_\lambda^t \Phi_\lambda)$ and $E(\Phi_\lambda)$ we have:

$$E(\boldsymbol{\Phi_\lambda^T \Phi_\lambda}) = \sum_{i=1}^{N} \boldsymbol{\Phi_i}(\boldsymbol{\Sigma_\lambda} + \boldsymbol{\mu_\lambda}\boldsymbol{\mu_\lambda^T})\boldsymbol{\Phi_i^T},$$

$$E(\boldsymbol{\Phi_\lambda^T}) = [\boldsymbol{\Phi_1 \mu_\lambda \Phi_2 \mu_\lambda} \cdots \boldsymbol{\Phi_N \mu_\lambda}].$$

If we follow the same procedure for obtaining $q_{\boldsymbol{\alpha_w}}(\boldsymbol{\alpha_w})$, $q_{\alpha_\lambda}(\boldsymbol{\alpha_\lambda})$ and $q_\beta(\beta)$, we conclude that all these functions are Gamma distributions as:

$$q_{\alpha_{w_i}} = \Gamma(a_{w_i}^0 + 1/2, b_{w_i}^0 + E(w_i^2)/2) \tag{16}$$

$$q_{\alpha_{\lambda_j}} = \Gamma(a_{\lambda_j}^0 + 1/2, b_{\lambda_j}^0 + E(\lambda_j^2)/2) \tag{17}$$

$$q_\beta = \Gamma\left(a_\beta^0 + \frac{N+1}{2}, b_\beta^0 + \frac{1}{2}E((\boldsymbol{t}-\boldsymbol{y})^T(\boldsymbol{t}-\boldsymbol{y}))\right) \tag{18}$$

where in Eqs. (16) and (17) we have :

$$E(w_i^2) = \boldsymbol{\mu_w}(i)^2 + \boldsymbol{\Sigma_w}(i,i)\ ,\ E(\lambda_i^2) = \boldsymbol{\mu_\lambda}(i)^2 + \boldsymbol{\Sigma_\lambda}(i,i)$$

And in Eq. (18) we have:

$$E((\boldsymbol{t}-\boldsymbol{y})^T(\boldsymbol{t}-\boldsymbol{y})) = E((\boldsymbol{t}-\boldsymbol{\Phi_\lambda w})^T(\boldsymbol{t}-\boldsymbol{\Phi_\lambda w})) = \boldsymbol{t}^T\boldsymbol{t} - 2E(\boldsymbol{w})^T E(\boldsymbol{\Phi_\lambda})^T\boldsymbol{t} + E(\boldsymbol{w}^T\boldsymbol{\Phi_\lambda^T}\boldsymbol{\Phi_\lambda w})$$

$$E(\boldsymbol{w}^T\boldsymbol{\Phi_\lambda^T}\boldsymbol{\Phi_\lambda w}) = E(tr(\boldsymbol{\Phi_\lambda^T}\boldsymbol{\Phi_\lambda w w}^T)) = tr(E(\boldsymbol{\Phi_\lambda^T}\boldsymbol{\Phi_\lambda})E(\boldsymbol{ww}^T))$$

$$E(\boldsymbol{ww}^T) = \boldsymbol{\mu_w}\boldsymbol{\mu_w^T} + \boldsymbol{\Sigma_w}$$

Evidently Eqs. (14)–(18) are dependent on each other. Therefore, we employ an iterative algorithm to solve them. After some iterations, mean of some coefficients ($\boldsymbol{\lambda}$, $\boldsymbol{w}$) tends towards zero and accordingly those coefficients can be pruned. Algorithm 1 summarizes the various steps of learning for the VRSFM model. While the RSFM prunes the irrelevant samples or features by imposing an upper threshold on $\boldsymbol{\alpha_w}$ and $\boldsymbol{\alpha_\lambda}$, here we prune irrelevant features and samples based on a lower threshold on weights. Since in our fully Bayesian model we define a Gaussian prior probability for $\boldsymbol{\alpha_w}$ and $\boldsymbol{\alpha_\lambda}$, there is no reason that they go to infinity, as noted in [15].

Algorithm 1 Learning the parameters of model.

Require: $\mathbf{X} = [X_1, X_2, \ldots, X_N]$, $\mathbf{t}$
  while maximum changes in μ_w and μ_λ is more than a small value **do**
    calculate $\boldsymbol{\Sigma_w}$ and $\boldsymbol{\mu_w}$ using equations in (14)
    calculate $\boldsymbol{\Sigma_\lambda}$ and $\boldsymbol{\mu_\lambda}$ using equations in (15)
    calculate $\boldsymbol{a_w}$ and $\boldsymbol{b_w}$ using Eq. (16)
    calculate $\boldsymbol{a_\lambda}$ and $\boldsymbol{b_\lambda}$ using Eq. (17)
    calculate a_β and b_β using Eq. (18)
    if $|\mu_{w_i}| < TH_w \times max(\boldsymbol{\mu_w})$ **then**
      Prune i^{th} sample from relevant samples
    end if
    if $|\mu_{\lambda_j}| < TH_\lambda \times max(\boldsymbol{\mu_\lambda})$ **then**
      Prune jth feature from relevant features
    end if
  end while

4.2. Classification case

In a classification task, we cannot consider a Gaussian distribution for the model outputs (sample labels). For simplicity, a binary classification problem is assumed and a binomial distribution is considered on the class labels as:

$$p(\boldsymbol{t}) = \prod_{i=1}^{N} \sigma(y_n)^{t_n}(1-\sigma(y_n))^{1-t_n} \tag{19}$$

In the above equation, σ is the sigmoid function ($\sigma(x) = 1/(1+e^{-x})$). Please notice that in the classification model there is no regression error with the parameter β. Similar to the regression case, the prior distributions introduced in Eqs. (3) and (10) are assumed on the parameters and hyper-parameters of the model. Since the sigmoid function is a non-linear function we cannot follow the procedure we employed in the regression case for Eq. (19). This is because the expectation operator and a non-linear function are not interchangeable. Therefore, to resolve this problem, we use the following inequality introduced by Jaakkola and Jordan [18]

$$\sigma(y_n)^{t_n}(1-\sigma(y_n))^{1-t_n} = \sigma(z) \geq \sigma(\zeta)\exp\left(\frac{z-\zeta}{2} - f(\zeta)(z^2-\zeta^2)\right), \tag{20}$$

where $t = \{0,1\}$ and $z = (2t-1)y$ and $f(\zeta) = (1/4\zeta)\tanh(\zeta/2)$. Here ζ is known as variational parameter and the equality condition is obtained when $\zeta = z$. Therefore, we can write

$$p(\boldsymbol{t}|\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\alpha_w},\boldsymbol{\alpha_\lambda}) \geq p_{lb}(\boldsymbol{t}|\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\alpha_w},\boldsymbol{\alpha_\lambda},\boldsymbol{\zeta}) = \prod_{n=1}^{N}\sigma(\zeta_n)\exp\left(\frac{z_n-\zeta_n}{2} - f(\zeta_n)(z_n^2-\zeta_n^2)\right) \tag{21}$$

where $z_n = (2t_n-1)\boldsymbol{w^T\Phi_n\lambda}$ and p_{lb} is a lower bound for conditional probability. Using this inequality we can obtain a lower bound for joint distribution:

$$p(\boldsymbol{t},\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\alpha_w},\boldsymbol{\alpha_\lambda}) = p_{lb}(\boldsymbol{t},\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\alpha_w},\boldsymbol{\alpha_\lambda}) \geq p_{lb}(\boldsymbol{t}|\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\alpha_w},\boldsymbol{\alpha_\lambda},\boldsymbol{\zeta})p(\boldsymbol{w}|\boldsymbol{\alpha_w})p(\boldsymbol{\lambda}|\boldsymbol{\alpha_\lambda})p(\boldsymbol{\alpha_w})p(\boldsymbol{\alpha_\lambda}) \tag{22}$$

So we can find another approximated lower bound for evidence $p(\boldsymbol{t})$

$$p(\boldsymbol{t}) \geq \mathcal{L} \geq \int q(\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\alpha_w},\boldsymbol{\alpha_\lambda},\beta)\ln\left\{\frac{p_{lb}(\boldsymbol{t},\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\alpha_w},\boldsymbol{\alpha_\lambda})}{q(\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\alpha_w},\boldsymbol{\alpha_\lambda},\beta)}\right\} d\boldsymbol{w}d\boldsymbol{\lambda}d\boldsymbol{\alpha_w}d\boldsymbol{\alpha_\lambda}d\beta \tag{23}$$

Now we should maximize the new lower bound in the right hand side of Eq. (23) with respect to $q(.)$ and the variational parameter $\boldsymbol{\zeta}$. This is a constrained optimization problem subject to factorized form of q as introduced previously in Eq. (12) (note that for the case of classification, there is no β in the model). Using Eq. (9) we conclude that $q_w(\boldsymbol{w})$ will be Gaussian with the following parameters:

$$\boldsymbol{\Sigma_w^{-1}} = 2\sum_{n=1}^{N}[f(\zeta_n)(\boldsymbol{\Phi_n}(\boldsymbol{\mu_\lambda}\boldsymbol{\mu_\lambda^T} + \boldsymbol{\Sigma_\lambda})\boldsymbol{\Phi_n^T})] + E(\boldsymbol{A_w}), \tag{24}$$

$$\boldsymbol{\mu_w} = \boldsymbol{\Sigma_w}(\frac{1}{2}\sum_{n=1}^{N}(2t_n-1)\boldsymbol{\Phi_n})\boldsymbol{\mu_\lambda}, \tag{25}$$

where $E(\boldsymbol{A_w}) = diag(E(\alpha_{w1}), E(\alpha_{w2}), \ldots, E(\alpha_{wN}))$. Similarly, $q_\lambda(\boldsymbol{\lambda})$ will be a Gaussian distribution with the following parameters:

$$\boldsymbol{\Sigma_\lambda^{-1}} = 2\sum_{n=1}^{N}[f(\zeta_n)(\boldsymbol{\Phi_n^T}(\boldsymbol{\mu_w}\boldsymbol{\mu_w^T} + \boldsymbol{\Sigma_w})\boldsymbol{\Phi_n})] + E(\boldsymbol{A_\lambda}), \tag{26}$$

$$\boldsymbol{\mu_\lambda} = \boldsymbol{\Sigma_\lambda}\left(\frac{1}{2}\sum_{n=1}^{N}(2t_n-1)\boldsymbol{\Phi_n^T}\right)\boldsymbol{\mu_w}, \tag{27}$$

where $E(\boldsymbol{A_\lambda}) = diag(E(\alpha_{\lambda 1}), E(\alpha_{\lambda 2}), \ldots, E(\alpha_{\lambda L}))$. For the hyper-parameters of our model, $\boldsymbol{\alpha_w}$ and $\boldsymbol{\alpha_\lambda}$, we achieve the same equations as the regression case:

$$q_{\boldsymbol{\alpha_w}} = \prod_{i=1}^{N}\Gamma\left(a_w^0 + \frac{1}{2}, b_w^0 + \frac{1}{2}E(w_i^2)\right) \tag{28}$$

$$q_{\boldsymbol{\alpha_\lambda}} = \prod_{i=1}^{L}\Gamma\left(a_\lambda^0 + \frac{1}{2}, b_\lambda^0 + \frac{1}{2}E(\lambda_i^2)\right) \tag{29}$$

Maximizing the right hand side of Eq. (23) with respect to $\boldsymbol{\zeta}$ we obtain:

$$\zeta_n^2 = E(y_n^2) = E(\boldsymbol{\lambda^T\Phi_w^T\Phi_w\lambda}) = E(tr(\boldsymbol{\Phi_w^T\Phi_w\lambda\lambda^T})) = tr(E(\boldsymbol{\Phi_w^T\Phi_w})E(\boldsymbol{\lambda\lambda^T})) \tag{30}$$

Please cite this article as: A. Mirzaei et al., Variational Relevant Sample-Feature Machine: A fully Bayesian approach for embedded feature selection, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.02.057

5. Performance evaluation

In this section we evaluate the proposed algorithm on both regression and classification tasks.

5.1. Regression problems

The performance of VRSFM is compared to those of Relevance Vector Machine (RVM) [15], Relevance Sample-Feature Machine (RSFM) [11], Support Vector Regression (SVR) [19] and several feature selection methods which are used with SVR such as: Hilbert-Schmidt Feature Selection (HSFS) [7], Gaussian Recursive Feature Elimination (RFE_GAU) [9], Laplace Recursive Feature Elimination (RFE_LAP) [9], Sample Label Canonical Correlation Analysis (SLCCA) [20] and Random SLCCA (RSLCCA) [21]. RVM is a sparse Bayesian learning method which does not perform any feature selection. RSFM is an extension of RVM which is sparse in both sample and feature domains. SVR is a regression method based on Support Vector Machines, not including any feature selection. Therefore, it uses all the features to learn a regression model. HSFS is a filtering method for feature selection. The SVR model is used to learn a model based on selected features with HSFS. The family of RFE's is a set of feature selection methods which rank the features according to their importance for classification or regression. The RFE Methods are classified in the wrapper feature selection methods. In RFE_GAU and RFE_LAP, the SVR is used for evaluating all the possible subsets of features. SLCCA and RSLCCA are two filtering feature selection methods which rank the features according to their canonical correlation with labels. Similar to other filter feature selection methods, SVR is used as a regressor for these two methods.

In order to evaluate our method as well as other competing methods, we selected 30 realizations of data and randomly partitioned the samples in each dataset to 3 subsets as training, validation and test sets (denoted as N_{tr}, N_{val}, and N_{tst} samples, respectively) for each of 30 realizations, and reported the averaged results over these 30 realizations. In each algorithm, there are some parameters that should be optimized through evaluating on the validation set.

- *RVM, RSFM and VRSFM*
  In RVM, RSFM and the proposed method (VRSFM), the kernel width is found by a validation procedure. The kernel width is selected from the grid of $[2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6]$ by training a model using the training set and evaluating it on the validation set for each of the 30 realizations. In VRSFM, all parameters of the Gamma prior probabilities (Eqs. (10)) are set to 10^{-6} for all experiments. We prune irrelevant samples or features when the corresponding coefficients are less than 0.01 times of maximum coefficient ($Th_w = 0.01, TH_\lambda = 0.01$).
- *SVR*
  SVR has three parameters that should be chosen carefully: trade-off parameter (C), inverse of kernel width (γ) and the value of epsilon (ϵ) in epsilon-SVR method. During validation procedure, the grid

  $$[2^{-2}, 2^{-1}, 2^0, 2, 2^2, 2^3, 2^4, 2^5, 2^6] \times [2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2] \times [2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1]$$

  is searched for [C, γ, ϵ], respectively.
- *HSFS*
  HSFS has a parameter and this parameter (λ) is searched over the set of {10, 20, 50, 100, 200}. The performance of the parameter and selected features are evaluated with the SVR model using the evaluation datasets. The best parameter and subset of features are chosen using the validation datasets.
- *RFE, SLCCA and RSLCCA*
  As it was already explained, the SVR functions as the evaluation algorithm for the top rank features in REF, SLCCA and RSLCCA. Therefore, in these methods for learning a regression model in addition to SVR parameters after ranking the features, the optimum number of features should also be determined. The three parameters of SVR are searched on the same grid (as the one employed for SVR) and then the accuracy of the learned model using k top rank features is evaluated on the validation set. Next, the SVR parameters and also the top rank features (along with their number (k)) which minimize the MSE ($MSE = \sum_{n=1}^{N}(\hat{y}_n - y_n)^2/N$) are selected. Afterwards, using these known parameters which are optimized independently on each realization, the model is trained on the training set and is tested on the corresponding test set. RFE and SLCCA do not have any parameters. The number of iterations in RSLCCA is set to 19 (as proposed by authors in their code) for all experiments.

To compare the performances of the evaluated methods, the Mean Square Error (MSE) and Squared Correlation Coefficient (SCC) are employed. MSE and SCC are defined as:

$$MSE = \frac{\sum_{n=1}^{N}(\hat{y}_n - y_n)^2}{N} \tag{31}$$

$$SCC = \frac{(N\sum_{n=1}^{N}\hat{y}_n y_n - \sum_{n=1}^{N}\hat{y}_n \sum_{n=1}^{N} y_n)^2}{(N\sum_{n=1}^{N}\hat{y}_n^2 - \sum_{n=1}^{N}\hat{y}_n \sum_{n=1}^{N}\hat{y}_n)(N\sum_{n=1}^{N} y_n^2 - \sum_{n=1}^{N} y_n \sum_{n=1}^{N} y_n)} \tag{32}$$

where N is the number of test or validation sets, y_n and $\hat{y}$ are target value and the predicted value by the trained model, respectively.

5.2. Synthetic datasets

In this section, we evaluate proposed algorithm of VRSFM on the following synthetic regression problems:

1. Additive function:
   $$y = 0.1\exp(4x_{(1)}) + 4/(1 + \exp(-20(x_{(2)} - 0.5))) + 3x_{(3)} + 2x_{(4)} + x_{(5)} + \delta.$$
2. 2D-sin function
   $$y = \sin(x_{(1)}) + 0.1x_{(2)} + \delta.$$

where for additive function $x_{(i)}, i = 1 \ldots 5$ is uniformly distributed within the range [0, 1] and for 2D-Sin function $x_{(i)}, i = 1, 2$ is uniformly distributed within the range $[-10, 10]$. For both functions, δ is generated from a zero-mean Gaussian noise with unity standard deviation. The additive function has been used frequently for evaluating feature selection methods in the literature [9,22]. However, the use of 2D-Sin function is new. Additive and 2D-Sin functions are concatenated respectively with five and eight irrelevant features which are selected from a zero-mean Gaussian distribution with unity standard deviation. The performances of all the algorithms are tested on different sizes of training set (30, 50, 70, 100, 200). For all training sets, 100 samples are used as validation set while 2000 samples are employed for test set. We generate these three sets for 30 times and evaluate all the algorithms on the same 30 realizations of the synthetic data. For each realization, all parameters are obtained separately using validation set and in reporting, the time spent on searching for parameters is also considered as a part of training time.

Figs. 2–5 compares the MSE, SSC of the proposed method with other competing methods for different training set sizes on the Additive and 2D-Sin datasets. As depicted, VRSFM demonstrates the

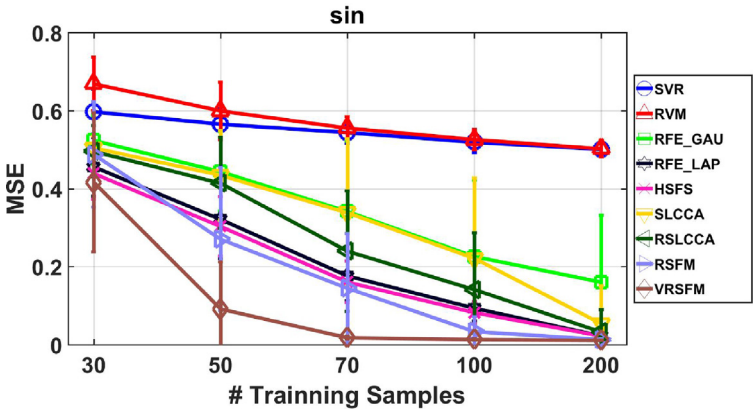

Fig. 2. MSE for 2D-Sin datasets.

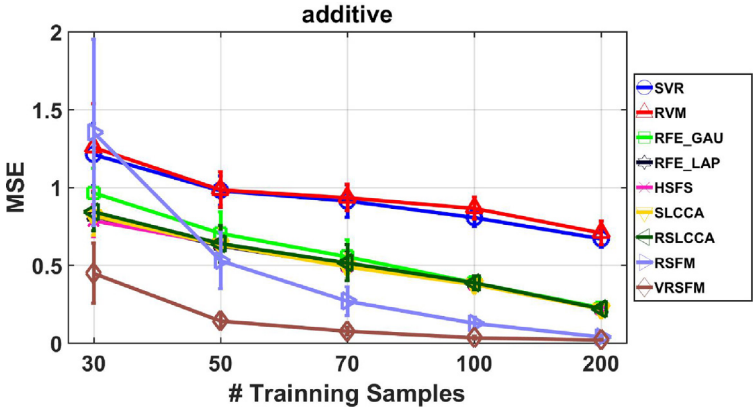

Fig. 3. MSE for Additive dataset.

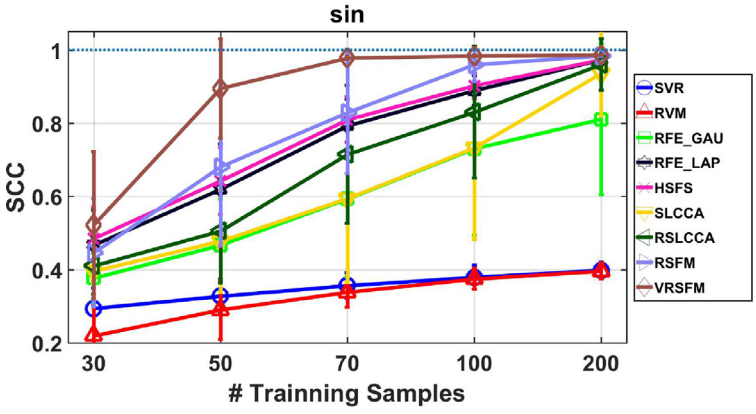

Fig. 4. SCC for 2D-Sin dataset.

best performance in terms of MSE, SCC on the Additive and 2D-Sin datasets. Particularly, when the size of training set is small, the VRSFM accuracy is much better than RSFM. When the increase of training set size, the accuracy of both methods converge. As it was predictable, the RVM and SVR results are much worse than other methods, because they cannot eliminate noisy features. To prove that VRSFM performance is significantly better than other methods, the t-test is performed on SCC and MSE scores of 30 realizations for each dataset size. In Additive dataset, the p-Value for SCC and MSE of all methods compared to VRSFM was far less than 0.001. Thus the performance of VRSFM was significantly better in all dataset sizes compared with all other methods. For the Sin dataset, the p-Values for MSE and SCC comparing with other methods are reported in Table 1. In this test, we ignore the RVM and SVR methods, because these methods do not perform any feature selection and their results are significantly worse than other methods. As shown in Table 1 the p-Values for most of methods and dataset sizes are less than 0.05 and therefore the superiority of VRSFM in comparison with other methods is statistically significant.

5.3. Real datasets

In this section we evaluate the VRSFM performance on six real benchmark datasets. *Abalone*,[1] *Boston Housing*,[1] *Bodyfat*,[2]

[1] Available at https://archive.ics.uci.edu/ml/datasets/.
[2] Available at http://calcnet.mth.cmich.edu/org/spss/Prj_body_fat_data.htm.

Please cite this article as: A. Mirzaei et al., Variational Relevant Sample-Feature Machine: A fully Bayesian approach for embedded feature selection, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.02.057

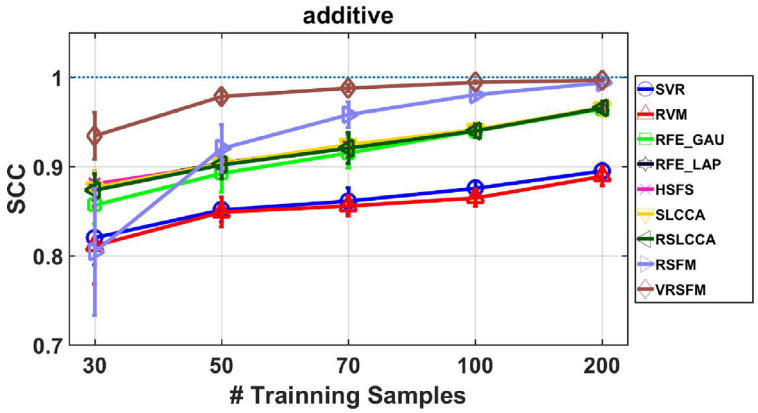

Fig. 5. SCC for Additive dataset.

Table 1
p-Values in *t*-test comparisons of VRSFM with other methods on Sin Dataset. * denotes the *p*-Value is less than 0.05 and NS (Not Significant) indicates the *p*-Value is more than 0.05 and the result is not significantly better. For simplicity the *p*-Value is not reported when its value is far less than 0.001.

Methods	SCC				
	30	50	70	100	200
RFE_GAU	*	*	*	*	*
RFE_LAP	NS(p=0.08)	*	*	*	*
HSFS	NS(p=0.25)	*	*	*	*
SLCCA	*	*	*	*	*($p = 0.02$)
RSLCCA	*	*	*	*	*($p = 0.05$)
RSFM	*	*	*	*($p = 0.01$)	*
	MSE				
RFE_GAU	*($p = 0.002$)	*	*	*	*
RFE_LAP	NS(p=0.17)	*	*	*	*
HSFS	NS(p=0.41)	*	*	*	*
SLCCA	*($p = 0.009$)	*	*	*	*($p = 0.023$)
RSLCCA	*($p = 0.019$)	*	*	*	*($p = 0.046$)
RSFM	*($p = 0.001$)	*	*	*($p = 0.006$)	*

Table 2
Detailed number of samples for learning, validation and test on real datasets.

Dataset	N_{tr}	N_{val}	N_{tst}	# features
Abalone	6027	6027	2923	8
Boston Housing	228	228	50	13
Bodyfat	113	113	26	14
Cpusmall	410	410	7372	12
Mpg	127	127	40	7
Triazines	84	84	18	60

Cpusmall,[1] *Mpg*[1] and *Triazines*.[3] Similar to synthetic datasets, all available samples in each dataset are partitioned to three sets: training, validation and test. The detailed number of samples in each set is illustrated in Table 2. We randomly select 30 realizations for each dataset and save them to evaluate the performance of all the algorithms on the same data. Table 3 compares the performance of the six algorithms in terms of MSE, SSC, number of relevant samples (RSs), number of relevant features (RFs) and model complexity. Model complexity is defined as: $ModelComplexity = \#RelevantSamples \times \#RelevantFeatures$. To determine that the results are significantly different, one-way ANOVA test is used. In one-way ANOVA test the null hypothesis is that mean result of different methods are equal to each other. *p*-Values of ANOVA test are reported in Table 3. As shown, on two datasets of Bodyfat and Cpusmall, VRSFM outperforms the other methods in terms of MSE and SSC measures. In all datasets, VRSFM has a better performance in terms of MSE and SCC compared to its non-fully Bayesian counterpart (RSFM). It is worth to mention that on datasets like Abalone and Mpg which do not include irrelevant features, the non-feature selection method of SVR shows a superior performance as can be seen in Table 3. Interestingly, on the two datasets of Bodyfat and Cpusmall which VRSFM beats its competing methods, according to the ANOVA test the results are statistically significant.

5.4. Classification problems

We evaluate the proposed VRSFM in the classification tasks on both synthetic and real datasets.

5.4.1. Synthetic dataset

To show the superiority of VRSFM compared to RSFM, we use synthetic Ripley dataset.[4] In this dataset, there are 250 samples with 2 features per sample, generated from a mixture of Gaussian distributions. To show the ability of VRSFM in pruning the irrelevant features, we concatenate each sample with 8 noisy features which are generated from a zero-mean and unity variance Gaussian distribution. The experiment is conducted with different sizes of training data $N_{tr} = [30, 50, 70, 100, 200]$. For each training size, the remaining samples are taken as test data. VRSFM, RSFM and RVM are all evaluated on 30 realizations. Fig. 6 shows the accuracy and Figs. 7 and 8 shows the sparsity of these methods. As shown in Fig. 6, for all training sizes the VRSFM outperforms the other two methods (and especially the RSFM). Moreover, the variance of accuracies are less than those for the RSFM, implying that the VRSFM model is much more reliable than the RSFM model. As expected, RVM could not handle noisy features and the classification accuracy of this method is much lower than those of RSFM and VRSFM. As depicted in Fig. 7, the VRSFM can detect the two relevance features successfully in all training sizes. On the sparsity aspect as depicted in (Figs. 7 and 8), the VRSFM is also sparser than RVM and RSFM in both feature and sample domains.

5.4.2. Real dataset

In addition to synthetic dataset, we used three well-known benchmark real datasets to evaluate the proposed method of VRSFM. These three datasets are the *Pima Indians diabetes*,[5] the

[3] Available at http://www.dcc.fc.up.pt/ltorgo/Regression/DataSets.html.

[4] Available at http://www.stats.ox.ac.uk/pub/PRNN/.

[5] Available at https://www.stats.ox.ac.uk/pub/PRNN/.

Table 3
Results of trained model on test set of real datasets.

Dataset	*Method*	*MSE*	*SCC*	*RSs*	*RFs*	*Comp.*
Abalone	*SVR*	**4.700 ± 0.191**	**0.56 ± 0.01**	285.0 ± 145.1	14.0 ± 0.0	3990
	RVM	4.834 ± 0.476	0.54 ± 0.03	13.6 ± 6.8	8.0 ± 0.0	109
	RFE_Gau	4.703 ± 0.190	0.55 ± 0.01	284.3 ± 145.1	7.8 ± 0.5	2218
	RFE_Lap	4.783 ± 0.222	0.55 ± 0.02	319.0 ± 160.8	5.8 ± 2.1	1840
	SLCCA	4.774 ± 0.237	0.55 ± 0.02	296.5 ± 153.2	5.8 ± 2.1	1730
	RSLCCA	4.778 ± 0.237	0.55 ± 0.02	300.7 ± 154.8	6.1 ± 2.1	1844
	RSFM	4.786 ± 0.176	0.54 ± 0.01	**4.0 ± 1.4**	7.2 ± 0.9	**29**
	VRSFM	4.769 ± 0.177	0.55 ± 0.01	5.5 ± 1.8	6.1 ± 0.8	34
	ANOVA	NS(p=0.50)	NS(p=0.07)	–	–	–
B.Housing	*SVR*	16.04 ± 8.50	0.80 ± 0.08	158.6 ± 50.5	14.0 ± 0.0	2220
	RVM	15.04 ± 9.17	0.82 ± 0.08	27.2 ± 10.0	13.0 ± 0.0	353
	RFE_Gau	15.27 ± 8.40	0.82 ± 0.08	177.9 ± 47.2	9.7 ± 2.5	1725
	RFE_Lap	14.22 ± 7.61	0.83 ± 0.07	174.6 ± 41.9	9.6 ± 2.0	1682
	SLCCA	14.85 ± 8.67	0.82 ± 0.08	163.1 ± 51.7	9.3 ± 2.4	1517
	RSLCCA	**14.00 ± 6.98**	**0.83 ± 0.06**	186.2 ± 45.8	9.5 ± 1.6	1775
	RSFM	19.36 ± 10.93	0.77 ± 0.09	**5.3 ± 2.7**	8.5 ± 1.0	**46**
	VRSFM	17.74 ± 9.35	0.79 ± 0.08	12.3 ± 7.2	9.2 ± 1.2	113
	ANOVA	NS(p=0.23)	NS(p=0.05)	–	–	–
Bodyfat	*SVR*	0.00023 ± 0.00009	0.64 ± 0.17	14.0 ± 0.0	14.0 ± 0.0	196
	RVM	0.00003 ± 0.00003	0.93 ± 0.08	39.9 ± 31.5	14.0 ± 0.0	559
	RFE_Gau	0.00023 ± 0.00009	0.61 ± 0.19	12.1 ± 2.8	12.1 ± 2.8	146
	RFE_Lap	0.00015 ± 0.00006	0.95 ± 0.05	**2.3 ± 0.5**	1.0 ± 0.2	**2**
	SLCCA	0.00015 ± 0.00006	0.95 ± 0.06	2.5 ± 1.1	1.3 ± 1.3	3
	RSLCCA	0.00016 ± 0.00007	0.92 ± 0.11	2.4 ± 0.8	1.2 ± 0.8	3
	RSFM	0.00002 ± 0.00003	0.96 ± 0.07	4.6 ± 2.1	3.3 ± 1.0	15
	VRSFM	**0.00002 ± 0.00002**	**0.97 ± 0.05**	8.7 ± 3.7	2.3 ± 0.6	20
	ANOVA	*(p < <0.001)	*(p < <0.001)	–	–	–
Cpusmall	*SVR*	35.53 ± 4.99	0.90 ± 0.01	236.2 ± 67.2	14.0 ± 0.0	3307
	RVM	24.81 ± 7.00	0.93 ± 0.02	34.9 ± 6.8	12.0 ± 0.0	419
	RFE_Gau	28.14 ± 6.19	0.92 ± 0.02	343.8 ± 73.4	6.4 ± 2.6	2189
	RFE_Lap	18.76 ± 3.63	0.95 ± 0.01	315.5 ± 82.7	3.3 ± 0.6	1031
	SLCCA	24.62 ± 3.57	0.93 ± 0.01	301.4 ± 77.4	5.2 ± 2.5	1557
	RSLCCA	22.50 ± 4.17	0.94 ± 0.01	340.6 ± 73.6	4.0 ± 1.6	1363
	RSFM	13.66 ± 1.06	0.96 ± 0.00	**9.7 ± 3.1**	10.5 ± 1.2	**102**
	VRSFM	**13.24 ± 0.51**	**0.96 ± 0.00**	14.1 ± 4.1	9.9 ± 0.9	140
	ANOVA	*(p < <0.001)	*(p < <0.001)	–	–	-
Mpg	*SVR*	**8.14 ± 3.36**	**0.88 ± 0.04**	120.0 ± 40.2	14.0 ± 0.0	1680
	RVM	8.29 ± 3.14	0.87 ± 0.04	12.5 ± 5.3	7.0 ± 0.0	87
	RFE_Gau	8.23 ± 3.44	0.87 ± 0.04	113.4 ± 41.4	5.6 ± 1.4	639
	RFE_Lap	8.55 ± 3.51	0.87 ± 0.04	123.5 ± 39.1	5.6 ± 1.5	688
	SLCCA	8.46 ± 3.77	0.87 ± 0.05	111.7 ± 40.6	5.7 ± 1.7	633
	RSLCCA	8.41 ± 3.68	0.87 ± 0.04	129.0 ± 38.1	5.3 ± 1.5	684
	RSFM	8.91 ± 3.17	0.86 ± 0.04	**4.1 ± 0.9**	4.9 ± 0.8	**20**
	VRSFM	8.88 ± 3.36	0.86 ± 0.04	4.5 ± 1.9	5.0 ± 0.7	23
	ANOVA	NS(p=0.98)	NS(p=0.92)	–	–	–
Triazines	*SVR*	0.02281 ± 0.00971	0.13 ± 0.13	62.6 ± 3.3	14.0 ± 0.0	877
	RVM	0.02513 ± 0.01154	0.12 ± 0.11	31.2 ± 25.9	60.0 ± 0.0	1874
	RFE_Gau	0.02266 ± 0.00929	0.15 ± 0.16	58.9 ± 6.7	52.7 ± 6.5	3106
	RFE_Lap	**0.01997 ± 0.01082**	**0.29 ± 0.21**	56.4 ± 9.6	8.5 ± 3.3	478
	SLCCA	0.02188 ± 0.01076	0.22 ± 0.21	53.3 ± 14.1	17.0 ± 8.4	906
	RSLCCA	0.02470 ± 0.01094	0.15 ± 0.17	52.0 ± 13.4	21.1 ± 12.7	1098
	RSFM	0.02147 ± 0.00843	0.18 ± 0.16	**4.3 ± 1.1**	6.0 ± 1.4	**26**
	VRSFM	0.02160 ± 0.00939	0.19 ± 0.15	11.4 ± 6.5	7.6 ± 3.9	87
	ANOVA	NS(p=0.54)	*(p=0.0015)	–	–	–

Leptograpsus crabs,[6] and the *Wisconsin breast cancer (WBC)*.[7] The goal in *Pima* dataset is to decide whether a person, based on seven measured features, has diabetes or not. This dataset has 200 predefined training samples and 332 test samples. In *Crabs* case, the problem is to determine sex of crabs based on five geometric properties of crabs. There are 50 samples available for crabs of each sex and color, making a total of 200 labeled samples. We use 80 samples (20 samples from each sex and color) as the training set and other 120 samples as the test set. The training and test sets are randomly generated for 30 times and the average accuracy is reported. In *WBC*, the goal is to decide whether the suspected cancer case is benign or malignant based on 30 numerical features. There are totally 569 samples. We train the model using the first predefined 300 samples and test the model on the remaining 269 samples. In all datasets, the parameter of the VRSFM is selected with a 5-fold cross validation on the training dataset. The results of other methods are reported directly from their corresponding papers.

As depicted in Table 4, the VRSFM outperforms other methods at two datasets of Crabs and WBC and its error is near to the best error in Pima dataset.

[6] Available at https://www.stats.ox.ac.uk/pub/PRNN/.
[7] Available at http://archive.ics.uci.edu/ml/.

Please cite this article as: A. Mirzaei et al., Variational Relevant Sample-Feature Machine: A fully Bayesian approach for embedded feature selection, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.02.057

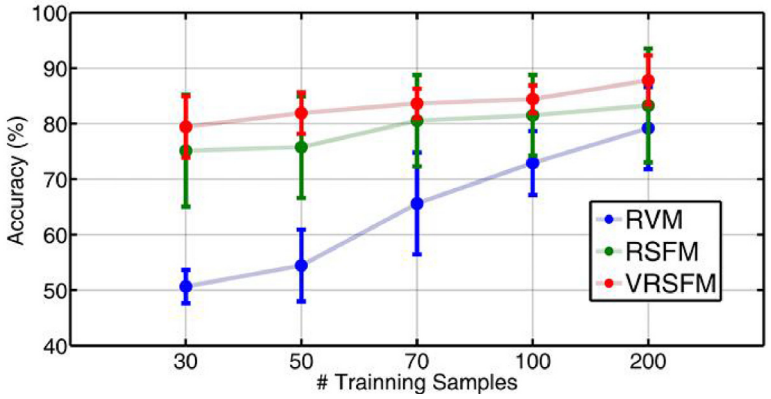

Fig. 6. Classification accuracy for Ripley dataset.

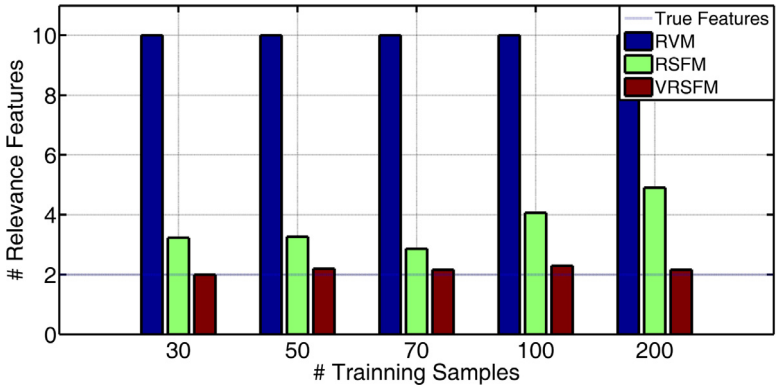

Fig. 7. Average number of relevance features for Ripley dataset.

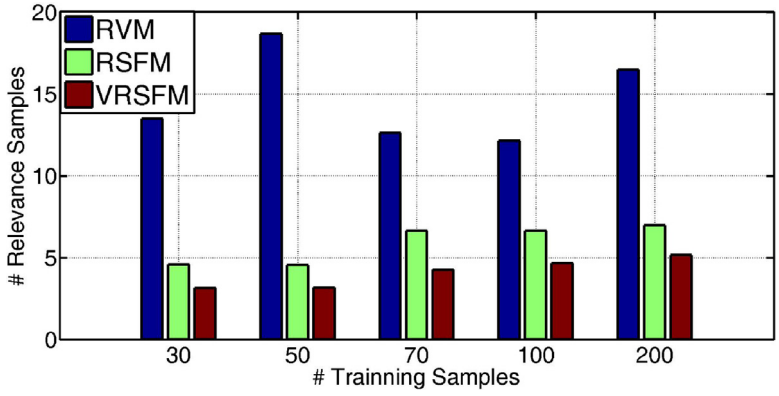

Fig. 8. Average number of relevance samples for Ripley dataset.

Table 4
Error of algorithms on three benchmark real datasets based on average number of errors.

Method	Pima	Crabs	WBC
Linear discriminant [23]	67	3	19
Neural network [24]	75	3	N/A
Gaussian process [24]	67	3	8
SVM [23]	64	4	9
Logistic regression [24]	66	4	N/A
RVM [15]	65	0	9
Sparse probit regression [25]	62	0	9
JCFO [13]	64	0	8
IRSFM [12]	65	1	6
RSFM [11]	66	0	9
VRSFM	63	0	7

6. Conclusion

In this paper we proposed a fully Bayesian approach for embedded feature selection applied to regression and classification tasks. Defining prior Gaussian distributions on the model parameters and then corresponding hyper-parameters, we employed a Variational Bayesian approximation method to find the posterior distributions of the parameters and hyper-parameters. We presented a detailed derivation of the update equations for the proposed method called VRSFM. The proposed algorithm is applied to feature selection for regression and classification tasks, leading to performance improvements, in terms of MSE and SSC measures, in regression tasks as well as accuracy improvements for classification problems by eliminating noisy and irrelevant features. Our

Please cite this article as: A. Mirzaei et al., Variational Relevant Sample-Feature Machine: A fully Bayesian approach for embedded feature selection, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.02.057

experiments on synthetic as well as real-world datasets (especially on the datasets with many noisy and irrelevant features) demonstrated that the proposed algorithm improved the accuracy. In addition to improved accuracies, the complexity of the VRSFM model is much lower than the existing wrapper feature selection methods like RFE methods. This is because in the VRSFM model, the irrelevant features as well as the irrelevant samples are pruned during training. Moreover, there is only one parameter (the kernel width) to be optimized during the validation phase.

References

[1] S. Maldonado, R. Weber, J. Basak, Simultaneous feature selection and classification using kernel-penalized support vector machines, Inf. Sci. 181 (1) (2011) 115–128.
[2] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (Mar.) (2003) 1157–1182.
[3] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (1) (2014) 16–28.
[4] L. Yu, H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, in: Proceedings of International Conference on Machine Learning (ICML), 3, 2003, pp. 856–863.
[5] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.
[6] P. Radivojac, Z. Obradovic, A.K. Dunker, S. Vucetic, Feature selection filters based on the permutation test, in: Proceedings of European Conference on Machine Learning (ECML 2004), Springer, 2004, pp. 334–346.
[7] M. Masaeli, J.G. Dy, G.M. Fung, From transformation-based dimensionality reduction to feature selection, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 751–758.
[8] A. Sharma, S. Imoto, S. Miyano, A top-r feature selection algorithm for microarray gene expression data, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (3) (2012) 754–764.
[9] J.-B. Yang, C.-J. Ong, Feature selection using probabilistic prediction of support vector regression, IEEE Trans. Neural Netw. 22 (6) (2011) 954–962.
[10] A. Rakotomamonjy, Analysis of SVM regression bounds for variable ranking, Neurocomputing 70 (7) (2007) 1489–1501.
[11] Y. Mohsenzadeh, H. Sheikhzadeh, A. Reza, N. Bathaee, M. Kalayeh, The relevance sample-feature machine: a sparse Bayesian learning approach to joint feature-sample selection, IEEE Trans. Cybern. 43 (6) (2013) 2241–2254, doi:10.1109/TCYB.2013.2260736.
[12] Y. Mohsenzadeh, H. Sheikhzadeh, S. Nazari, Incremental relevance sample-feature machine: a fast marginal likelihood maximization approach for joint feature selection and classification, Pattern Recognit. 60 (2016) 835–848.
[13] B. Krishnapuram, A. Harternink, L. Carin, M.A. Figueiredo, A Bayesian approach to joint feature selection and classifier design, IEEE Trans. Pattern Anal. Mach. Intell. 26 (9) (2004) 1105–1111.
[14] À. Lapedriza, S. Seguí, D. Masip, J. Vitrià, A sparse Bayesian approach for joint feature selection and classifier learning, Pattern Anal. Appl. 11 (3–4) (2008) 299–308.
[15] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, J. Mach. Learn. Res. 1 (2001) 211–244.
[16] C.M. Bishop, et al., Pattern Recognition and Machine Learning, 1, Springer, New York, 2006.
[17] C.M. Bishop, M.E. Tipping, Variational relevance vector machines, in: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 2000, pp. 46–53.
[18] T.S. Jaakkola, M.I. Jordan, Bayesian parameter estimation via variational methods, Stat. Comput. 10 (1) (2000) 25–37.
[19] A. Smola, V. Vapnik, Support vector regression machines, Adv. Neural Inf. Process. Syst. 9 (1997) 155–161.
[20] H. Kaya, F. Eyben, A.A. Salah, B. Schuller, CCA based feature selection with application to continuous depression recognition from acoustic speech features, in: Proceedings of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2014, pp. 3729–3733.
[21] H. Kaya, T. Özkaptan, A.A. Salah, F. Gürgen, Random discriminative projection based feature selection with application to conflict recognition, IEEE Signal Process. Lett. 22 (6) (2015) 671–675.
[22] X. Peng, D. Xu, A local information-based feature-selection algorithm for data regression, Pattern Recognit. 46 (9) (2013) 2519–2530.
[23] M. Seeger, Bayesian model selection for support vector machines, Gaussian processes and other kernel classifiers, in: Proceedings of the 13th Annual Conference on Neural Information Processing Systems, in: EPFL-CONF-161324, 2000, pp. 603–609.
[24] C.K. Williams, D. Barber, Bayesian classification with Gaussian processes, IEEE Trans. Pattern Anal. Mach. Intell. 20 (12) (1998) 1342–1351.
[25] M.A. Figueiredo, Adaptive sparseness for supervised learning, IEEE Trans. Pattern Anal. Mach. Intell. 25 (9) (2003) 1150–1159.

Ali Mirzaei received his B.S. and M.S. degrees in electrical engineering from Amirkabir University of Technology, Tehran, Iran, in 2013 and 2015, respectively. He also received a second B.S. degree in computer science from the same university in 2015. He is currently a senior research engineer in Faraadid Co., Tehran, Iran, working on machine learning algorithms and deep neural networks in computer vision tasks, such as object detection, object tracking, image classification and scene classification. His research interests include computer vision, machine learning, pattern recognition and deep learning.

Yalda Mohsenzadeh received her Ph.D.degree in Electrical Engineering from Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran in 2014. From 2014 to 2016 she was a Postdoctoral Researcher in the Visuomotor Neuroscience Lab, Center for Vision Research at York University, Toronto, Canada where she worked on computational modeling of human brain visual system and transsaccadic perception. She is currently a postdoctoral associate at McGovern Institute for Brain Research at Massachusetts Institute of Technology (MIT), Cambridge, MA, USA working on spatiotemporal dynamics of human visual perception and memory using MEG and fMRI. Her research interests include machine learning, pattern recognition, Bayesian learning, computational neuroscience, and neural basis of human perception and memory.

Hamid Sheikhzadeh 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 in 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 an associate professor in the Electrical Engineering Department of Amirkabir University of Technology. His 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.

Please cite this article as: A. Mirzaei et al., Variational Relevant Sample-Feature Machine: A fully Bayesian approach for embedded feature selection, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.02.057