Nonlinear Feature Selection by Relevance Feature Vector Machine*

Haibin Cheng[1], Haifeng Chen[2], Guofei Jiang[2], and Kenji Yoshihira[2]

[1] CSE Department, Michigan State University
East Lansing, MI 48824
chenghai@msu.edu
[2] NEC Laboratories America, Inc.
4 Independence Way, Princeton, NJ 08540
{haifeng,gfj,kenji}@nec-labs.com

Abstract. Support vector machine (SVM) has received much attention in feature selection recently because of its ability to incorporate kernels to discover nonlinear dependencies between features. However it is known that the number of support vectors required in SVM typically grows linearly with the size of the training data set. Such a limitation of SVM becomes more critical when we need to select a small subset of relevant features from a very large number of candidates. To solve this issue, this paper proposes a novel algorithm, called the 'relevance feature vector machine'(RFVM), for nonlinear feature selection. The RFVM algorithm utilizes a highly sparse learning algorithm, the relevance vector machine (RVM), and incorporates kernels to extract important features with both linear and nonlinear relationships. As a result, our proposed approach can reduce many false alarms, e.g. including irrelevant features, while still maintain good selection performance. We compare the performances between RFVM and other state of the art nonlinear feature selection algorithms in our experiments. The results confirm our conclusions.

1 Introduction

Feature selection is to identify a small subset of features that are most relevant to the response variable. It plays an important role in many data mining applications where the number of features is huge such as text processing of web documents, gene expression array analysis, and so on. First of all, the selection of a small feature subset will significantly reduce the computation cost in model building, e.g. the redundant independent variables will be filtered by feature selection to obtain a simple regression model. Secondly, the selected features usually characterize the data better and hence help us to better understand the data. For instance, in the study of genome in bioinformatics, the best feature (gene) subset can reveal the mechanisms of different diseases[6]. Finally, by eliminating the irrelevant features, feature selection can avoid the problem of "curse

* The work was performed when the first author worked as a summer intern at NEC Laboratories America, Inc.

P. Perner (Ed.): MLDM 2007, LNAI 4571, pp. 144–159, 2007.

of dimensionality" in case when the number of data examples is small in the high-dimensional feature space [2].

The common approach to feature selection uses greedy local heuristic search, which incrementally adds and/or deletes features to obtain a subset of relevant features with respect to the response[21]. While those methods search in the combinatorial space of feature subsets, regularization or shrinkage methods [20][18] trim the feature space by constraining the magnitude of parameters. For example, Tibshirani [18] proposed the Lasso regression technique which relies on the polyhedral structure of L_1 norm regularization to force a subset of parameter values to be exactly zero at the optimum. However, both the combinatorial search based methods and regularization based methods assume the linear dependencies between features and the response, and can not handle their nonlinear relationships.

Due to the sparse property of support vector machine (SVM), recent work [3][9] reformulated the feature selection problem into SVM based framework by switching the roles of features and data examples. The support vectors after optimization are then regarded as the relevant features. By doing so, we can apply nonlinear kernels on feature vectors to capture the nonlinear relationships between the features and the response variable. In this paper we utilize such promising characteristic of SVM to accomplish nonlinear feature selection. However, we also notice that in the past few years the data generated in a variety of applications tend to have thousands of features. For instance, in the gene selection problem, the number of features, the gene expression coefficients corresponding to the abundance of mRNA, in the raw data ranges from 6000 to 60000 [19]. This large number of features presents a significant challenge to the SVM based feature selection because it has been shown [7] that the number of support vectors required in SVM typically grows linearly with the size of the training data set. When the number of features is large, the standard SVM based feature selection may produce many false alarms, e.g. include irrelevant features in the final results.

To effectively select relevant features from vast amount of attributes, this paper proposes to use the "Relevance Vector Machine" (RVM) for feature selection. Relevance vector machine is a Bayesian treatment of SVM with the same decision function [1]. It produces highly sparse solutions by introducing some prior probability distribution to constrain the model weights governed by a set of hyper-parameters. As a consequence, the selected features by RVM are much fewer than those learned by SVM while maintaining comparable selection performance. In this paper we incorporate a nonlinear feature kernel into the relevance vector machine to achieve nonlinear feature selection from large number of features. Experimental results show that our proposed algorithm, which we call the "Relevance Feature Vector Machine" (RFVM), can discover nonlinear relevant features with good detection rate but low rate of false alarms. Furthermore, compared with the SVM based feature selection methods [3][9], our proposed RFVM algorithm offers other compelling benefits. For instance, the parameters in RFVM are automatically learned by the maximum likelihood estimation rather than the time-consuming cross validation procedure as does in the SVM based methods.

The rest of the paper is organized as follows. In Section 2, we will summarize the related work of nonlinear feature selection using SVM. In Section 3, we extend the relevance vector machine for the task of nonlinear feature selection. The experimental results and conclusions are presented in Section 4 and Section 5 respectively.

2 Preliminaries

Given a data set $D = \left[X_{n\times m}, \boldsymbol{y}_{n\times 1}\right]$, where $X_{n\times m}$ represents the n input examples with m features and $\boldsymbol{y}_{n\times 1}$ represents the responses, we first describe definitions of *feature space* and *example space* with respect to the data. In the feature space, each dimension is related to one specific feature, the data set is regarded as a group of data examples $D = [(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_n, y_n)]^T$, where $\boldsymbol{x}_i$s are the rows of X, $X = [\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \cdots, \boldsymbol{x}_n^T]^T$. The sparse methods such as SVM in the feature space try to learn a sparse example weight vector $\overline{\boldsymbol{\alpha}} = [\alpha_1, \alpha_2, \cdots, \alpha_n]$ associated with the n data examples. The examples with nonzero values α_i are regarded as support vectors, which are illustrated as solid circles in Figure 1(a). Alternatively, each dimension in the example space is related to each data sample $\boldsymbol{x}_i$, and the data is denoted as a collection of features $X = [\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_m]$ and response $\boldsymbol{y}$. The sparse solution in the example space is then related to a weight vector $\boldsymbol{w} = [w_1, w_2, \cdots, w_m]^T$ associated with m features. Only those features with nonzero elements in $\boldsymbol{w}$ are regarded as relevant ones or "support features". If we use SVM to obtain the sparse solution, those relevant features

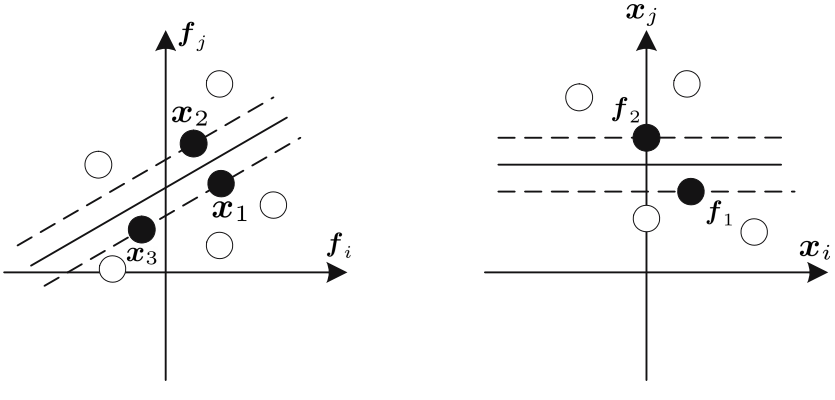


Fig. 1. (a) The feature space where each dimension is related to one feature ($\boldsymbol{f}$) in the data. SVM learns the sparse solution (denoted as black points) of weight vector $\overline{\boldsymbol{\alpha}}$ associated with data examples $\boldsymbol{x}_i$. (b) The example space in which each dimension is a data example $\boldsymbol{x}_i$. The sparse solution (denoted as black points) of weight vector $\boldsymbol{w}$ is associated with related features ($\boldsymbol{f}$).

are derived from the support features as shown in Figure 1(b). In this section, we first describe feature selection in the SVM framework. Then we will present nonlinear feature selection solutions.

2.1 Feature Selection by SVM

Support Vector Machine [13] is a very popular machine learning technique for the task of classification and regression. The standard SVM-regression [14] aims to find a predictive function $f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{w} \rangle + b$ that has at most ϵ deviation from the actual value y and is as flat as possible, where $\boldsymbol{w}$ is the feature weight vector as described before and b is the offset of function f. If the solution can be further relaxed by allowing certain degree of error, the optimization problem of SVM-regression can be formulated as

$$\begin{aligned} &min\ \frac{1}{2}||\boldsymbol{w}||^2 + C\mathbf{1}^T(\boldsymbol{\xi}^+ + \boldsymbol{\xi}^-) \\ &sub.\ \begin{cases} \boldsymbol{y} - \langle X, \boldsymbol{w} \rangle - b\mathbf{1} \leq \epsilon\mathbf{1} + \boldsymbol{\xi}^+ \\ \langle X, \boldsymbol{w} \rangle + b\mathbf{1} - \boldsymbol{y} \leq \epsilon\mathbf{1} + \boldsymbol{\xi}^- \\ \boldsymbol{\xi}^+, \boldsymbol{\xi}^- \geq 0 \end{cases} \end{aligned} \tag{1}$$

where $\boldsymbol{\xi}^+$ and $\boldsymbol{\xi}^-$ represent the errors, C measures the trade-off between error relaxation and flatness of function, and $\mathbf{1}$ denotes the vector whose elements are all 1s. Instead of solving this optimization problem directly, it is usually much easier to solve its dual form [14] by SMO algorithm. The dual problem of the SVM-regression can be derived from Lagrange optimization with KKT conditions and Lagrange multipliers $\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-$:

$$\begin{aligned} &min\ \frac{1}{2}(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-)^T\langle X, X^T\rangle(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-) \\ &\quad -\boldsymbol{y}^T(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-) + \epsilon\mathbf{1}^T(\boldsymbol{\alpha}^+ + \boldsymbol{\alpha}^-) \\ &sub.\ \mathbf{1}^T(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-) = 0, 0 \leq \boldsymbol{\alpha}^+ \leq C\mathbf{1}, 0 \leq \boldsymbol{\alpha}^- \leq C\mathbf{1} \end{aligned} \tag{2}$$

The dual form also provides an easy way to model nonlinear dependencies by incorporating nonlinear kernels. That is, a kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ defined over the examples $\boldsymbol{x}_i, \boldsymbol{x}_j$ is used to replace the dot product $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ in equation (2). The term $\epsilon\mathbf{1}^T(\boldsymbol{\alpha}^+ + \boldsymbol{\alpha}^-)$ in (2) works as the shrinkage factor and leads to the sparse solution of the example weight vector $\overline{\boldsymbol{\alpha}} = (\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-)$, which is associated with data examples in the feature space.

While the SVM algorithm is frequently used in the *feature space* to achieve sparse solution $\overline{\boldsymbol{\alpha}}$ for classification and regression tasks, the paper [3] employed SVM in the *example space* to learn a sparse solution of feature weight vector $\boldsymbol{w}$ for the purpose of feature selection by switching the roles of features and data examples. After data normalization such that $X^T\mathbf{1} = 0$ and thus $X^Tb\mathbf{1} = 0$, the SVM based feature selection described in [3] can be formulated as the following optimization problem.

$$min\ \frac{1}{2}||X\boldsymbol{w}||^2 + C\mathbf{1}^T(\boldsymbol{\xi}^+ + \boldsymbol{\xi}^-) \tag{3}$$

$$sub. \begin{cases} \langle X^T, \boldsymbol{y} \rangle - \langle X^T, X \rangle \boldsymbol{w} \leq \epsilon \mathbf{1} + \boldsymbol{\xi}^+ \\ \langle X^T, X \rangle \boldsymbol{w} - \langle X^T, \boldsymbol{y} \rangle \leq \epsilon \mathbf{1} + \boldsymbol{\xi}^- \\ \boldsymbol{\xi}^+, \boldsymbol{\xi}^- \geq 0 \end{cases}$$

The above equation (3) makes it easy to model nonlinear dependencies between features and response, which has also been explored in the work [9]. Similarly, the dual problem of (3) can also be obtained with Lagrange multiplies $\boldsymbol{w}^+, \boldsymbol{w}^-$ and KKT conditions

$$\begin{aligned} min\ & \frac{1}{2}(\boldsymbol{w}^+ - \boldsymbol{w}^-)^T \langle X^T, X \rangle (\boldsymbol{w}^+ - \boldsymbol{w}^-) \\ & - \langle \boldsymbol{y}^T, X \rangle (\boldsymbol{w}^+ - \boldsymbol{w}^-) + \epsilon \mathbf{1}^T (\boldsymbol{w}^+ + \boldsymbol{w}^-) \\ sub.\ & 0 \leq \boldsymbol{w}^+ \leq C\mathbf{1}, 0 \leq \boldsymbol{w}^- \leq C\mathbf{1} \end{aligned} \tag{4}$$

The intuition behind the dual optimization problem (4) is very obvious. It tries to minimize the mutual feature correlation noted as $\langle X^T, X \rangle$ and maximize the response feature correlation $\langle \boldsymbol{y}^T, X \rangle$. The parameter "C" in equation (4) controls the redundancy of the selected features. Small value of "C" reduces the importance of mutual feature correlation $\langle X^T, X \rangle$ and thus allow more redundancy. The term $\epsilon \mathbf{1}^T (\boldsymbol{w}^+ + \boldsymbol{w}^-)$ in the above dual form (4) achieves the sparseness of the feature weight vector $\boldsymbol{w} = (\boldsymbol{w}^+ - \boldsymbol{w}^-)$. After optimization, the nonzero elements in $\boldsymbol{w}$ are related to the relevant features in the example space. For the detailed explanation about the derivation of (3) and (4), please see [3].

2.2 Nonlinear Feature Selection

If we set $\epsilon = \frac{\lambda}{2}$ and ignore the error relaxation in the primal problem (3), the optimization form (3) can be rewritten in the example space using features $X = [\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_m]$ and the response $\boldsymbol{y}$

$$\begin{aligned} min\ & \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} w_i w_j \langle \boldsymbol{f}_i, \boldsymbol{f}_j \rangle \\ sub.\ & |\sum_{i=1}^{m} w_i \langle \boldsymbol{f}_j, \boldsymbol{f}_i \rangle - \langle \boldsymbol{f}_j, \boldsymbol{y} \rangle| \leq \frac{\lambda}{2}\ , \quad \forall j \end{aligned} \tag{5}$$

The optimization problem in (5) has been proved in [9] to be equivalent to the Lasso regression (6) [18] which has been widely used for linear feature selection

$$min||X\boldsymbol{w} - \boldsymbol{y}||^2 + \lambda ||\boldsymbol{w}||_1\ . \tag{6}$$

While the Lasso regression (6) is performed in the *feature space* of data set to achieve feature selection, the optimization (5) formulates the feature selection problem in the *example space*. As a consequence, we can define nonlinear kernels over the feature vectors to model nonlinear interactions between features. For the feature vectors $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ with nonlinear dependency, we assume that they can be projected to a high dimensional space by a mapping function ϕ so that

they interact linearly in the mapped space. Therefore the nonlinear dependency can be represented by introducing the feature kernel $K(\boldsymbol{f}_i, \boldsymbol{f}_j) = \phi(\boldsymbol{f}_i)^T \phi(\boldsymbol{f}_j)$. If we replace the dot product $\langle , \rangle$ in (5) with the feature kernel K, we can obtain its nonlinear version:

$$\begin{aligned} &min \ \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} w_i w_j K(\boldsymbol{f}_i, \boldsymbol{f}_j) \\ &sub. \ |\sum_{i=1}^{m} w_i K(\boldsymbol{f}_j, \boldsymbol{f}_i) - K(\boldsymbol{f}_j, \boldsymbol{y})| \leq \frac{\lambda}{2} \ , \quad \forall j \ . \end{aligned} \tag{7}$$

In the same way, we can incorporate nonlinear feature kernels into the general expression (4) and obtain

$$\begin{aligned} min \ & \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} (w_i^+ - w_i^-) K(\boldsymbol{f}_i, \boldsymbol{f}_j)(w_j^+ - w_j^-) \\ & - \sum_{i=1}^{m} K(\boldsymbol{y}, \boldsymbol{f}_i)(w_i^+ - w_i^-) + \epsilon \sum_{i=1}^{n} (w_i^+ + w_j^-) \\ sub. \ & 0 \leq w_i^+ \leq C, 0 \leq w_i^- \leq C \ , \ \forall i \end{aligned} \tag{8}$$

Both (7) and (8) can be used for nonlinear feature selection. However, they are both derived from the SVM framework and share the same weakness of standard SVM algorithm. For instance, the number of support features will grow linearly with the size of the feature set in the training data. As a result, the provided solution in the example space is not sparse enough. This will lead to a serious problem of high false alarm rate, e.g. including many irrelevant features, when the feature set is large. To solve this issue, this paper proposes a RVM based solution for nonlinear feature selection, which is called "Relevance Feature Vector Machine". RFVM achieves more sparse solution in the example space by introducing priors over the feature weights. As a result, RFVM is able to select the most relevant features as well as decrease the number of false alarms significantly. Furthermore, we will also show that RFVM can learn the hyper-parameters automatically and hence avoids the effort of cross validation to determine the trade-off parameter "C" in SVM optimization (8).

3 Relevance Feature Vector Machine

In this section, we will investigate the problem of using Relevance Vector Machine for nonlinear feature selection. We will first introduce the Bayesian framework of standard Relevance Vector Machine algorithm [1]. Then we present our Relevance Feature Vector Machine algorithm which utilizes RVM in the example space and exploits the mutual information kernel for nonlinear feature selection.

3.1 Relevance Vector Machine

The standard RVM [1] is to learn the vector $\tilde{\boldsymbol{\alpha}}_{(n+1)\times 1} = [\alpha_0, \overline{\boldsymbol{\alpha}}]$ with $\alpha_0 = b$ denoting the "offset" and $\overline{\boldsymbol{\alpha}} = [\alpha_1, \alpha_2, \cdots, \alpha_n]$ as the "relevance feature weight

vector" associated with data examples in the feature space. It assumes that the response y_i is sampled from the model $f(\boldsymbol{x}_i)$ with noise ϵ, and the model function is expressed as

$$f(\boldsymbol{x}) = \sum_{j=1}^{n} \alpha_j \langle \boldsymbol{x}, \boldsymbol{x}_j \rangle + \alpha_0 + \epsilon \tag{9}$$

where ϵ is assumed to be sampled independently from a Gaussian distribution noise with mean zero and variance σ^2. If we use kernel to model the dependencies between the examples in the feature space, we can get the $n \times (n+1)$ 'design' matrix Φ:

$$\Phi = \begin{bmatrix} 1 \; K(\boldsymbol{x}_1, \boldsymbol{x}_1) \; K(\boldsymbol{x}_1, \boldsymbol{x}_2) \cdots K(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ 1 \; K(\boldsymbol{x}_2, \boldsymbol{x}_1) \; K(\boldsymbol{x}_2, \boldsymbol{x}_2) \cdots K(\boldsymbol{x}_2, \boldsymbol{x}_n) \\ \vdots \\ 1 \; K(\boldsymbol{x}_n, \boldsymbol{x}_1) \; K(\boldsymbol{x}_n, \boldsymbol{x}_2) \cdots K(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix}$$

In order to estimate the coefficients $\alpha_0, \cdots, \alpha_n$ in equation (9) from a set of training data, the likelihood of the given data set is written as

$$p(\boldsymbol{y}|\tilde{\boldsymbol{\alpha}}, \sigma^2) = (2\pi\sigma^2)^{-\frac{n}{2}} exp\left\{ -\frac{1}{\sigma^2} ||\boldsymbol{y} - \Phi\tilde{\boldsymbol{\alpha}}||^2 \right\} \tag{10}$$

In addition, RVM defines prior probability distributions on parameters $\tilde{\boldsymbol{\alpha}}$ in order to obtain sparse solutions. Such prior distribution is expressed with $n+1$ hyper-parameters $\tilde{\boldsymbol{\beta}}_{(n+1)\times 1} = [\beta_0, \beta_1, \cdots, \beta_n]$:

$$p(\tilde{\boldsymbol{\alpha}}|\tilde{\boldsymbol{\beta}}) = \prod_{i=0}^{n} N(\alpha_i|0, \beta_i^{-1}) \tag{11}$$

The unknowns $\tilde{\boldsymbol{\alpha}}$,$\tilde{\boldsymbol{\beta}}$ and σ^2 can be estimated by maximizing the posterior distribution $p(\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}}, \sigma^2|\boldsymbol{y})$, which can be decomposed as:

$$p(\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}}, \sigma^2|\boldsymbol{y}) = p(\tilde{\boldsymbol{\alpha}}|\boldsymbol{y}, \tilde{\boldsymbol{\beta}}, \sigma^2) p(\tilde{\boldsymbol{\beta}}, \sigma^2|\boldsymbol{y}) \; . \tag{12}$$

Such decomposition allows us to use two steps to find the solution $\tilde{\boldsymbol{\alpha}}$ together with hyper-parameters $\tilde{\boldsymbol{\beta}}$ and σ^2. For details of the optimization procedure, please see [1]. Compared with SVM, RVM produces a more sparse solution $\tilde{\boldsymbol{\alpha}}$ as well as determines the hyper-parameters simultaneously.

To the best of our knowledge, current RVM algorithm is always performed in the feature space in which the relevance weight vector $\overline{\boldsymbol{\alpha}}$ in RVM is associated with data examples. This paper is the first to utilize the promising characteristics of RVM for feature selection. In the next section, we reformulate the Relevance Vector Machine in the example space and incorporate nonlinear feature kernels to learn nonlinear "relevant features".

3.2 Nonlinear Feature Selection with Relevance Feature Vector Machine

This section presents the relevance feature vector machine (RFVM) algorithm, which utilizes RVM in the example space to select relevant features. We will

also show how the kernel trick can be applied to accomplish nonlinear feature selection. Again, we assume the data $(X, \boldsymbol{y})$ is standardized. We start by rewriting the function (9) into an equivalent form by incorporating the feature weight vector $\boldsymbol{w}$

$$\boldsymbol{y} = \sum_{j=1}^{m} w_j \boldsymbol{f}_j + \boldsymbol{\epsilon} \tag{13}$$

The above formula assumes the linear dependency between features and the response. When such relationship is nonlinear, we project the features and responses into high dimensional space by a function ϕ so that the dependency in the mapped space becomes linear

$$\phi(\boldsymbol{y}) = \sum_{j=1}^{m} w_j \phi(\boldsymbol{f}_j) + \boldsymbol{\epsilon} \ . \tag{14}$$

Accordingly the likelihood function given the training data can be expressed as

$$p(\phi(\boldsymbol{y})|\boldsymbol{w}, \sigma^2) = (2\pi\sigma^2)^{-\frac{n}{2}} exp\left\{ -\frac{||\phi(\boldsymbol{y}) - \phi(X)\boldsymbol{w}||^2}{\sigma^2} \right\} \tag{15}$$

where $\phi(X) = [\phi(\boldsymbol{f}_1), \phi(\boldsymbol{f}_2), \cdots, \phi(\boldsymbol{f}_m)]$. We expand the squared error term in the above likelihood function and replace the dot product with certain feature kernel K to model the nonlinear interaction between the feature vectors and response, which results in

$$\begin{aligned} & ||\phi(\boldsymbol{y}) - \phi(X)\boldsymbol{w}||^2 \\ &= (\phi(\boldsymbol{y}) - \phi(X)\boldsymbol{w})^T (\phi(\boldsymbol{y}) - \phi(X)\boldsymbol{w}) \\ &= \phi(\boldsymbol{y})^T \phi(\boldsymbol{y}) - 2\boldsymbol{w}^T \phi(X)^T \phi(\boldsymbol{y}) + \boldsymbol{w}^T \phi(X)^T \phi(X)\boldsymbol{w} \\ &= K(\boldsymbol{y}^T, \boldsymbol{y}) - 2\boldsymbol{w}^T K(X^T, \boldsymbol{y}) + \boldsymbol{w}^T K(X^T, X)\boldsymbol{w} \end{aligned}$$

where:

$$K(X^T, \boldsymbol{y}) = \begin{bmatrix} K(\boldsymbol{y}, \boldsymbol{f}_1) \\ K(\boldsymbol{y}, \boldsymbol{f}_2) \\ \vdots \\ K(\boldsymbol{y}, \boldsymbol{f}_m) \end{bmatrix}$$

and

$$K(X^T, X) = \begin{bmatrix} K(\boldsymbol{f}_1, \boldsymbol{f}_1) & K(\boldsymbol{f}_1, \boldsymbol{f}_2) & \cdots & K(\boldsymbol{f}_1, \boldsymbol{f}_m) \\ K(\boldsymbol{f}_2, \boldsymbol{f}_1) & K(\boldsymbol{f}_2, \boldsymbol{f}_2) & \cdots & K(\boldsymbol{f}_2, \boldsymbol{f}_m) \\ \vdots & & & \\ K(\boldsymbol{f}_m, \boldsymbol{f}_1) & K(\boldsymbol{f}_m, \boldsymbol{f}_2) & \cdots & K(\boldsymbol{f}_m, \boldsymbol{f}_m) \end{bmatrix}$$

After some manipulations, the likelihood function (15) can be reformulated as

$$\begin{aligned} p(\phi(\boldsymbol{y})|\boldsymbol{w}, \sigma^2) = (2\pi\sigma^2)^{-\frac{n}{2}} exp\big\{ \big(& -K(\boldsymbol{y}^T, \boldsymbol{y}) + \\ & 2\boldsymbol{w}^T K(X^T, \boldsymbol{y}) - \boldsymbol{w}^T K(X^T, X)\boldsymbol{w} \big) / \sigma^2 \big\} \end{aligned} \tag{16}$$

Note that RFVM differs from traditional RVM in that the prior $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots, \beta_m]$ is defined over the relevance feature vector weight $\boldsymbol{w}$.

$$p(\boldsymbol{w}|\boldsymbol{\beta}) = \prod_{i=1}^{m} N(w_i|0, \beta_i^{-1}) \tag{17}$$

The sparse solution $\boldsymbol{w}$ corresponding to relevant features can be obtained by maximizing

$$p(\boldsymbol{w}, \boldsymbol{\beta}, \sigma^2|\phi(\boldsymbol{y})) = p(\boldsymbol{w}|\phi(\boldsymbol{y}), \boldsymbol{\beta}, \sigma^2)p(\boldsymbol{\beta}, \sigma^2|\phi(\boldsymbol{y})) \tag{18}$$

Similar to RVM, we use two steps to find the maximized solution. The first step is now to maximize

$$\begin{aligned} p(\boldsymbol{w}|\phi(\boldsymbol{y}), \boldsymbol{\beta}, \sigma^2) &= \frac{p(\phi(\boldsymbol{y})|\boldsymbol{w}, \sigma^2)p(\boldsymbol{w}|\boldsymbol{\beta})}{p(\phi(\boldsymbol{y})|\boldsymbol{\beta}, \sigma^2)} \\ &= (2\pi)^{-\frac{n+1}{2}}|\Sigma|^{-\frac{1}{2}} exp\left\{-\frac{1}{2}(\boldsymbol{w} - \boldsymbol{\mu})^T|\Sigma|^{-1}(\boldsymbol{w} - \boldsymbol{\mu})\right\} \end{aligned} \tag{19}$$

Given the current estimation of $\boldsymbol{\beta}$ and σ^2, the covariance Σ and mean $\boldsymbol{\mu}$ of the feature weight vector $\boldsymbol{w}$ are

$$\Sigma = (\sigma^{-2}K(X^T, X) + B)^{-1} \tag{20}$$

$$\boldsymbol{\mu} = \sigma^{-2}\Sigma K(X^T, \boldsymbol{y}) \tag{21}$$

and $B = diag(\beta_1, \cdots, \beta_n)$.

Once we get the current estimation of $\boldsymbol{w}$, the second step is to learn the hyper-parameters $\boldsymbol{\beta}$ and σ^2 by maximizing $p(\boldsymbol{\beta}, \sigma^2|\phi(\boldsymbol{y})) \propto p(\phi(\boldsymbol{y})|\boldsymbol{\beta}, \sigma^2)p(\boldsymbol{\beta})p(\sigma^2)$. Since we assume the hyper-parameters are uniformly distributed, e.g. $p(\boldsymbol{\beta})$ and $p(\sigma^2)$ are constant, it is equivalent to maximize the marginal likelihood $p(\phi(\boldsymbol{y})|\boldsymbol{\beta}, \sigma^2)$, which is computed by:

$$\begin{aligned} p(\phi(\boldsymbol{y})|\boldsymbol{\beta}, \sigma^2) &= \int p(\phi(\boldsymbol{y})|\boldsymbol{w}, \sigma^2)p(\boldsymbol{w}|\boldsymbol{\beta})d\boldsymbol{w} \\ &= (2\pi)^{-\frac{n}{2}}|\sigma^2\boldsymbol{I} + \phi(X)B^{-1}\phi(X)^T|^{-\frac{1}{2}} \\ &\quad *exp\left\{-\frac{1}{2}\boldsymbol{y}^T(\sigma^2\boldsymbol{I} + \phi(X)B^{-1}\phi(X)^T)^{-1}\boldsymbol{y}\right\} \end{aligned} \tag{22}$$

By differentiation of equation (22), we can update the hyper-parameters $\boldsymbol{\beta}$ and σ^2 by:

$$\beta_i^{new} = \frac{1 - \beta_i N_{ii}}{{\mu_i}^2} \tag{23}$$

$${\sigma^2}^{new} = \frac{||\phi(\boldsymbol{y}) - \phi(X)\boldsymbol{\mu}||^2}{n - \sum_i(1 - \beta_i N_{ii})} \tag{24}$$

where N_{ii} is i_{th} diagonal element of the covariance from equation (20) and $\boldsymbol{\mu}$ is computed from equation (21) with current $\boldsymbol{\beta}$ and σ^2 values. The final optimal

set of $\boldsymbol{w}$, $\boldsymbol{\beta}$ and σ^2 are then learned by repeating the first step to update the covariance Σ (20) and mean $\boldsymbol{\mu}$ (21) of the feature weight vector $\boldsymbol{w}$ and the second step to update the hyper-parameters $\boldsymbol{\beta}$ (23) and σ^2 (24) iteratively.

RFVM learns a sparse feature weight vector $\boldsymbol{w}$ in which most of the elements are zeros. Those zero elements in $\boldsymbol{w}$ indicate that the corresponding features are irrelevant and should be filtered out. On the other hand, large values of elements in $\boldsymbol{w}$ indicate high importance of the related features. In this paper we use mutual information as the kernel function $K(\cdot, \cdot)$, which will be introduced in the following section. In that case, $K(X^T, \boldsymbol{y})$ actually measures the relevance between the response $\boldsymbol{y}$ and features in the data matrix X and $K(X^T, X)$ indicates the redundancy between features in the data matrix X. The likelihood maximization procedure of RFVM tends to maximize the relevance between the features and response and minimize the mutual redundancy within the features.

3.3 Mutual Information Feature Kernel

While kernels are usually defined over data examples in the feature space, the RFVM algorithm places the nonlinear kernel over the feature and response vectors for the purpose of feature selection. As we know, the mutual information [16] of two variables measures how much uncertainty can be reduced about one variable given the knowledge of the other variable. Such property can be used as the metric to measure the relevance between features. Given two discrete variables U and V with their observations denoted as u and v respectively, the mutual information I between them is formulated as

$$I(U, V) = \sum_{u \in U} \sum_{y \in V} p(u, v) log_2 \frac{p(u, v)}{p(u)p(v)} \tag{25}$$

where $p(u, v)$ is the joint probability density function of U and V, and $p(u)$ and $p(v)$ are the marginal probability density functions of U and V respectively.

Now given two feature vectors $\boldsymbol{f}_u$ and $\boldsymbol{f}_v$, we use the following way to calculate the value of their mutual information kernel $K(\boldsymbol{f}_u, \boldsymbol{f}_v)$. We regard all the elements in the vector $\boldsymbol{f}_u$ (or $\boldsymbol{f}_v$) as multiple observations of a variable $\boldsymbol{f}_u$ (or $\boldsymbol{f}_v$), and discretize those observations into bins for each variable. That is, we sort the values in the feature vectors $\boldsymbol{f}_u$ and $\boldsymbol{f}_v$ separately and discretize each vector into N bins, with the same interval for each bin. For example, if the the maximal value of $\boldsymbol{f}_u$ is u_{max} and the minimal value is u_{min}, the interval for each bin of feature vector $\boldsymbol{f}_u$ is $(u_{max} - u_{min})/N$. Now for each value u in feature vector $\boldsymbol{f}_u$ and v in feature vector $\boldsymbol{f}_v$, we assign $u = i$ and $v = j$ if u falls into the i_{th} bin and v falls into the j_{th} bin of their discretized regions respectively. The probability density functions $p(\boldsymbol{f}_u, \boldsymbol{f}_v)$, $p(\boldsymbol{f}_u)$ and $p(\boldsymbol{f}_v)$ are calculated as the ratio of the number of elements within corresponding bin to the length of vector n. As a result, we have

$$\begin{aligned} p(u = i) &= counts(u = i)/n \\ p(v = j) &= counts(v = j)/n \\ p(u = i, v = j) &= counts(u = i \; and \; v = j)/n \end{aligned}$$

and

$$K(\boldsymbol{f}_u, \boldsymbol{f}_v) = \sum_{i=1}^{N}\sum_{j=1}^{N} p(u=i, v=j) log_2 \frac{p(u=i, v=j)}{p(u=i)p(v=j)} \tag{26}$$

The mutual information kernel is symmetric and non-negative with $K(\boldsymbol{f}_u, \boldsymbol{f}_v) = K(\boldsymbol{f}_v, \boldsymbol{f}_u)$ and $K(\boldsymbol{f}_u, \boldsymbol{f}_v) \geq 0$. It also satisfies the mercer's condition [13], which guarantees the convergence of the proposed RFVM algorithm. In this paper we set the number of bins for discretization as $\log_2(m)$, where m is the number of features.

4 Experimental Results

Experimental results are presented in this section to demonstrate the effectiveness of our proposed RFVM algorithm. We compare RFVM with other two state of the art nonlinear feature selection algorithms in [3]and [9]. To be convenient, we call the algorithm proposed in [3] as P-SVM and that in [9] as FVM algorithm. All the experiments are conducted on a Pentium 4 machine with 3GHZ CPU and 1GB of RAM.

4.1 Nonlinear Feature Selection by RFVM

In order to verify that the proposed RFVM is able to catch the nonlinear dependency between the response and feature vectors, we simulated 1000 data examples with 99 features and one response. The response y is generated by the summation of three base functions of f_1, f_2, f_3 respectively, together with the Gaussian noise ϵ distributed as $N(0, 0.005)$.

$$\begin{aligned} y &= f(f_1, f_2, f_3, \cdots, f_{99}) \\ &= 9f_1 + 20(1-f_2)^3 + 17\sin(80 * f_3 - 7) + \epsilon \end{aligned}$$

The three base functions are shown in Figure 2(b)(c)(d), in which the first is a linear function and the other two are nonlinear. Figure 2(a) also plots the distribution of y with respect to the two nonlinear features f_2 and f_3. The values of features f_1, f_2, f_3 are generated by a uniform distribution in $[0, 1]$. The other 96 features $f_4, f_5, \cdots, f_{99}$ are generated uniformly in $[0, 20]$ and are independent with the response y.

We modified the MATLAB code provided by Mike Tipping [17] to implement RFVM for the task of feature selection. The RFVM is solved by updating the posterior covariance Σ in equation (20) and the mean $\boldsymbol{\mu}$ in equation (21) along with the hyper-parameters $\boldsymbol{\beta}$ in equation (23) and σ^2 in equation (24) iteratively using the two step procedure. The nonlinear dependencies between response and features by using mutual information kernel in RFVM. That is, we replace the dot product of the features and response, $< X^T, y >$ and $< X^T, X >$, by the precomputed mutual information kernel $K(X^T, y)$ and $K(X^T, X)$. The optimal

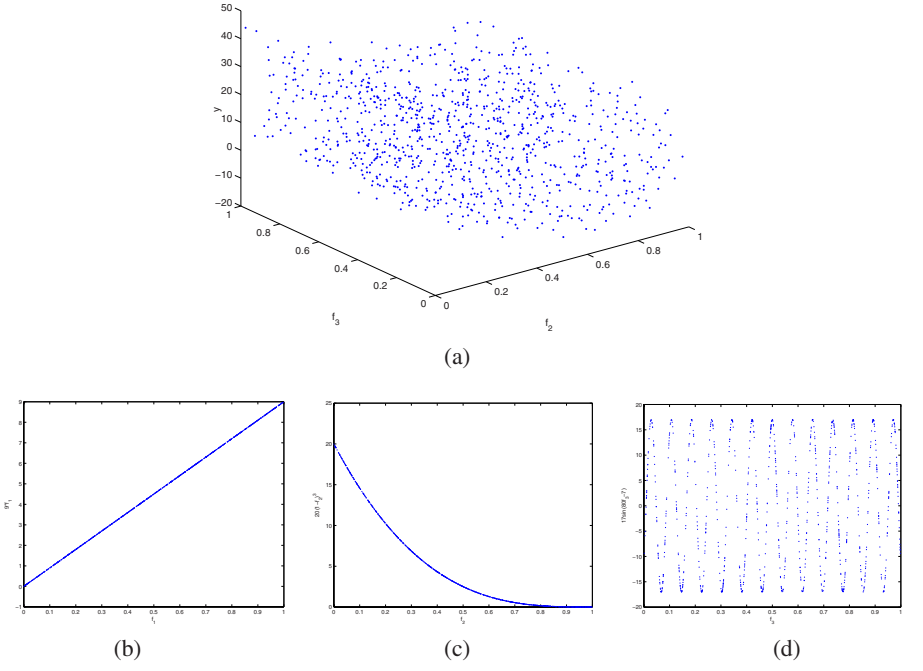


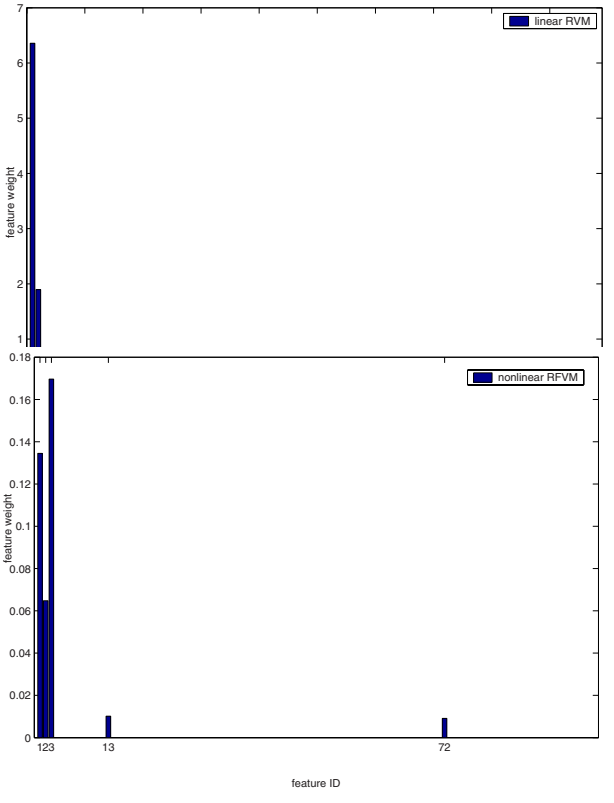
Fig. 2. (a) The distribution of response y with respect to two nonlinear features f_2 and f_3. The bottom three figures illustrate the three components of the simulated function f: (b) linear, (c) cubic and (d) sin.

set of feature vector weight $\boldsymbol{w}$ along with the hyper-parameters $\boldsymbol{\beta}$ and σ^2 in RFVM are automatically learned by a two step updating procedure. The initial values of the hyper-parameters are set as $\boldsymbol{\beta} = 10$ and $\sigma^2 = std(y)/10$.

Figure 3(a) and (b) plot the values of feature weights computed by linear RVM and nonlinear RFVM over the simulated data. From Figure 3(a), we see that the linear RVM can detect the linear dependent feature f_1, as well as the feature f_2 which has cubical relationship. The reason that f_2 is also detected by linear RVM is that the cubical curve can be approximated by a linear line in certain degree, which is shown in Figure 2(b). However, RVM missed the feature f_3 completely, which is a highly nonlinear feature with periodical sin wave. On the other hand, the nonlinear RFVM detects all the three features successfully, which is shown in Figure 3(b). Furthermore, the detected feature set is pretty sparse compared with the results of linear RVM.

4.2 Performance Comparison

This section compares the performance of RFVM algorithm with other nonlinear feature selection algorithms such as FVM in [9] and P-SVM in [3]. To demonstrate that RFVM is able to select most relevant features with much lower false



(b)

Fig. 3. (a) The histogram of feature weights from linear RVM. It detects f_1 and f_2 but misses the highly nonlinear relevant feature f_3. (b) The histogram of feature weights from nonlinear RVM. It detects all the three features.

alarm rate, we simulate another data set with 2000 data examples and 100 features. The first 20 features are simulated uniformly in $[-0.5, 0.5]$ and the rest are generated uniformly in $[0, 20]$ with Gaussian noise. The response y is the summation of functions $F_i(\cdot)$ on the first 20 features

$$y = \sum_{i=1}^{20} F_i(f_i) \ . \tag{27}$$

The basis function $F_i(\cdot)$ is randomly chosen from the pool of eight candidate functions

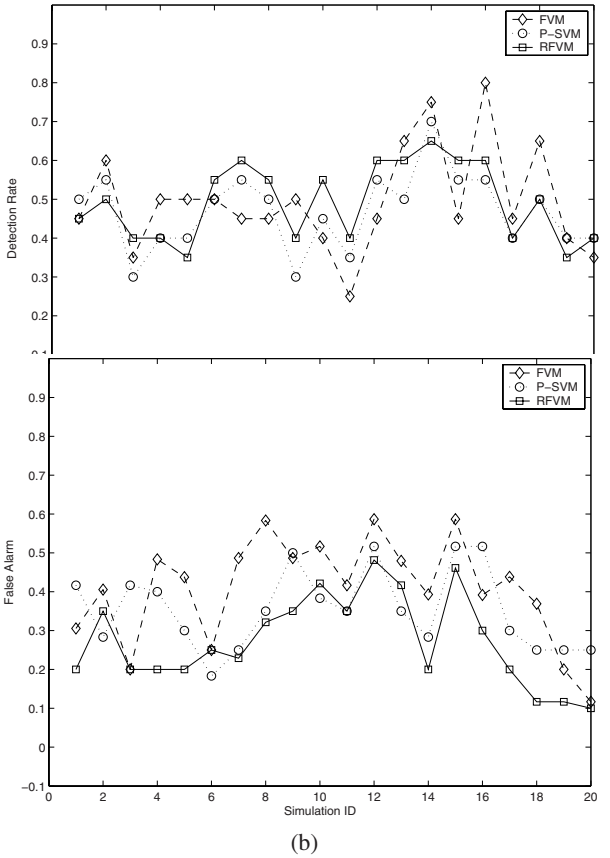
$$F_i(f_i) \in \{F_1(f_i), F_2(f_i), \cdots, F_8(f_i)\} \tag{28}$$

where the expressions of those candidate functions are described in Table 1. As you can see our simulation covers almost all kinds of common nonlinear relationships.

Table 1. The 8 basis function

$j =$	1	2	3	4
$F_j(f_i) =$	$40f_i$	$20(1 - f_i^2)$	$23f_i^3$	$20\sin(40f_i - 5)$
$j =$	5	6	7	8
$F_j(f_i) =$	$20e^{f_i}$	$-\log_2(\lvert f_i \rvert)$	$20\sqrt{1 - f_i}$	$20\cos(20f_i - 7)$

We divide the data into two parts, the first 1000 examples are used as training data to determine the parameter λ in FVM and ϵ, C in P-SVM by 10 fold cross validation. The rest 1000 data examples are used for test. The performances of those algorithms are compared in terms of detection rate and false alarm rate. We run 20 rounds of such simulations and present the results in Figure 4. Figure 4(a) plots the detection rate of RFVM together with those of FVM and P-SVM. It shows that RFVM maintains comparable detection rate as the other



(b)

Fig. 4. (a) The detection rates of FVM, P-SVM and RFVM. (b) The false alarm rates of FVM, P-SVM and RFVM.

two algorithms. Note since the nonlinear relationship (27) generated in our simulation is very strong, the range of detection rates for those algorithms is reasonable. Figure 4(b) plots the false alarm rates of FVM, P-SVM and RFVM algorithms. It demonstrates that RFVM has lower false alarm rate generally, which is due to the sparseness of RFVM in the example space compared with FVM and P-SVM. Also note in the experiment we don't need to predetermine any parameters in RFVM since the parameters are automatically learned by two step maximum likelihood method, while FVM and P-SVM are both sensitive to parameters and need extra efforts of cross validation to determine those values.

5 Conclusions

This paper has proposed a new method, the "Relevance Feature Vector Machine"(RFVM), to detect features with nonlinear dependency. Compared with other state of the art nonlinear feature selection algorithms, RFVM has two unique advantages based on our theoretical analysis and experimental results. First, by utilizing the highly sparseness nature of RVM, the RFVM algorithm reduces the false alarms in feature selection significantly while still maintains desirable detection rate. Furthermore, unlike other SVM based nonlinear feature selection algorithms whose performances are sensitive to the selection of parameter values, RFVM learns the hyper-parameters automatically by maximizing the "marginal likelihood" in the second step of the two-step updating procedure. In the future, we will apply RFVM to some real applications to further demonstrate the advantages of our algorithm.

References

1. Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. Journal of Machine Learning Research 1, 211–244 (2001)
2. Bellman, R.E.: Adaptive Control Processes. Princeton University Press, Princeton, NJ (1961)
3. Hochreiter, S., Obermayer, K.: Nonlinear feature selection with the potential support vector machine. In: Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L. (eds.) Feature extraction, foundations and applications, Springer, Berlin (2005)
4. Figueiredo, M., Jain, A.K.: Bayesian Learning of Sparse Classifiers. In: Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition, vol. 1, 35–41 (2001)
5. Figueiredo, M.A.T.: Adaptive sparseness for supervised learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(9), 1150–1159 (2003)
6. BΦ, T.H., Jonassen, I.: New feature subset selection procedures for classification of expression profiles, Genome Biology, 3 research 0017.1-0017.11 (2000)
7. Burges, C.: Simplified support vector decision rules. In: Proc. of the Thirteenth International Conf. on Machine Learning, pp. 71–77. Morgan Kaufmann, Seattle (1996)
8. Aizerman, M.E., Braverman, Rozonoer, L.: Theoretical foundations of the potential function method in pattern recognition learning. Automation and Remote Control 25, 821–837 (1964)

9. Li, F., Yang, Y., Xing, E.P.: From Lasso regression to Feature Vector Machine, Advances in Neural Information Processing Systems, 18 (2005)
10. Faul, A., Tipping, M.E.: Analysis of sparse bayesian learning. In: Dietterich, T., Becker, S., Ghahramani, Z. (eds.) Advances in Neural Information Processing Systems 14, pp. 383–389. MIT Press, Cambridge, MA (2002)
11. Faul, A., Tipping, M.: A variational approach to robust regression, in Artificial Neural Networks. In: Dorffner, G., Bischof, H., Hornik, K. (eds.), pp. 95–202 (2001)
12. Roth, V.: The Generalized LASSO, V. IEEE Transactions on Neural Networks, Dorffner, G. vol. 15(1). (2004)
13. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
14. Smola, A.J., Scholkopf, B.: A tutorial on support vector regression, NEUROCOLT Technical Report NC-TR-98-030, Royal Holloway College, London (1998)
15. Long, F., Ding, C.: Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8), 1226–1238 (2005)
16. Guiasu, Silviu.: Information Theory with Applications. McGraw-Hill, New York (1977)
17. Tipping, M.E.: Microsoft Corporation, `http://research.microsoft.com/MLP/RVM/`
18. Tibshirani, R.: Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Series B (Methodological) 58(1), 267–288 (1999)
19. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
20. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
21. Reeves, S.J., Zhao, Z.: Sequential algorithms for observation selection. IEEE Transactions on Signal Processing 47, 123–132 (1999)