International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 21, No. 8 (2007) 1279–1292
© World Scientific Publishing Company

World Scientific
www.worldscientific.com

FEATURE SELECTION BASED ON MINIMUM ERROR MINIMAX PROBABILITY MACHINE

ZENGLIN XU*, IRWIN KING† and MICHAEL R. LYU‡

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
**zlxu@cse.cuhk.edu.hk*
†*king@cse.cuhk.edu.hk*
‡*lyu@cse.cuhk.edu.hk*

Feature selection is an important task in pattern recognition. Support Vector Machine (SVM) and Minimax Probability Machine (MPM) have been successfully used as the classification framework for feature selection. However, these paradigms cannot automatically control the balance between prediction accuracy and the number of selected features. In addition, the selected feature subsets are also not stable in different data partitions. Minimum Error Minimax Probability Machine (MEMPM) has been proposed for classification recently. In this paper, we outline MEMPM to select the optimal feature subset with good stability and automatic balance between prediction accuracy and the size of feature subset. The experiments against feature selection with SVM and MPM show the advantages of the proposed MEMPM formulation in stability and automatic balance between the feature subset size and the prediction accuracy.

Keywords: Feature selection; classification; minimax probability machine; minimum error minimax probability machine; support vector machine.

1. Introduction

Feature selection has attracted a lot of interest in the machine learning field.[3,5,9] The problem of feature selection in this paper is defined as follows: given a data set $\mathcal{D} = \{(\mathbf{z}_1, C_1), (\mathbf{z}_2, C_2), \ldots, (\mathbf{z}_N, C_N)\} \in \mathbb{R}^{n+1}$ where n is the dimension of $\mathbf{z}_i$ and $C_i \in \mathbb{R}$ for $1 \leq i \leq N$, the objective of feature selection is to find a separating hyperplane $f = \mathbf{w}^T\mathbf{z} - b$ to discriminate these two classes and to further make the most of the elements in $\mathbf{w}$ to be zeros. Feature selection can be used as a process to reduce data dimensions for classification by removing nondiscriminant features. More specifically, this problem can be formulated as the minimization of l_1-norm[a]

[a] In fact, this should be l_0-norm (which is defined as the cardinality of a set), but it will result in a combinatorial problem, which is too complex for the high-dimensional problem. l_0-norm of $\mathbf{w}$ is often substituted with l_1-norm of $\mathbf{w}$.

of **w** subject to some classification framework:

$$\begin{aligned}&\min_{\mathbf{w},b} \|\mathbf{w}\|_1,\\ &\text{s.t. some classification framework.}\end{aligned} \tag{1}$$

The above formulation can be categorized as an embedded approach,[3] where the feature selection process is embedded into the classification framework. For other categorizations, i.e. the filter approach and the wrapper approach, the readers can refer to earlier work[7–9,12,15,17,19] for more details.

The objective of feature selection has two folds: one is to select a small feature subset and the other is to maintain high classification accuracy. Then a question appears: how to control the balance between the prediction accuracy and the number of selected features? Currently, this problem is still an open problem. The state-of-the-art classifier Support Vector Machine (SVM)[14,16] has been successfully applied in feature selection problem.[4,13,18] Bhattacharyya formulates Minimax Probability Machine (MPM)[10,11] as another classification framework into the feature selection problem.[1] But in the SVM formulation for feature selection, there is no explicit worst-case accuracy bound and thus it is hard to control the number of selected features as well as the prediction accuracy. In the MPM formulation,[1] although it has the explicit accuracy bound, the user needs to handle this problem by hand.

In addition, some feature selection systems may require high stability on selected features, i.e. selecting similar features in different runs and data partitions. However, the feature selection algorithms with SVM and MPM do not consider this issue. SVM utilizes data points on the boundary (called support vectors) to determine the separating hyperplane. MPM utilizes the mean and the covariance of each class to find a decision hyperplane. Besides, MPM assumes the same worst-case accuracy bound for each class. Would the bias introduced by different information used in the classification frameworks affect the stability of resulted feature selection algorithms?

Minimum Error Minimax Probability Machine (MEMPM) was recently proposed for data classification.[6] MEMPM includes and extends MPM by assuming different worst-case accuracy bounds for different classes, in order to better capture the data distribution. This paper outlines MEMPM on feature selection to attack the above problems. The special properties of the proposed feature selection algorithm using MEMPM include:

(1) **Controllable Balance**. MEMPM is a global learning classifier and it better captures the data scatter geometrically; thus, it separates the data more reasonably. The proposed feature selection algorithm with MEMPM has different explicit worst-case accuracy bounds for different classes. In addition, we propose two different criteria to control the balance between the accuracy bounds and the number of selected features.

(2) High Stability. Geometrically, MEMPM tries to find two tangent ellipsoids with different radii to include each class of data. The global learning scheme in MEMPM plus the good capture of data scatter makes MEMPM insensitive to different data partitions. On the other hand, SVM tries to find features to maximize the margin. However, the margin is sensitive to different data partitions.

The paper is organized as follows. In Sec. 2, related feature selection algorithms are discussed. In Sec. 3, the feature selection algorithm with MEMPM is proposed. In Sec. 4, we analyze the experimental results of different algorithms for feature selection. Section 5 concludes the paper and lists future directions.

2. Related Feature Selection Methods

Regarded as two important classification techniques, SVM and MPM are used as the classification framework for feature selection. In the following subsections, we examine, in detail, the characteristics of these frameworks and their correspondent feature selection algorithms.

2.1. l_1-*SVM for feature selection*

The l_1-SVM formulation is employed as a linear programming framework for feature selection in Ref. 4. The difference from the feature selection formulation to the original l_1-SVM formulation is that $\mathbf{w}$ is written as $\mathbf{w} = \mathbf{u} - \mathbf{v}$ where $\mathbf{u} = (u_1, \ldots, u_N) \in \mathbb{R}^N$, $\mathbf{v} = (v_1, \ldots, v_N) \in \mathbb{R}^N$, and all elements in $\mathbf{u}$ and $\mathbf{v}$ are non-negative. Besides, the l_1-norm of $\mathbf{w}$ is replaced by $(\mathbf{u}+\mathbf{v})^T\mathbf{e}$ where $\mathbf{e}$ is a column vector with each element being one. More specifically, the l_1-SVM formulation can be stated as the following linear programming problem:

$$\begin{aligned} &\min_{\mathbf{u},\mathbf{v},b} \ (\mathbf{u}+\mathbf{v})^T\mathbf{e} + C\sum_{i=1}^{N_x+N_y} \xi_i/(N_x+N_y) \\ &\text{s.t.} \ \ (\mathbf{u}-\mathbf{v})^T\mathbf{x}_i - b \ge 1-\xi_i, \qquad 1 \le i \le N_x, \\ &\qquad -(\mathbf{u}-\mathbf{v})^T\mathbf{y}_j + b \ge 1-\xi_j, \quad 1 \le i \le N_y, \\ &\qquad u_i \ge 0, \quad v_i \ge \mathbf{0}, \quad \xi_i \ge 0, \quad 1 \le i \le N, \end{aligned} \tag{2}$$

where $\mathbf{x}$ and $\mathbf{y}$ denote samples that belong to the positive class and the negative class, respectively. N_x and N_y are the cardinality of $\mathbf{x}$ and $\mathbf{y}$. $\xi \in \mathbb{R}^N$ is introduced as a slack variable to represent the margin error in the nonseparable case.

The above formulation is a linear programming problem. One notable advantage of the feature selection algorithm with l_1-SVM is its efficiency.

2.2. *MPM formulation for feature selection*

Different from SVM, where the hyperplane is determined by the support vectors, MPM[11] uses the first two moments, i.e. the means and the covariance matrices, to bound the classification accuracy of each class under the worst-case scenario. We use $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ to note the mean of the class $\mathbf{x}$ and that of the class $\mathbf{y}$, respectively. Their relevant covariance matrices are represented as $\Sigma_{\mathbf{x}}$ and $\Sigma_{\mathbf{y}}$ accordingly. The objective of MPM is to minimize the worst-case misclassification probability of future data points. The authors formulate the classification problem with the Chebychev inequality to the following expression:

$$\begin{aligned} &\max_{\alpha,\mathbf{w},b}\ \kappa(\alpha) \\ &\text{s.t.}\ \ \mathbf{w}^T\bar{\mathbf{x}} - b \geq \kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}, \\ &\qquad -\mathbf{w}^T\bar{\mathbf{y}} + b \geq \kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}, \end{aligned} \tag{3}$$

where α is the worst-case misclassification probability and $\kappa(\alpha) = \sqrt{\frac{\alpha}{1-\alpha}}$.

The advantages of MPM are that it makes no assumption on data distribution and it has an explicit lower bound on the worst-case prediction accuracy. The disadvantage is mainly that the performance depends on the estimation of means and covariance matrices. When there are not enough examples, the estimation may not be very accurate.

Motivated by the formulas of MPM, Bhattacharyya[1] used the above two inequalities to constrain the feature selection problem. The objective is changed to minimize the 1-norm of $\mathbf{w}$ for a given α. The formulation can be stated as follows:

$$\begin{aligned} &\min_{\mathbf{u},\mathbf{v},b}\ (\mathbf{u}+\mathbf{v})^T\mathbf{e} \\ &\text{s.t.}\ \ (\mathbf{u}-\mathbf{v})^T\bar{\mathbf{x}} - b \geq \kappa(\alpha)\sqrt{(\mathbf{u}-\mathbf{v})^T\Sigma_{\mathbf{x}}(\mathbf{u}-\mathbf{v})}, \\ &\qquad -(\mathbf{u}-\mathbf{v})^T\bar{\mathbf{y}} + b \geq \kappa(\alpha)\sqrt{(\mathbf{u}-\mathbf{v})^T\Sigma_{\mathbf{y}}(\mathbf{u}-\mathbf{v})}, \\ &\qquad (\mathbf{u}-\mathbf{v})^T - b \geq 1, \\ &\qquad -(\mathbf{u}-\mathbf{v})^T + b \geq 1, \\ &\qquad u_i \geq 0, \quad v_i \geq \mathbf{0}, \quad 1 \leq i \leq N, \end{aligned} \tag{4}$$

where $(\mathbf{u}-\mathbf{v})^T - b \geq 1$ and $-(\mathbf{u}-\mathbf{v})^T + b \geq 1$ are introduced to reduce the extra free degree of the variables.

One advantage of the above formulation is that it involves an explicit upper bound of worst-case misclassification accuracy after selecting a subset of features. The user can control the tradeoff between the number of features and worst-case misclassification accuracy by controlling the bound α.

3. Proposed Feature Selection Model

Minimum Error Minimax Probability Machine (MEMPM)[6] extends MPM by assuming different worst-case error bounds and thus minimizes the worst-case Bayes error rate of future data. MPM assumes the same worst-case misclassification probability for each class; however, this is unreasonable in real problems. The MEMPM model is shown in the following:

$$\begin{aligned} &\max_{\mathbf{w},b,\alpha,\beta} \ \theta\alpha + (1-\theta)\beta \\ &\text{s.t.} \quad \mathbf{w}^T\bar{\mathbf{x}} - b \geq \kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}, \\ &\qquad -\mathbf{w}^T\bar{\mathbf{y}} + b \geq \kappa(\beta)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}, \end{aligned} \tag{5}$$

where α and β represent the worst-case lower accuracy bound for classes $\mathbf{x}$ and $\mathbf{y}$, respectively. θ is a constant within $[0, 1]$ to control the balance between α and β.

3.1. *Comparison among SVM, MPM and MEMPM*

Before formulating MEMPM to solve the feature selection problem, a comparison among these classification frameworks is summarized in Table 1.

We focus on comparing their solving methods and learning schemes. The solution of l_2-SVM can be achieved by solving a Quadratic Programming (QP) problem, while the solution of MPM and MEMPM reduces to a Second Order Cone Programming (SOCP) and Fractional Programming (FP) problem,[6,11] respectively. Generally, solving SVM and MPM reduces to the same time complexity as $\mathcal{O}(n^3+Nn^2)$,[11] where N is the number of data points and n is the dimension of the input space; while solving MEMPM reduces to a time complexity scaled as $\mathcal{O}(Ln^3 + Nn^2)$, where L indicates the steps for linearly searching α. A special property of MPM and MEMPM is that there is a worst case accuracy bound for future prediction. From the perspective of learning, SVM represents the local learning scheme, while MPM and MEMPM represent the global learning scheme. In the next section, we will examine how the learning schemes can affect by experimental results the feature subset selection.

Table 1. Comparisons of different classifiers.

Model	l_2-SVM	MPM	MEMPM
Generalization	—	—	>MPM
Worst Case Accuracy Bound	No	Yes	Yes
Learning Scheme	Local	Global	Global
Time Complexity	$\mathcal{O}(n^3 + Nn^2)$	$\mathcal{O}(n^3 + Nn^2)$	$\mathcal{O}(Ln^3 + Nn^2)$
Problem Genre	QP	SOCP	Sequential FP

3.2. *MEMPM formulation for feature selection*

For a given worst-case accuracy bound $\theta\alpha + (1-\theta)\beta$, the feature selection problem with MEMPM can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{u},\mathbf{v},b}\ & (\mathbf{u}+\mathbf{v})^T\mathbf{e} \\ \text{s.t.}\ & (\mathbf{u}-\mathbf{v})^T\bar{\mathbf{x}} - b \geq \kappa(\alpha)\sqrt{(\mathbf{u}-\mathbf{v})^T\Sigma_{\mathbf{x}}(\mathbf{u}-\mathbf{v})}, \\ & -(\mathbf{u}-\mathbf{v})^T\bar{\mathbf{y}} + b \geq \kappa(\beta)\sqrt{(\mathbf{u}-\mathbf{v})^T\Sigma_{\mathbf{y}}(\mathbf{u}-\mathbf{v})}, \\ & (\mathbf{u}-\mathbf{v})^T - b \geq 1, \\ & -(\mathbf{u}-\mathbf{v})^T + b \geq 1, \\ & u_i \geq 0, \quad v_i \geq \mathbf{0}, \quad 1 \leq i \leq N, \end{aligned} \tag{6}$$

where $\kappa(\alpha) = \sqrt{\frac{\alpha}{1-\alpha}}$ and $\kappa(\beta) = \sqrt{\frac{\beta}{1-\beta}}$. Similar to the feature selection algorithm with MPM, we need to control the feature selection process by specifying the worst case accuracy bounds α and β.

The objective of feature selection problem with MEMPM has two folds — one is to find a sparse $\mathbf{w}$, the other is to maintain the accuracy bounds α and β as large as possible. In Ref. 1, the user needs to input the worst-case accuracy bound α for the MPM formulation. However, the value of α cannot approach 1 infinitely and its bound is determined by the moments. To avoid handling this problem manually, here we propose two criteria to automatically find the appropriate α and β for the MEMPM formulation. One is to use an iterative search method to find the maximal $\theta\alpha + (1-\theta)\beta$ and the sparse $\mathbf{w}$ under the prespecified α and β . The other is to use a similar search method to find the α, β, and $\mathbf{w}$ to make $\frac{\theta\alpha+(1-\theta)\beta}{\text{NSF}}$ (where NSF means the number of selected features[b]) maximal. In the first criterion, the maximal α and β may result in a high prediction accuracy, but does not necessarily assure a sparse $\mathbf{w}$. The second criterion considers both the worst-case accuracy bound and the sparsity of $\mathbf{w}$. For convenience, these two criteria are denoted as Criterion 1 and Criterion 2, respectively. The iterative search method for Criterion 1 is described in Algorithm 1.

In the case of Criterion 2, the search method is similar except that the objective is to maximize $\frac{\theta\alpha+(1-\theta)\beta}{\text{NSF}}$.

Remark. It is interesting to note that a series of criteria within the range of the above two criteria can be obtained by setting an adjustment factor t where $1 \leq t \leq \text{NSF}$. In this way, the resulting criterion is $t\frac{\theta\alpha+(1-\theta)\beta}{\text{NSF}}$. An example is to set $t = \log(\text{NSF})$ if we want to decrease the effect of NSF.

[b] If a relative measure $\frac{|\mathbf{w}_i|}{\max_i\{|\mathbf{w}_i|\}} \geq \epsilon$, then this feature is selected; otherwise, it is not selected. In this paper, $\epsilon = 0.001$.

Algorithm 1 The Feature Selection Algorithm with MEMPM. Input: $S = (X, Y)$: the training data; (α_0, β_0): the initial value of α and β; θ: a weighing constant; α_{step}: the iterative step of α; ϵ: the termination parameter. Output: $(\mathbf{w}, b)$

```
$\alpha_i = \alpha_0$
while $\alpha_i \leq 1$ do
    $\beta_n = 1$
    while $|\beta_0 - \beta_n| \leq \epsilon$ do
        $\beta_j = (\beta_0 + \beta_n)/2$
        Run an SOCP procedure to solve the MEMPM formulation for feature
        selection with current $(\alpha_i, \beta_j)$
        if $\beta_j$ is feasible then
            $\beta_0 = \beta_j$
        else
            $\beta_n = \beta_j$
        end if
        if both $\alpha_i$ and $\beta_j$ are feasible then
            Calculate $\theta\alpha + (1 - \theta)\beta$
            Keep $(\alpha_i, \beta_j, \mathbf{w}, b)$ in memory
        end if
    end while
    $\alpha = \alpha + \alpha_{step}$
end while
Return the optimal $(\mathbf{w}, b)$ with the maximal $\theta\alpha + (1 - \theta)\beta$.
```

4. Experiments

The proposed feature selection algorithm is evaluated in real world data sets. Here we conduct two experiments. The first experiment is to compare the criteria for FS-MEMPM (here and thereafter, we use a prefix “FS-” to represent feature selection algorithms based on the revelent classification frameworks). The second experiment is used to compare the performance of different feature selection algorithms.

4.1. *Experiment protocol*

The data sets used are Sonar, Ionosphere, Pima and Wdbc from the UCI machine learning repository.[2] These data sets have 60, 34, 8 and 30 features, respectively. The experiments are conducted on a PC with Intel Pentium 4 CPU, 3.20 GHz and 0.99 GB of RAM. The parameters C of SVM is tuned by ten-cross validation to maximize the test accuracy. All the experimental results are obtained by averaging ten trials and each trial is with ten-cross fold validation for each data set.

In the first experiment, FS-MEMPM and FS-MPM equipped with Criterion 1 and Criterion 2 are compared. We omit the search method of FS-MPM because it is similar and even simpler since it has only one control parameter α. In the second experiment, FS-MEMPM (with Criterion 2) is compared with FS-SVM and FS-MPM (with Criterion 2) on the prediction accuracy, the number of selected features and the stability on selecting similar feature subsets.

4.2. *Experimental results and discussion*

4.2.1. *Comparison of different criteria*

The prediction accuracy and the number of selected features of FS-MEMPM with Criterion 1 and Criterion 2 are listed in Table 2. For an easy comparison with FS-MEMPM, the results of FS-MPM are also listed in Table 3.

In each table, the second row denotes the classification accuracies obtained without feature selection on each data set by MEMPM or MPM. Then the next two rows are the number of selected features and the test accuracy obtained by Criterion 1. The last two rows are the number of selected features and the test accuracy obtained by Criterion 2.

For the first experiment, we can see that the test accuracy degrades a little or shows improvement on some data sets after adopting Criterion 1 compared to that before feature selection. But it only removes a few nondiscriminant features. Criterion 1 is recommended in the case that test accuracy is a much more important measure. However, the removal of features is very limited and so Criterion 1 is very conservative. Feature selection algorithm with Criterion 2 discards many irrelevant features while the test accuracy does not change much. In the case of Sonar, the test accuracy even improves slightly. Therefore, it is recommended to adopt Criterion 2 when the dimension reduction is much more important than the test accuracy. The results can be interpreted as follows: when the worst-case accuracy bound $\theta\alpha+(1-\theta)\beta$ is very large, many features are required to contribute to the covariance

Table 2. Comparison of different selection criteria in FS-MEMPM.

Data Set	Sonar	Ionosphere	Pima	Wdbc
Accuracy Before Feature Selection (%)	74.7	**88.5**	74.5	**97.0**
NSF (by Criterion 1)	57.97	32.58	7.04	24.66
Test Accuracy (%)	**74.9**	87.6	**76.4**	94.6
NSF (by Criterion 2)	**13.69**	**6.5**	**2.1**	**1**
Test Accuracy (%)	74.8	86.7	73.2	90.7

Table 3. Comparison of different selection criteria in FS-MPM.

Data Set	Sonar	Ionosphere	Pima	Wdbc
Accuracy Before Feature Selection (%)	75.3	84.7	75.6	**97.0**
NSF (by Criterion 1)	56.59	31.5	6.68	24.37
Test Accuracy (%)	**75.4**	**84.8**	**75.7**	**97.0**
NSF (by Criterion 2)	**34.59**	**9.96**	**3.3**	**1**
Test Accuracy (%)	74.6	83.6	73.1	90.1

matrices and thus **w** is not very sparse. When α and β are not very large, the requirement of features' contribution to the covariance matrices can be slightly slackened and thus result in a sparse **w**. Due to the different nature of data and the unknown difference between the bound α and β and the real future prediction accuracy, a slightly smaller α or β does not necessarily result in lower accuracy.

The difference between the two criteria also suggests that there should exist an in-between criterion, which selects a few more features than Criterion 2, but improves the prediction accuracy. These criteria provide FS-MEMPM a way for users to control the balance between the prediction accuracy and the number of selected features. It can be also similarly analyzed for Criterion 1 and Criterion 2 in FS-MPM.

4.2.2. *Comparison among different feature selection algorithms*

Table 4 shows the comparison results for all algorithms on the average number of selected features and the prediction accuracy of resulted sparse classifiers. For the convenience of comparison, we put the classification accuracy without feature selection as one row for each data set.

First, looking at the number of selected features, we observe that FS-MEMPM always selects the least number of features. Geometrically, FS-MPM uses fewer features to include these two classes into two tangent ellipsoids with the same radius; while FS-MEMPM tries to use fewer features to include the classes into two tangent ellipsoids with different radii. In fact, these two classes often have different distributions; thus ellipsoids with different radii can easily contain the data. Further, the bounds in MEMPM are much tighter than the bounds in MPM.[6] As a result, FS-MEMPM is more reasonable than MPM and selects fewer features. Furthermore,

Table 4. Comparisons of experimental results on all data sets.

Feature Selection Algorithms	FS-SVM	FS-MPM	FS-MEMPM
Data Set		Sonar	
Accuracy Before Feature Selection (%)	76.0	75.3	74.7
The Number of Selected Features	16.3	34.59	**13.69**
Test Accuracy (%)	74.3	74.6	**74.8**
Data Set		Ionosphere	
Accuracy Before Feature Selection (%)	87.3	84.8	88.5
The Number of Selected Features	18.01	9.96	**6.5**
Test Accuracy (%)	**87.5**	83.6	86.7
Data Set		Pima	
Accuracy Before Feature Selection (%)	77.1	75.7	74.5
The Number of Selected Features	3.95	3.3	**2.1**
Test Accuracy (%)	**75.9**	73.1	73.2
Data Set		Wdbc	
Accuracy Before Feature Selection (%)	97.4	97.0	97.0
the Number of Selected Features	4	**1**	**1**
Test Accuracy (%)	**95.2**	90.0	90.7

the statistics on selected feature subset shows that the feature set of FS-MEMPM is almost the subset of FS-MPM. FS-SVM tries to project the data to a low dimension space to separate two classes of data as much as possible. Comparing with FS-SVM, the good performance of FS-MEMPM benefits from the automatic balance between the accuracy bound and the size of feature subset.

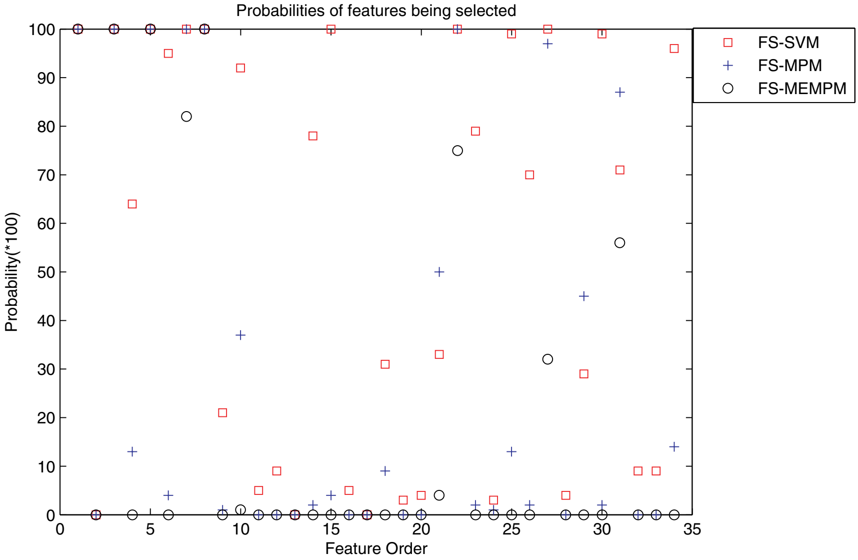
Second, looking at the test accuracy, FS-SVM shows improvements on Ionosphere; while FS-MPM and FS-MEMPM improves on Sonar. Generally FS-SVM has higher prediction accuracy but at the cost of more selected features. It is important to notice that our proposed algorithm obtains a good balance between the prediction accuracy and the number of selected features. One typical example is text classification, where the bags-of-words representation scheme produces a high dimension in the scale of thousands or millions. Effectively reducing the number of dimensionality will greatly decrease the response time and reduce the memory requirement.

Figure 1 describes the frequencies of features being selected by these three feature selection algorithms. With the limited space, only results from Ionospere and Sonar are listed. Similar results can be observed on Pima. Because of easy separability of data itself, three algorithms always find the same features on Wdbc in all runs and all divisions. It is observed that the algorithms differ much on consistently selecting same features in different runs and partitions. The features selected by FS-SVM and FS-MPM are very different in different runs and partitions of these data sets, since the frequencies of features being selected are further away from 0 or 1. However, FS-MEMPM selects more similar features in all partitions, which can be observed from points approximating to the floor and the ceiling. This shows that global classifiers which accurately describe the distributions of data can have a more stable performance than local classifiers in the feature selection task.

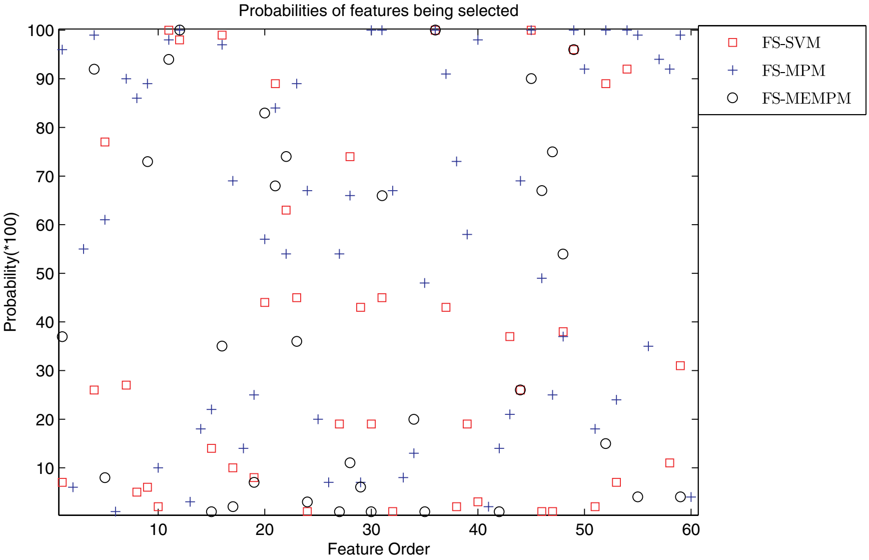
5. Conclusion and Future Work

A feature selection algorithm based on MEMPM, noted as FS-MEMPM, is proposed in this paper. Two criteria to control the balance between the number of selected features and the prediction accuracy are proposed for FS-MEMPM. The experimental results show that in Criterion 1, generally the algorithm does not lose prediction accuracy but removes only a few features; while in Criterion 2, the feature selection algorithm selects the least number of features and at the same time maintains quite high prediction accuracy. These criteria make FS-MEMPM controllable in balancing between the prediction accuracy and the feature subset size. The experiment among the proposed FS-MEMPM and other two feature selection algorithms FS-SVM and FS-MPM shows that FS-MEMPM is more likely to select the least number of features and is more stable in selecting similar feature subsets in different runs and data partitions.

A future work is to find an optimal criteria for FS-MEMPM to explore the optimal tradeoff between the lower bound of the prediction accuracy and the number



(a) Probabilities of features being selected in iono



(b) Probabilities of features being selected in sonar

Fig. 1. The scatter plot of the probability of features being selected by FS-SVM, FS-MPM and FS-MEMPM on data set Ionosphere and Sonar. The algorithm is thought as more stable when the probability is closer to 0 or 1. In this graph, (a) and (b) are the results observed on Ionoshere and Sonar, respectively.

of selected features. Another future work is to extend the current feature selection algorithm to a nonlinearly separable case by using the kernel trick. Besides, we plan to apply the proposed algorithm in gene selection from microarray data and text categorization, where the dimension is usually very large. We believe the balance between the prediction accuracy and the number of selected features provided by our algorithm will benefit the feature selection process.

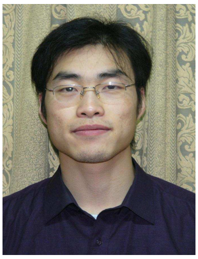
Acknowledgments

The work described in this paper is fully supported by two grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4205/04E and Project No. CUHK4235/04E).

References

1. C. Bhattacharyya, Second order cone programming formulation for feature selection, *J. Mach. Learn. Res.* **5** (2004) 1417–1433.
2. C. L. Blake and C. J. Merz, *Repository of Machine Learning Databases*, University of California, Irvine (1998) http://www.ics.uci.edu/∼mlearn/mlrepository.html.
3. A. Blum and P. Langley, Selection of relevant features and examples in machine learning, *Artif. Intell.* **97**(1–2) (1997) 245–271.
4. L. R. Grate, C. Bhattacharyya, M. I. Jordan and I. S. Mian, Simultaneous relevant feature identification and classification in high-dimensional spaces, *Proc. Second Int. Workshop on Algorithms in Bioinformatics (WABI-2002)* **2452**, eds. R. Guigo and D. Gusfield (2002), pp. 1–9.
5. I. Guyon and A. Elisseeff, An introduction to variable and feature selection, *J. Mach. Learn. Res.* **3** (2003) 1157–1182.
6. K. Huang, H. Yang, I. King, M. R. Lyu and L. Chan, Minimum error minimax probability machine, *J. Mach. Learn. Res.* **5** (2004) 1253–1286.
7. A. Jain and D. Zongker, Feature selection: evaluation, application, and small sample performance, *IEEE Trans. Patt. Anal. Mach. Intell.* **19**(2) (1997) 153–158.
8. K. Kira and L. A. Rendell, A practical approach to feature selection, *Proc. Ninth Int. Workshop on Machine Learning*, San Francisco, CA, USA (Morgan Kaufmann Publishers Inc., 1992), pp. 249–256.
9. R. Kohavi and G. H. John, Wrappers for feature subset selection, *Artif. Intell.* **97**(1–2) (1997) 273–324.
10. G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya and M. I. Jordan, Minimax probability machine, *Advances in Neural Information Processing Systems (NIPS 15)* (2001).
11. G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya and M. I. Jordan, A robust minimax approach to classification, *J. Mach. Learn. Res.* **3** (2002) 555–582.
12. Y. Liu, Z. Qin, Z. Xu and X. He, Feature selection with particle swarms, in *Computational and Information Science*, Lecture Notes in Computer Science, Vol. 3314 (2004), pp. 425–430.
13. A. Rakotomamonjy, Variable selection using SVM-based criteria, *J. Mach. Learn. Res.* **3** (2003) 1357–1370.
14. A. J. Smola, P. L. Bartlett, B. Scholkopf and D. Schuurmans, *Advances in Large Margin Classifiers* (The MIT Press, 2000).

15. Y. Sun and J. Li, Iterative relief for feature weighting, *Proc. 23th Int. Conf. Machine Learning (ICML-2006)* (2006).
16. V. N. Vapnik, *Statistical Learning Theory* (John Wiley & Sons, 1998).
17. D. Wang, D. S. Yeung, E. C. C. Tsang and L. Shi, Gene selection through sensitivity analysis of support vector machines, in *CompLife 2005*, LNBI, Vol. 3695 (2005), pp. 117–127.
18. J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio and V. Vapnik, Feature selection for SVMs, NIPS (2000), pp. 668–674.
19. J. Yang and V. Honavar, Feature subset selection using a genetic algorithm, *IEEE Intell. Syst.* **13** (1998) 44–49.



Zenglin Xu received his B.S. degree in computer science and technology from Xi'an Polytechnic University, China in 2002 and the M.S. degree in computer software and theory from Xi'an Jiaotong University, China in 2005. Currently, he is a Ph.D. candidate in the Department of Computer Science and Engineering of the Chinese University of Hong Kong.

His research interests include machine learning, pattern recognition, information retrieval, data mining and evolutionary computation.

Irwin King received the B.Sc. degree in engineering and applied science from California Institute of Technology, Pasadena, in 1984. He received his M.Sc. and Ph.D. degree in computer science from the University of Southern California, Los Angeles, in 1988 and 1993 respectively. He joined the Chinese University of Hong Kong in 1993.

He is a member of ACM, IEEE Computer Society, International Neural Network Society (INNS), and Asian Pacific Neural Network Assembly (APNNA). Currently, he is serving in the Neural Network Technical Committee (NNTC) and the Data Mining Technical Committee under the IEEE Computational Intelligence Society (formerly the IEEE Neural Network Society). He is also a governing board member of the APNNA. He is a member of the Editorial Board of the *Open Information Systems Journal*, Journal of NIP-LR. He has also served as Special Issue Guest Editor for *Neurocomputing* and *Journal of Computational Intelligent Research*. He has served as program and/or organizing member in international conferences and workshops, e.g. WWW, ACM MM, ICME, ICASSP, IJCANN, ICONIP, ICPR, etc. He has also served as reviewer for international conferences as well as journals, e.g. Information Fusion, SIGMOD, IEEE TCAS, TNN, TPAMI, TMM, TKDE, TSMC, etc.

His research interests include machine learning, multimedia processing, and web intelligence.

Michael R. Lyu received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1981; the M.S. degree in computer engineering from University of California, Santa Barbara, in 1985; and the Ph.D. degree in computer science from University of California, Los Angeles, in 1988. He is currently a Professor in the Computer Science and Engineering Department of the Chinese University of Hong Kong. He worked at the Jet Propulsion Laboratory as a Technical Staff Member from 1988 to 1990. From 1990 to 1992, he was with the Electrical and Computer Engineering Department at the University of Iowa as an Assistant Professor. From 1992 to 1995, he was a Member of the Technical Staff in the Applied Research Area of the Bell Communications Research, Bellcore. From 1995 to 1997, he was a research Member of the Technical Staff at Bell Laboratories, which was first part of AT&T, and later became part of Lucent Technologies.

Dr. Lyu has published over 250 refereed journal and conference papers in his research areas. He has been an associated editor of *IEEE Transactions on Reliability, IEEE Transactions on Knowledge and Data Engineering*, and *Journal of Information Science and Engineering*. He was elected to IEEE Fellow (2004) and AAAS Fellow (2007) for his contributions to software reliability engineering and software fault tolerance.

His research interests include software reliability engineering, distributed systems, fault-tolerant computing, web technologies, mobile networks, digital video library, multimedia processing, and video searching and delivery.