Statistica Sinica **24** (2014), 493-514
doi:http://dx.doi.org/10.5705/ss.2012.220

LINEAR DISCRIMINANT ANALYSIS OF CHARACTER SEQUENCES USING OCCURRENCES OF WORDS

Subhajit Dutta[1], Probal Chaudhuri[2] and Anil K. Ghosh[2]

[1]*King Abdullah University of Science and Technology and*
[2]*Indian Statistical Institute*

Abstract: Classification of character sequences, where the characters come from a finite set, arises in disciplines such as molecular biology and computer science. For discriminant analysis of such character sequences, the Bayes classifier based on Markov models turns out to have class boundaries defined by linear functions of occurrences of words in the sequences. It is shown that for such classifiers based on Markov models with unknown orders, if the orders are estimated from the data using cross-validation, the resulting classifier has Bayes risk consistency under suitable conditions. Even when Markov models are not valid for the data, we develop methods for constructing classifiers based on linear functions of occurrences of words, where the word length is chosen by cross-validation. Such linear classifiers are constructed using ideas of support vector machines, regression depth, and distance weighted discrimination. We show that classifiers with linear class boundaries have certain optimal properties in terms of their asymptotic misclassification probabilities. The performance of these classifiers is demonstrated in various simulated and benchmark data sets.

Key words and phrases: Bayes classifier, Markov and hidden Markov models, misclassification probability, order of a Markov model, V-fold cross-validation, word frequency.

1. Introduction

Discriminant analysis problems involving character sequences, where the characters come from a finite set, arise in many scientific disciplines, and we begin with some examples. Consider an example related to different segments of the DNA sequence of the organism *Escherichia coli* (see, e.g., Harley and Reynolds (1987)). In such a sequence, segments that code for proteins are called genes. The *promoter region* located near a gene facilitates the transcription of that gene. An *intron* is a segment within a gene that is non-coding and not translated into a protein. *Exons* are parts of the gene that code for amino acids, which are building blocks for a protein. In the process of protein synthesis, genes are spliced at different sites, the *splice sites*, into *introns* and *exons. Exons* are retained after gene splicing and used to form the messenger-RNA sequence, which is a sequence of codons each corresponding to a specific amino acid.

The messenger-RNA carries the information about the basic building blocks required for the synthesis of a protein. Associated with this example, we have two data sets both of which are available at the UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml`). For the first, the task is to predict whether a DNA sequence is a member (or not) of a class of sequences with biological *promoter activity* (see Harley and Reynolds (1987)). There are 53 sequences in each of the two classes (*promoters* and *non-promoters*), and each sequence consists of 57 nucleotides from the set {A=Adenine, T=Thyamine, C=Cytosine, G=Guanine}. For the second data set, one is interested in identifying the boundaries between the *exons* and the *introns* (see Noordewier, Towell, and Shavlik (1991)). Here, given a DNA sequence, we want to predict whether it is an *exon to intron* site (EI), an *intron to exon* site (IE), or neither. In the UCI database, there are 767 sequences classified as EI sites, 768 sequences as IE sites, and 1655 as neither. Each sequence consists of 60 nucleotides.

As a second example, we consider a classification problem involving the single proton emission computed tomography (SPECT) images obtained and studied by Kurgan et al. (2001). The data are also available at the UCI database. SPECT imaging is used as a diagnostic tool for myocardial perfusion, where a patient is first injected with a radioactive tracer, and then investigations are carried out, one under stress and another under rest. Each investigation yields a three-dimensional image that represents left ventricle muscle perfusion. Each of those two 3-D images is then displayed as three sets of 2-D images or slices. Out of those 6 slices, a total of 5 slices are selected. In each slice, there are 4 or 5 regions of interest (ROIs), and a total of 22 slices are selected for each mode of study. Then an image analysis algorithm (see Kurgan et al. (2001) for details) is used to extract 44 continuous features (a number that measures radioactive counts) representing perfusion in 22 ROIs under stress and rest conditions. Based on these features, 22 partial diagnoses, each recorded as 0 or 1, are generated using the CLIP3 algorithm (see, e.g., Cios, Wedding, and Liu (1997)). Based on these partial diagnoses, each patient is classified as *normal* or *abnormal.* The SPECT database contains 267 binary sequences, each having length 22, with specified training and test sets. While the training sample consists of 40 observations from each of the two classes, in the test set, there are 15 and 172 observations from the *normal* and the *abnormal* classes, respectively.

In each example, we have a supervised classification problem with a training data set of labeled sequences, and we need to develop a decision rule for assigning a future observation to one of several competing classes. Let S^d be the collection of all sequences of length d over a common finite state space S. In the training sample, the sequences in the jth $(1 \leq j \leq J)$ class are denoted as $\mathbf{x}_{j1}, \ldots, \mathbf{x}_{jn_j}$, where $\mathbf{x}_{ji} = (x_{ji1}, \ldots, x_{jid})$ is a sequence of length d, and $x_{jik} \in S$ for all $1 \leq i \leq$

n_j, $1 \leq j \leq J$ and $1 \leq k \leq d$. In the first example, S ={A, T, C, G}. For the first problem there, we have $d = 57$ and $J = 2$, while for the second, $d = 60$ and $J = 3$. In the second example, we have $S = \{0, 1\}$, $d = 22$, and $J = 2$.

In these examples, we can consider the sequences in the jth class as independent realizations of some stochastic sequence $\mathbf{X} = (X_1, \ldots, X_d)$ generated from the finite state space S according to a probability distribution G_j. It is well known that the Bayes rule $\delta_B(\mathbf{x})$ with minimum misclassification probability is given by

$$\delta_B(\mathbf{x}) = \delta_B(x_1, \ldots, x_d) = \arg \max_{1 \leq j \leq J} \pi_j P_{G_j}(X_1 = x_1, \ldots, X_d = x_d),$$

where π_j denotes the prior probability of the jth class $(1 \leq j \leq J)$. In practice, one would require estimates of $s^d - 1$ probabilities for each of the J classes to implement this rule, where s is the cardinality of S. However, in the examples discussed above, the size of the training sample corresponding to any class is much smaller than $s^d - 1$, and this is often the case in practice. Consequently, one needs to look for some parsimonious models involving fewer parameters for those $s^d - 1$ probabilities.

For any integer $k > 0$, we refer to the elements of S^k as k-words. For a fixed k, and a fixed k-word $(m_1, \ldots, m_k) \in S^k$, if $I(\cdot)$ is the indicator variable, then

$$f_{\mathbf{x}_{ji}}(m_1, \ldots, m_k) = \sum_{l=1}^{(d-k+1)} I(x_{jil} = m_1, x_{ji(l+1)} = m_2, \ldots, x_{ji(l+k-1)} = m_k)$$

is the frequency (count) for the k-word $(m_1, \ldots, m_k)$ in $\mathbf{x}_{ji}$. In molecular biology literature, several authors (see, e.g., Waterman (1995); Reinert, Schbath, and Waterman (2000); Basu, Burma, and Chaudhuri (2003) for a detailed review) have used such word frequencies, popularly known as oligonucleotide frequencies, to analyze DNA sequences. The frequencies for different k-words can be viewed as features of the character sequence $\mathbf{x}_{ji}$, and they can be used as variables to carry out discriminant analysis of the character sequences. We shall see in subsequent sections that classifiers based on occurrences of words have certain optimal properties in terms of their misclassification probabilities.

In the next section, we develop and study likelihood based classifiers when Markov models hold for the observed sequences. In Section 3, we investigate classifiers based on linear functions of occurrences of words when Markov models may not hold for the observed sequences. In Section 4, we carry out simulation studies to compare the performance of these classifiers. In Section 5, we analyze the data sets mentioned at the beginning of this section and investigate the performance of different classifiers when applied to those data sets. Section 6 contains some concluding remarks. All proofs and mathematical details are in the Appendix.

2. Linear Classifiers Based on Markov Models

Suppose that for $1 \leq j \leq J$, the probability distribution G_j corresponding to the jth class is Markov with order k_j. Assume that G_j has stationary transition probabilities (t.p.) $\{p_j(m_{k_j+1}|m_1, \ldots, m_{k_j});\, (m_1, \ldots, m_{k_j+1}) \in S^{k_j+1}\}$, and initial probabilities (i.p.) $\{q_j(m_1,\, \ldots, m_{k_j}); (m_1, \ldots, m_{k_j}) \in S^{k_j}\}$. Note that there are $s^{k_j+1} - 1$ free real parameters in the model, and we assume that each of them belongs to the open interval $(0, 1)$. Depending on the situation, $s^{k_j+1} - 1$ may be substantially smaller than $s^d - 1$. Let $\boldsymbol{\theta}_j(k_j)$ denote the vector of model parameters that include elements of the vector of i.p. and the matrix of t.p. Take $\mathbf{k} = (k_1, \ldots, k_J)$ and $\boldsymbol{\phi}(\mathbf{k}) = (\boldsymbol{\theta}_1(k_1), \ldots, \boldsymbol{\theta}_J(k_J))$. Then, the logarithm of the likelihood corresponding to the jth class for a sequence $\mathbf{x}$ under such a Markov model is given by

$$\begin{aligned}\log P_{G_j}(\mathbf{x}) = & \sum_{(m_1,\ldots,m_{k_j})\,\in\, S^{k_j}} I(x_1 = m_1, \ldots, x_{k_j} = m_{k_j}) \log q_j(m_1, \ldots, m_{k_j}) \\ & + \sum_{(m_1,\ldots,m_{k_j+1})\,\in\, S^{k_j+1}} f_{\mathbf{x}}(m_1, \ldots, m_{k_j+1}) \log p_j(m_{k_j+1}|m_1, \ldots, m_{k_j}),\end{aligned}$$

and the Bayes rule based on such Markov likelihoods is $\delta(\mathbf{x}, \boldsymbol{\phi}(\mathbf{k}), \mathbf{k}) = \arg\max_{1 \leq j \leq J} D_j(\mathbf{x})$, where $D_j(\mathbf{x}) = \log \pi_j + \log P_{G_j}(\mathbf{x})$.

Note that Markov models with different orders form a nested family in the sense that, for any $k \geq 0$, a Markov model of order k is also a Markov model of order k' with $k' > k$. It is easy to see that for $k \geq 1$, the frequency of a k-word in the sequence $\mathbf{x}$ is related to the initial occurrence of that k-word and the frequencies of $(k+1)$-words by the equation $f_{\mathbf{x}}(m_1, \ldots, m_k) = \sum_{m_0 \in S} f_{\mathbf{x}}(m_0, m_1, \ldots, m_k) + I(x_1 = m_1, \ldots, x_k = m_k)$. Hence, $D_j(\mathbf{x})$ is a linear function of such variables for any $k \geq k_j$. So, all these Markov likelihoods corresponding to J different classes can be viewed as Markov likelihoods with a common order $K = \max_{1 \leq j \leq J} k_j$, and we have the following result.

Fact 1 : *Let G_j be Markov with order k_j having stationary t.p. for $1 \leq j \leq J$. For the variables $I(x_1 = m_1, \ldots,\, x_K = m_K)$'s and $f_{\mathbf{x}}(m_1, \ldots, m_{K+1})$'s associated with the character sequence $\mathbf{x}$, the Bayes rule $\delta_B(\mathbf{x})$ has class boundaries defined by linear functions of these variables.*

Note that when the vector of i.p. $\{q_j(m_1, \ldots, m_K) : (m_1, \ldots, m_K) \in S^K\}$ is same for all $1 \leq j \leq J$, $D_j(\mathbf{x})$ can be taken to be a linear function of only the count variables $\{f_{\mathbf{x}}(m_1, \ldots,\, m_{K+1}) : (m_1, \ldots, m_{K+1}) \in S^{K+1}\}$. Due to the discrete nature of the probability distributions involved here, for an observation $\mathbf{x}$, the maximum value of $D_j(\mathbf{x})$ can occur for more than one value of j. In such a case, $\mathbf{x}$ can be classified into any of the classes for which $D_j(\mathbf{x})$ attains its highest value, and that does not affect the misclassification probability of the classifier. We adapt the convention of classifying $\mathbf{x}$ into the class corresponding to the minimum value of the index j in such a situation. Here, for a classifier $\delta(\mathbf{x})$, its misclassification probability is $\sum_{j=1}^{J} \pi_j P_{G_j}\{\delta(\mathbf{X}) \neq j\}$.

To build the classifier, one needs to estimate the orders of the Markov models and the associated model parameters that include the vectors of i.p. and the matrices of t.p. If G_j is specified to be Markov with order k_j and stationary t.p., it is easy to verify that the m.l.e. $\hat{\boldsymbol{\theta}}_j(k_j)$ of the model parameters based on i.i.d. observations $\mathbf{x}_{j1}, \ldots, \mathbf{x}_{jn_j} \sim G_j$ can be obtained as

$$\hat{q}_j(m_1, \ldots, m_{k_j}) = \frac{\sum_{i=1}^{n_j} I(x_{ji1} = m_1, \ldots, x_{jik_j} = m_{k_j})}{n_j}$$

and

$$\hat{p}_j(m_{k_j+1} | m_1, \ldots, m_{k_j}) = \frac{\sum_{i=1}^{n_j} f_{\mathbf{x}_{ji}}(m_1, \ldots, m_{k_j}, m_{k_j+1})}{\sum_{i=1}^{n_j} \sum_{m \in S} f_{\mathbf{x}_{ji}}(m_1, \ldots, m_{k_j}, m)}.$$

If $k_j = 0$, we interpret a 0th order stationary Markov sequence $\mathbf{X} = (X_1, \ldots, X_d)$ as an i.i.d. sequence, and in that case, the parameters associated with G_j are the probabilities $P_{G_j}(X_k = m) = p_j(m)$, where $m \in S$ and $1 \leq k \leq d$. So for $k_j = 0$, the m.l.e. $\hat{\boldsymbol{\theta}}_j(0)$ based on the training sample $\mathbf{x}_{j1}, \ldots, \mathbf{x}_{jn_j} \sim G_j$ is

$$\hat{p}_j(m) = \frac{\sum_{i=1}^{n_j} f_{\mathbf{x}_{ji}}(m)}{n_j d}.$$

Fact 2 : *Assume that under G_j, each word of size k_j has positive probability. Since $\hat{\boldsymbol{\theta}}_j(k_j)$ is based on simple averages of bounded i.i.d. random variables, it has an almost sure limit $\boldsymbol{\theta}_j^\star(k_j)$ (say) as $n_j \to \infty$. This is true irrespective of whether the specified Markov model, under which the m.l.e. $\hat{\boldsymbol{\theta}}_j(k_j)$ is computed, is correct or not. If the Markov model is valid for G_j, and $k_j^\star$ is its correct order, then $\boldsymbol{\theta}_j^\star(k_j^\star)$ is the same as the true vector of model parameters.*

In practice, one has to estimate the order of the Markov model for each class. In a classification problem, one can estimate the orders of the Markov models by minimizing the estimated misclassification probability. Here we estimate $\mathbf{k} = (k_1, \ldots, k_J)$, the vector of orders of the Markov models, by minimizing the V-fold cross-validation (CV) estimate (see, e.g., Hastie, Tibshirani, and Friedman (2009)) of the misclassification probability with an appropriate choice of V. For any fixed $(n_1, \ldots, n_J)$, consider a partition of the training sample into V folds.

Let $n_j(v)$ denote the number of observations from the jth class in the vth fold. We choose $n_j(v) = \lceil n_j/V \rceil$ or $\lfloor n_j/V \rfloor$ in an appropriate way to ensure that $\sum_v n_j(v) = n_j$ for all $1 \leq j \leq J$. For any fixed $\mathbf{k}$, let $\hat{\boldsymbol{\theta}}_j^{(-v)}(k_j)$ be the m.l.e. obtained from the training sample of the jth class when the vth $(1 \leq v \leq V)$ fold is left out. Then the V-fold cross-validation estimate of the misclassification probability is given by

$$\widehat{\Delta}_{CV}(\mathbf{k}) = \frac{1}{V}\sum_{v=1}^{V}\Big\{\sum_{j=1}^{J}\frac{\pi_j}{n_j(v)}\sum_{i\,:\,\mathbf{x}_{ji}\in\, v\text{th fold}} I\{\delta(\mathbf{x}_{ji},\hat{\boldsymbol{\phi}}^{(-v)}(\mathbf{k}),\mathbf{k}) \neq j\}\Big\},$$

and $\widehat{\mathbf{k}}_{CV} = (\hat{k}_1^{CV}, \ldots, \hat{k}_J^{CV})$ can be obtained by minimizing $\widehat{\Delta}_{CV}(\mathbf{k})$ with respect to $\mathbf{k} \in \{0, 1, \ldots, \bar{k}\}^J$, where $\bar{k}$ is some suitable upper bound for the orders of the Markov models. In practice, one considers several random splits of the training data, and the V-fold CV estimate of the misclassification probability is an average of the estimates obtained in different splits.

Denote the misclassification probability of the classifier $\delta(\mathbf{x}, \boldsymbol{\phi}(\mathbf{k}), \mathbf{k})$ by

$$\Delta(\boldsymbol{\phi}(\mathbf{k}), \mathbf{k}) = \sum_{j=1}^{J} \pi_j \; P_{G_j}\{\delta(\mathbf{X}, \boldsymbol{\phi}(\mathbf{k}), \mathbf{k}) \neq j\}.$$

If $G_1, \ldots, G_J$ happen to be Markov with $\mathbf{k}^\star = (k_1^\star, \ldots, k_J^\star)$ as the vector of true orders of the Markov models, and $\boldsymbol{\phi}^\star(\mathbf{k}^\star) = (\boldsymbol{\theta}_1^\star(k_1^\star), \ldots, \boldsymbol{\theta}_J^\star(k_J^\star))$ as the corresponding vector of true model parameters, then $\Delta(\boldsymbol{\phi}^\star(\mathbf{k}^\star), \mathbf{k}^\star) = \Delta_B$ (say) is the Bayes risk corresponding to the classification problem. Here, Δ_B is the misclassification probability of the Bayes classifier $\delta_B(\mathbf{x})$. Since a Markov model of order $\mathbf{k}^\star$ can always be viewed as a higher order Markov model with appropriate model parameters, we have $\Delta(\boldsymbol{\phi}^\star(\mathbf{k}), \mathbf{k}) = \Delta(\boldsymbol{\phi}^\star(\mathbf{k}^\star), \mathbf{k}^\star) = \Delta_B$ for all $\mathbf{k}^\star \leq \mathbf{k} \leq \bar{\mathbf{k}}$, where $\bar{\mathbf{k}} = (\bar{k}, \ldots, \bar{k})$. Here, we write $(k_1, \ldots, k_J) \geq (k_1', \ldots, k_J')$ if $k_j \geq k_j'$ for all $1 \leq j \leq J$. Next, take $\Delta(\mathbf{k}) = \inf_{\boldsymbol{\phi}(\mathbf{k})} \Delta(\boldsymbol{\phi}(\mathbf{k}), \mathbf{k})$, so that for all $\mathbf{k}^\star \leq \mathbf{k} \leq \bar{\mathbf{k}}$, $\Delta(\mathbf{k}) = \Delta_B$. The following theorem establishes Bayes risk consistency of the classifier based on Markov likelihoods under appropriate conditions when the orders of the Markov models are chosen using CV.

Theorem 1. *Suppose that $G_1, \ldots, G_J$ are Markov with stationary t.p. Assume that for $1 \leq j \leq J$, G_j has order $k_j^\star$, and all the elements of the vector of i.p. and the matrix of t.p. for G_j are positive. Then, if $\Delta(\mathbf{k}) > \Delta_B$ for any $\mathbf{k} \ngeq \mathbf{k}^\star$, the classifier constructed using Markov likelihoods and $\widehat{\mathbf{k}}_{CV}$ has a misclassification probability that converges to Δ_B as min$\{n_1, \ldots, n_J\} \to \infty$.*

The problem of estimating the unknown order of a Markov model can be viewed as a problem in model selection. In the past, several authors have formulated this as a multiple hypothesis testing problem and investigated likelihood ratio tests (see, e.g., Billingsley (1961a,b) for detailed reviews). As pointed out by Tong (1975), one of the subjective elements of this approach is the choice of the levels of significance associated with these tests. He proposed to use AIC (see Akaike (1974)) for selecting the optimal order of the Markov chain from a class of competing Markov models. However, Katz (1981) proved AIC to be asymptotically inconsistent. It is known that procedures based on likelihood ratio tests are also inconsistent if a fixed positive level of significance is used. Katz (1981) advocated in favour of using the Bayesian information criteria (BIC) (see Schwarz (1978)) to select the unknown order of a Markov chain based on an observed sequence. However, all these authors have studied the problem of estimation of the order of a Markov chain based on a single observed sequence, and none of them considered classification problems involving several observed sequences. One limitation of procedures based on AIC, BIC, or multiple testing based on likelihood ratio tests is the requirement of specified parametric models, like Markov models. This is one of our main reasons for considering CV as a method for selecting the appropriate word length when we are dealing with classification problems involving character sequences. As is evident in the next two sections, parametric models like Markov models may not be suitable for characters sequences in practice.

3. Other Linear Classifiers

Classifiers based on Markov models work well for synthetic data generated from Markov models, but can perform poorly in practice; classifiers based on linear functions of occurrences of words may still be useful. Apart from being computationally and conceptually simple, such a classifier provides useful geometric views of class separability and helps to detect important discriminating features. We consider classifiers based on linear functions of word occurrences when we do not assume Markov models for the underlying distributions. This can be viewed as a generalization since it is not difficult to construct examples in which the Bayes classifiers defined by the likelihood ratio have class boundaries described by linear functions of word occurrences.

Classifiers are constructed using certain optimization criteria that aim to choose an optimum linear function of occurrences of words to discriminate between observations from two classes. In the case of a two class problem and for a fixed $k \in \{1, \ldots, \bar{k}\}$, a linear classifier based on the initial occurrences of $(k-1)$-words and the frequencies of k-words is of the form

$$\delta^{\mathcal{L}}(\mathbf{x}, \boldsymbol{\gamma}(k), k) = \begin{cases} 1 & \text{if } \boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}}(k) \geq 0, \\ 2 & \text{otherwise,} \end{cases}$$

where, for $k > 1$, $\mathbf{T}_{\mathbf{X}}(k)$ denotes the $(s^{k-1} + s^k)$-dimensional column vector for which the first s^{k-1} components are the initial occurrences of all $(k-1)$-words, and the next s^k components are the frequencies of k-words. Here $\boldsymbol{\gamma}(k)$ is the $(s^{k-1}+s^k)$-dimensional row vector of real parameters. For $k = 1$, $\mathbf{T}_{\mathbf{X}}(k)$ involves the frequencies of 1-words only and, in that case, both $\mathbf{T}_{\mathbf{X}}(k)$ and $\boldsymbol{\gamma}(k)$ are s-dimensional. Note that the sum of the components of $\mathbf{T}_{\mathbf{X}}(k)$ is always a constant (d for $k = 1$ and $(d-k+2)$ for $k > 1$), and hence we do not include any constant term in the expression of $\delta^{\mathcal{L}}$.

For a fixed k, we estimate $\boldsymbol{\gamma}(k)$ from the training data using a criterion based on the idea of regression depth (RD) (see, e.g., Rousseeuw and Hubert (1999)), that has been used later for constructing classifiers by Christmann and Rousseeuw (2001), Christmann, Fischer, and Joachims (2002) and Ghosh and Chaudhuri (2005). It can be shown that weighted RD with suitable choice of weights is the average training sample misclassification rate given by

$$\widehat{\Delta}(\boldsymbol{\gamma}(k), k) = \frac{\pi_1}{n_1} \sum_{i=1}^{n_1} I[\boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}_{1i}}(k) < 0] + \frac{\pi_2}{n_2} \sum_{i=1}^{n_2} I[\boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}_{2i}}(k) \geq 0],$$

and we can get an estimate $\widehat{\boldsymbol{\gamma}}_R(k)$ of $\boldsymbol{\gamma}(k)$ by minimizing this. Note that without loss of generality, we can assume $\|\boldsymbol{\gamma}(k)\| \leq 1$.

We can choose the value of k by minimizing the V-fold cross-validation estimate of the misclassification probability as described in Section 2. For any fixed k, let $\hat{\boldsymbol{\gamma}}_R^{(-v)}(k)$ be an estimate of $\boldsymbol{\gamma}(k)$ obtained from the training sample when the vth $(1 \leq v \leq V)$ fold is left out. Then the V-fold cross-validation estimate of the misclassification probability is

$$\widehat{\Delta}_R(k) = \frac{1}{V} \sum_{v=1}^{V} \Big\{ \sum_{j=1}^{2} \frac{\pi_j}{n_j(v)} \sum_{i\,:\,\mathbf{x}_{ji} \in\, v\text{th fold}} I[\delta^{\mathcal{L}}(\mathbf{x}_{ji}, \hat{\boldsymbol{\gamma}}_R^{(-v)}(k), k) \neq j] \Big\}, \qquad (3.1)$$

and the estimate of k is $\hat{k}_R = \arg\min_{1 \leq k \leq \bar{k}} \widehat{\Delta}_R(k)$.

Take $\Delta(\boldsymbol{\gamma}(k), k) = E[\widehat{\Delta}(\boldsymbol{\gamma}(k), k)]$, and let $\Delta^{\mathcal{L}}(k) = \inf_{\boldsymbol{\gamma}(k)} \Delta(\boldsymbol{\gamma}(k), k)$ be the misclassification probability of the best linear classifier based on $\mathbf{T}_{\mathbf{X}}(k)$. Define k° to be the least value of k at which $\Delta^{\mathcal{L}}(k)$ $(1 \leq k \leq \bar{k})$ attains its minimum. In view of the nested structure of classifiers with class boundaries that are linear functions of $\mathbf{T}_{\mathbf{X}}(k)$, we have $\Delta^{\mathcal{L}}(k) = \Delta^{\mathcal{L}}(k^{\circ})$ for all $k^{\circ} \leq k \leq \bar{k}$. Also, we have $\Delta^{\mathcal{L}}(k) > \Delta^{\mathcal{L}}(k^{\circ})$ for any $1 \leq k < k^{\circ}$ in view of the definition of k°.

Theorem 2. *In a two class problem, the misclassification probability of the classifier based on RD and $\hat{k}_R$ converges to $\Delta^{\mathcal{L}}(k^{\circ})$ as $min\{n_1, n_2\} \to \infty$. When the class boundary of the Bayes classifier $\delta_B(\mathbf{x})$ is a linear function of $\mathbf{T}_{\mathbf{X}}(k^{\circ})$, the classifier constructed using RD and $\hat{k}_R$ achieves the Bayes risk Δ_B asymptotically.*

While the construction of the linear classifier based on RD does not require any parametric model for the observed sequences, the optimization problem that one needs to solve to construct such a linear classifier is computationally complex. Further, with the increase in the value of k, the number of k-words grows at an exponential rate, and this increases the computational complexity of the classifier based on RD. For the classifiers based on Markov models considered in Section 2, the computation of the maximum likelihood estimates of parameters is fairly straight-forward; classifiers based on RD are computationally far less attractive than those based on Markov models. Methods based on support vector machines (SVM) (see Vapnik (1998); Hastie, Tibshirani, and Friedman (2009)) and distance weighted discrimination (DWD) (see Marron, Todd, and Ahn (2007); Qiao et al. (2010)) are two well-known classification techniques available in the literature that are well equipped to deal with high-dimensional data sets, and it is appropriate to consider linear classifiers based on $\mathbf{T}_{\mathbf{X}}(k)$ using SVM and DWD as alternatives to RD.

In DWD, for a fixed k, the estimate $\widehat{\boldsymbol{\gamma}}_D(k)$ of $\boldsymbol{\gamma}(k)$ is obtained by minimizing (see Qiao et al. (2010))

$$\widehat{D}(\boldsymbol{\gamma}(k), k) = \frac{1}{n}\sum_{i=1}^{n_1} W[\boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}_{1i}}(k)] + \frac{1}{n}\sum_{i=1}^{n_2} W[-\boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}_{2i}}(k)]$$

subject to $\|\boldsymbol{\gamma}(k)\| \leq 1$. Here $n = n_1 + n_2$, and

$$W(z) = \begin{cases} 2\sqrt{C_0} - C_0 z & z \leq 1/\sqrt{C_0}\ , \\ 1/z & \text{otherwise.} \end{cases}$$

In SVM, one obtains the estimate $\widehat{\boldsymbol{\gamma}}_S(k)$ of $\boldsymbol{\gamma}(k)$ by minimizing (see Lin (2002))

$$\widehat{S}(\boldsymbol{\gamma}(k), k) = \frac{1}{n}\sum_{i=1}^{n_1}[1 - \boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}_{1i}}(k)]_+ + \frac{1}{n}\sum_{i=1}^{n_2}[1 + \boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}_{2i}}(k)]_+ + \lambda\|\boldsymbol{\gamma}(k)\|^2,$$

where λ is the regularization parameter, and $[x]_+ = \max\{0, x\}$. Here also, we can impose the constraint $\|\boldsymbol{\gamma}(k)\| \leq 1$. For estimation of the word length k, we use CV. Denote the estimates of k by $\hat{k}_D$ and $\hat{k}_S$ for DWD and SVM, respectively, obtained by minimizing appropriate V-fold cross-validation estimates of the misclassification probabilities. For DWD and SVM, the cross-validation estimates of misclassification probabilities, denoted as $\widehat{\Delta}_D(k)$ and $\widehat{\Delta}_S(k)$ can be defined in the same way as $\widehat{\Delta}_R(k)$ in (3.1) with $\hat{\boldsymbol{\gamma}}_R^{(-v)}(k)$ replaced by $\hat{\boldsymbol{\gamma}}_D^{(-v)}(k)$ and $\hat{\boldsymbol{\gamma}}_S^{(-v)}(k)$, respectively.

Theorem 3. *For a classification problem with two classes, suppose that for* $j = 1$ *and* 2, $n_j/n \to \pi_j$ *as* $n \to \infty$, *where* $0 < \pi_1, \pi_2 < 1$ *and* $\pi_1 + \pi_2 = 1$. *If the class* *boundary of the Bayes classifier $\delta_B(\mathbf{x})$ is a linear function of $\mathbf{T}_{\mathbf{X}}(k^\circ)$, then for the classifier constructed using DWD and $\hat{k}_D$, the misclassification probability converges to the Bayes risk Δ_B as $n \to \infty$. If $\lambda \to 0$, the same convergence result holds for the misclassification probability of the classifier based on SVM and $\hat{k}_S$.*

If there are J (> 2) competing classes, we can perform $\binom{J}{2}$ binary classifications taking one pair of classes at a time, then combine the results of these pairwise classifications using majority voting (see, e.g., Hastie, Tibshirani, and Friedman (2009)). In our numerical work we have used this procedure for problems involving more than two classes.

4. Data Analysis Based on Simulated Examples

We carried out some simulation studies based on Markov and hidden Markov models to compare the performance of different classifiers developed in Sections 2 and 3. Consider a three class example (MM1) involving stationary Markov models each with state space $\{0, 1\}$, where we take $p_j(0|0) = p_j(1|1) = \alpha_j$ and $p_j(1|0) = p_j(0|1) = 1 - \alpha_j$ for $j = 1, 2$ and 3. Then the t.p. matrices are symmetric and doubly stochastic, and we chose $\alpha_1 = 0.45$, $\alpha_2 = 0.55$ and $\alpha_3 = 0.25$. For the i.p., we took $q_j(0) = q_j(1) = 1/2$ for $j = 1, 2$ and 3, which is the stationary initial distribution for the t.p. matrix chosen for any class. The motivation for such choices is that if the values of α for the two classes are close (or far apart, respectively), then the Bayes risk for the problem is expected to be high (or low, respectively).

In a second example (MM2), there are two classes involving Markov models of different orders. The model corresponding to the first class is a first order Markov model with $p_1(0|0) = p_1(1|1) = 0.55$, $p_1(1|0) = p_1(0|1) = 0.45$ and $q_1(0) = q_1(1) = 1/2$. Note that this is same as the model corresponding to the second class in MM1. The model corresponding to the second class in MM2 is chosen to be a Markov model of order 6 with an entry of the t.p. matrix defined as $p_2(m|m_1, \ldots, m_6) = (1/6)\sum_{i=1}^{6} p(m|m_i)$, where $p(0|0) = p(1|1) = 0.45$, $p(1|0) = p(0|1) = 0.55$, and $q_2(m_1, \ldots, m_6) = (1/2)^6$ for any $(m_1, \ldots, m_6) \in \{0, 1\}^6$.

We considered a three class example (HMM3) involving data generated from hidden Markov models (HMM) (see, e.g., Rabiner (1989); Juang and Rabiner (1991)) with two hidden states (0 and 1) and two observed states (0 and 1). For the HMM corresponding to the three classes, we chose the underlying two-state stationary Markov chains in a similar way as in example MM1. For the t.p. and the i.p. of the hidden chain corresponding to the first class, we took $p_1(0|0) = p_1(1|1) = 0.55$, $p_1(1|0) = p_1(0|1) = 0.45$, and $q_1(0) = q_1(1) = 1/2$, and for the hidden chains corresponding to the other two classes, we took $p_j(0|0) =$

$p_j(1|1) = 0.25$, $p_j(1|0) = p_j(0|1) = 0.75$, and $q_j(0) = q_j(1) = 1/2$ for $j = 2$ and 3. Denote the emission probabilities (e.p.) of a HMM by $e(x|h)$. Note that $e(x|h)$ is the conditional probability of observing the state x given that the hidden state is h. The e.p. for the first two classes were taken to be $e_j(0|0) = 0.2$, $e_j(1|0) = 0.8$, $e_j(0|1) = 0.8$, and $e_j(1|1) = 0.2$, where $j = 1$ and 2, while the e.p. for the third class was $e_3(0|0) = 0.8$, $e_3(1|0) = 0.2$, $e_3(0|1) = 0.2$, and $e_3(1|1) = 0.8$. The first two classes differ only in their t.p. matrices, while the second and the third classes differ only in their e.p.

In each example, we generated sequences of length 100 and formed training and test samples with 200 and 300 observations, respectively. This procedure was repeated 100 times, and the average test set misclassification rates of the classifiers along with their standard errors over these 100 Monte Carlo simulations are reported in Table 1. Average misclassification rates of the Bayes classifiers, constructed assuming the model parameters to be known, are also reported to facilitate comparison. The value of $\bar{k}$, which is the upper bound for the sizes of the words or the orders of the Markov models, was chosen to be 10 for implementing CV.

We also studied the performance of some other classifiers. The classifier based on classification and regression trees (CART) is standard and can be used on sequence data with finite state spaces. We also considered the nearest neighbor (NN) classifier with Hamming distance, where the number of neighbors was chosen by minimizing a cross-validated estimate of the misclassification probability. We have also run SVM and DWD on our data treating each sequence as a vector of d binary feature variables (see Hsu, Chang, and Lin (2010), who suggested a similar implementation of SVM for sequence data). We call the resulting classifiers SVM* and DWD*, respectively, to distinguish them from our earlier implementation of SVM and DWD based on word frequencies. We also considered a kernelized version of SVM based on the Hamming distance (see, e.g., Sonnenburg, Rätsch, and Schölkopf (2005)), and we call it SVM**. All these classifiers lead to very high misclassification rates in our simulated examples (see Table 1). The advantage of using counts of the k-words as features instead of the original data is clear from the superior performance of SVM and DWD compared to SVM*, SVM**, and DWD*.

Among our linear classifiers based on occurrences of words, the overall performance of the classifier based on RD was better than DWD and SVM. In the last example involving HMM, we also tried the Bayes classifier based on the likelihood of HMM, where the well-known Baum-Welch algorithm (see, e.g., Baum et al. (1970); Hastie, Tibshirani, and Friedman (2009)) was used for parameter estimation using the training data. In addition to estimating the usual parameters of the HMM, we estimated the order (equivalently, the cardinality of the state

Table 1. Misclassification rates with standard error (within parantheses) of different classifiers on simulated data sets.

Data set	MM1	MM2	HMM3
Bayes	0.1192	0.2625	0.4245
Markov	0.1283 (0.0012)	0.2793 (0.0017)	0.4269 (0.0014)
RD	0.1262 (0.0009)	0.2912 (0.0019)	0.4308 (0.0015)
DWD	0.1267 (0.0011)	0.2983 (0.0019)	0.4367 (0.0014)
SVM	0.1286 (0.0011)	0.2941 (0.0019)	0.4324 (0.0013)
CART	0.3315 (0.0011)	0.5082 (0.0019)	0.6685 (0.0016)
NN	0.4466 (0.0016)	0.4823 (0.0020)	0.6182 (0.0018)
DWD*	0.6542 (0.0016)	0.5022 (0.0020)	0.6655 (0.0016)
SVM*	0.6617 (0.0017)	0.5012 (0.0020)	0.6646 (0.0015)
SVM**	0.5325 (0.0019)	0.4939 (0.0021)	0.6441 (0.0016)

Table 2. Empirical probability distribution of $\hat{k}_1^{CV}$, $\hat{k}_2^{CV}$ and $\hat{k}_3^{CV}$ in MM1.

k	1	2	3	4	5
$\hat{k}_1^{CV}$	0.525	0.225	0.17	0.075	0.005
$\hat{k}_2^{CV}$	0.56	0.18	0.15	0.08	0.03
$\hat{k}_3^{CV}$	0.675	0.185	0.070	0.060	0.010

Table 3. Empirical probability distributions of $\hat{k}_R$, $\hat{k}_D$ and $\hat{k}_S$ in MM1.

k	2	3	4	5
$\hat{k}_R$	0.673	0.177	0.110	0.040
$\hat{k}_D$	0.74	0.17	0.047	0.043
$\hat{k}_S$	0.653	0.197	0.090	0.060

space) of the underlying Markov chain from the observed data. In this example, all our linear classifiers based on occurrences of words yielded better misclassification rates than the model specific classifier constructed using the Baum-Welch algorithm.

Estimates of the priors of the competing classes have been taken to be proportional to the training sample sizes for different classes, and we have tried different values of V while implementing the V-fold CV, yielding very similar results. We have reported the results only for 2-fold CV repeated over 10 random splits of the training data. With three classes, for classifiers based on Markov likelihood, CV needs simultaneous minimization of $\widehat{\Delta}_{CV}(k_1, k_2, k_3)$, so we adopted the pairwise classification approach to make the procedure computationally easier. The results of pairwise classification were combined using majority voting (see Hastie, Tibshirani, and Friedman (2009)) to obtain the final classification.

Table 4. Empirical probability distribution of $\hat{k}_1^{CV}$ and $\hat{k}_2^{CV}$ in MM2.

k	1	2	3	4
$\hat{k}_1^{CV}$	0.48	0.25	0.15	0.12
k	6	7	8	9
$\hat{k}_2^{CV}$	0.52	0.23	0.20	0.05

Table 5. Empirical probability distributions of $\hat{k}_R$, $\hat{k}_D$ and $\hat{k}_S$ in MM2.

k	7	8	9	10
$\hat{k}_R$	0.55	0.30	0.08	0.07
$\hat{k}_D$	0.59	0.21	0.11	0.09
$\hat{k}_S$	0.66	0.19	0.10	0.05

For SVM, we used programs available in R (see Dimitriadou et al. (2011), and we used MATLAB codes available at `http://www.unc.edu/~marron/marron_software.html` for DWD. We used our own codes in R for RD, implemented using the algorithm discussed in Ghosh and Chaudhuri (2005). It is our experience that if we use only k-word frequencies ignoring initial $(k-1)$-word occurrences to construct linear classifiers based on RD, DWD, and SVM, there is no real change in the misclassification rates, and a substantial cost saving. We have used programs available in R for CART (see Ripley (2011)) as well as for SVM** (see Karatzoglou et al. (2004)), and for NN we used our own codes in R. All R codes written by us are available at `http://www.isical.ac.in/~tijahbus/Words.htm`.

Our classifiers are based on cross-validation (CV), and a relevant issue is how well CV estimates $\mathbf{k}^\star$ and k°. In order to address this issue, we computed the empirical probability distributions of the estimates of $\mathbf{k}^\star$ and k°, and those are reported in Tables 2−5. For the implementation of CV in the case of the classifier based on Markov models, the range of values for the order of the Markov model was taken to be $k = 0, 1, \ldots, 9$. For the linear classifiers discussed in Section 3, we carried out CV over the range of values $k = 1, 2, \ldots, 10$. The figures reported in Tables 2−5 indicate the presence of a small amount of bias due to over-estimation, which is not surprising in view of the nested nature of both the models and the classifiers considered here, and the fact that in CV we minimize an estimate of the overall misclassification probability of a classifier.

5. Results from the Analysis of Benchmark Data Sets

We analyzed three data sets taken from the UCI machine learning repository. Description of the Promoter Gene data, the Splice Junction data, and the SPECT Heart data was given in Section 1. In the Splice Junction data set, we omitted the

15 sequences that had missing characters. In the SPECT Heart data set, we have specific training and test sets. In the other two cases, we carried out our analysis based on training and test sets formed by randomly partitioning the data. This random partitioning was done 100 times to form 100 training and test sets. The sizes of the training and the test sets in each example are reported in Table 6, along with the average misclassification rates and the corresponding standard errors of different classifiers. For the SPECT Heart data set, the misclassification rates and their standard errors have been computed based on outputs of the classifier in the given test set.

In the Promoter Gene data set, our linear classifiers based on RD, SVM, and DWD worked well, and outperformed the classifiers based on Markov models as well as CART, NN, and SVM**. To apply SVM* and DWD*, we transformed the original state space {A,T,C,G} to $\{(1,0,0),(0,1,0),(0,0,1),\ (0,0,0)\}$ (see, e.g., Hsu, Chang, and Lin (2010)), and then ran SVM and DWD on the transformed data. Misclassification rates of SVM* and DWD* turned out to be twice as high as the misclassification rates of SVM and DWD; demonstrates the advantage of using classifiers based on the frequencies of words instead of classifiers based on the original data. For the Splice Junction data set, among our classifiers, SVM based on the counts of words yielded the lowest misclassification rate. However, with a misclassification rate of 0.0723, NN turns out to be the best among all the classifiers considered, while SVM* yielded the second best performance. In this example, it seems that the frequencies of words do not succeed in extracting adequate information from the original data.

On the website of the UCI repository, the best reported misclassification rate for the Promoter Gene data set is 0.0377 obtained using KBANN (Knowledge Based Artificial Neural Net), and a worst rate 0.1792 obtained using ID3 (Quinlan's decision-tree builder). We obtained a best misclassification rate of 0.0638 using the classifier based on DWD. For the Splice Junction data set, the best reported misclassification rate of 0.0632 was obtained using KBANN, and the worst, 0.2074, obtained using a nearest neighbor classifier. Our best misclassification rate was 0.2395, obtained using the classifier based on SVM with an adaptive choice of the word length. For both of these data sets, KBANN seems to be an effective classification procedure, developed as a hybrid of 'explanation based' and 'empirical learning' algorithms (see, e.g., Towell and Shavlik (1994)), and it uses data specific scientific knowledge. For the sake of comparison, we ran a standard version of the classifier based on artificial neural networks (ANN) with a single hidden layer on the same transformed data on which we ran SVM* and DWD*. Here, the number of hidden nodes was chosen by minimizing a cross-validated estimate of the misclassification probability, and we ran ANN using codes that are available in R (see Venables and Ripley (2002)). The misclassification rates of ANN for the Promoter Gene and the Splice Junction data sets

Table 6. Misclassification rates with standard error in brackets of different classifiers on real data sets.

Data set	Promoter Gene	Splice Junction	SPECT Heart
Seq. Length	57	60	22
Training set size	40+40	700+700+1400	40+40
Test set size	13+13	62+65+248	15+172
Markov	0.3538 (0.0059)	0.3558 (0.0023)	0.2139 (0.0307)
RD	0.0746 (0.0061)	0.3648 (0.0026)	0.2406 (0.0302)
DWD	0.0638 (0.0056)	0.2747 (0.0022)	0.2246 (0.0303)
SVM	0.0827 (0.0068)	0.2395 (0.0021)	0.1658 (0.0264)
CART	0.4712 (0.0055)	0.5000 (0.0000)	0.2948 (0.0366)
NN	0.2146 (0.0060)	0.0723 (0.0015)	0.2746 (0.0370)
DWD*	0.1315 (0.0026)	0.4137 (0.0019)	0.2163 (0.0340)
SVM*	0.1192 (0.0024)	0.0809 (0.0015)	0.2890 (0.0352)
SVM**	0.2200 (0.0028)	0.1776 (0.0023)	0.2380 (0.0311)

turned out to be 0.5723 and 0.5025, respectively, which are quite high compared to those obtained using KBANN as well as other methods proposed here.

In the SPECT Heart data set, the classifier based on SVM had the best misclassification rate of 0.1658. The second best rate was produced by a classifier based on Markov models. However, all other competing classifiers performed poorly for this data. Note that both versions of SVM that are run on the original data, namely SVM* and SVM**, gave misclassification rates quite high compared to that of SVM based on the frequencies of k-words with an adaptive choice of k. The website of the UCI repository reports a misclassification rate of 0.16 for the classifier based on CLIP3 algorithm (see, e.g., Cios, Wedding, and Liu (1997)) when applied to this data. CLIP3 is a hybrid algorithm that combines tree based and rule based procedures for classification.

For the Promoter Gene data and the Splice Junction data, we computed the empirical probability distributions of the estimated orders of the Markov models and the estimated lengths of the word frequencies. These are reported in Tables 7−10. For the implementation of CV for the classifier based on Markov models, the range of values for the order of the Markov model was taken to be $k = 0, \ldots, 4$, and for the linear classifiers discussed in Section 3, we carried out CV over the range of values $k = 1, \ldots, 5$. In the case of the SPECT Heart data, we had fixed training and test sets.

6. Concluding Remarks

We have proposed a classifier based on Markov models where the orders of the Markov models are chosen by a cross-validated estimate of the misclassification probability. We relaxed the Markov assumption and constructed a classifier based

Table 7. Empirical probability distribution of $\hat{k}_1^{CV}$ and $\hat{k}_2^{CV}$ in Promoter Gene data set.

k	1	2	3
$\hat{k}_1^{CV}$	0.47	0.44	0.09
$\hat{k}_2^{CV}$	0.35	0.39	0.26

Table 8. Empirical probability distributions of $\hat{k}_R$, $\hat{k}_D$ and $\hat{k}_S$ in Promoter Gene data set.

k	2	3	4	5
$\hat{k}_R$	0.15	0.22	0.47	0.16
$\hat{k}_D$	0.16	0.11	0.45	0.28
$\hat{k}_S$	0.20	0.16	0.42	0.22

Table 9. Empirical probability distribution of $\hat{k}_1^{CV}$, $\hat{k}_2^{CV}$ and $\hat{k}_3^{CV}$ in Splice Junction data set.

k	1	2	3
$\hat{k}_1^{CV}$	0.275	0.650	0.075
$\hat{k}_2^{CV}$	0	0.825	0.175
$\hat{k}_3^{CV}$	0.79	0.175	0.035

Table 10. Empirical probability distributions of $\hat{k}_R$, $\hat{k}_D$ and $\hat{k}_S$ in Splice Junction data set.

k	2	3	4	5
$\hat{k}_R$	0.05	0.774	0.153	0.023
$\hat{k}_D$	0.033	0.75	0.183	0.034
$\hat{k}_S$	0.04	0.807	0.14	0.013

on RD with an adaptive choice of the word length; the misclassification rate of the classifier based on it converges to the misclassification rate of the best linear classifier based on occurrences of k-words. The classifier based on RD is computationally quite expensive, so we developed linear classifiers based on DWD and SVM. We studied the asymptotic behavior of the misclassification probabilities of these classifiers, and proved their Bayes risk consistency under suitable conditions.

The overall performance of various linear classifiers appears to be quite satisfactory when applied to simulated and benchmark data sets. The use of frequencies of words as features extracted from the sequences lead to a significant improvement in the performance of several classifiers in some situations.

Acknowledgement

We would like to thank an associate editor and two referees who carefully read an earlier version of the paper and made several useful suggestions.

Appendix : Proofs and Mathematical Details

Proof of Theorem 1. We derive the limiting behavior of $\hat{\mathbf{k}}_{CV}$. For each fixed $\mathbf{k} \in \{0, \ldots, \bar{k}\}^J$ and $\boldsymbol{\phi}(\mathbf{k}) = (\boldsymbol{\theta}_1(k_1), \ldots, \boldsymbol{\theta}_J(k_J))$, we show that

$$\sup_{\boldsymbol{\phi}(\mathbf{k})} |\widehat{\Delta}(\boldsymbol{\phi}(\mathbf{k}), \mathbf{k}) - \Delta(\boldsymbol{\phi}(\mathbf{k}), \mathbf{k})| \overset{a.s.}{\to} 0 \text{ as } \min\{n_1, \ldots, n_J\} \to \infty,$$

where $\widehat{\Delta}(\boldsymbol{\phi}(\mathbf{k}), \mathbf{k}) = \sum_{j=1}^J \pi_j/n_j \sum_{i=1}^{n_j} I(\delta(\mathbf{x}_{ji}, \boldsymbol{\phi}(\mathbf{k}), \mathbf{k}) \neq j)$. It follows from Hoeffding's inequality (see Hoeffding (1963)) that, for all $1 \le j \le J$ and every $\epsilon > 0$, we have

$$P_{G_j}\left\{\Big|\frac{1}{n_j}\sum_{i=1}^{n_j} I(\delta(\mathbf{x}_{ji}, \boldsymbol{\phi}(\mathbf{k}), \mathbf{k}) \neq j) - P_{G_j}(\delta(\mathbf{X}, \boldsymbol{\phi}(\mathbf{k}), \mathbf{k}) \neq j)\Big| > \epsilon\right\} < 2e^{-2n_j\epsilon^2}. \tag{A.1}$$

When the components of the vector $\boldsymbol{\phi}(\mathbf{k})$ varies over the interval $(0, 1)$, the sets $\{\mathbf{x} : \delta(\mathbf{x}, \boldsymbol{\phi}(\mathbf{k}), \mathbf{k}) \neq j\}$ form a VC class with finite VC index. This is a consequence of Fact 1 and Examples 19.17 and 19.18 in van der Vaart (2000). Using (A.1), we get

$$P_{G_j}\left\{\sup_{\boldsymbol{\phi}(\mathbf{k})}\Big|\frac{1}{n_j}\sum_{i=1}^{n_j} I(\delta(\mathbf{x}_{ji}, \boldsymbol{\phi}(\mathbf{k}), \mathbf{k}) \neq j) - P_{G_j}(\delta(\mathbf{X}, \boldsymbol{\phi}(\mathbf{k}), \mathbf{k}) \neq j)\Big| > \epsilon\right\}$$
$$< 2n_j^{D(k_j)} e^{-2n_j\epsilon^2},$$

where $D(k_j)$ is a constant depending on k_j related to the VC index of the above-mentioned VC class. Since $\sum_{n_j} n_j^{D(k_j)} e^{-2n_j\epsilon^2} < \infty$ for any $\epsilon > 0$, we have $\sup_{\boldsymbol{\phi}(\mathbf{k})}\Big|1/n_j \sum_{i=1}^{n_j} I(\delta(\mathbf{x}_{ji}, \boldsymbol{\phi}(\mathbf{k}), \mathbf{k}) \neq j) - P_{G_j}(\delta(\mathbf{X}, \boldsymbol{\phi}(\mathbf{k}), \mathbf{k}) \neq j)\Big| \overset{a.s.}{\to} 0$ as $n_j \to \infty$ by the Borel-Cantelli lemma. Using the triangle inequality, we now get

$$\sup_{\boldsymbol{\phi}(\mathbf{k})} |\widehat{\Delta}(\boldsymbol{\phi}(\mathbf{k}), \mathbf{k}) - \Delta(\boldsymbol{\phi}(\mathbf{k}), \mathbf{k})| \overset{a.s.}{\to} 0 \text{ as } \min\{n_1, \ldots, n_J\} \to \infty. \tag{A.2}$$

Consequently, $|\widehat{\Delta}(\hat{\boldsymbol{\phi}}(\mathbf{k}), \mathbf{k}) - \Delta(\hat{\boldsymbol{\phi}}(\mathbf{k}), \mathbf{k})| \overset{a.s.}{\to} 0$ and $|\widehat{\Delta}(\boldsymbol{\phi}^\star(\mathbf{k}), \mathbf{k}) - \Delta(\boldsymbol{\phi}^\star(\mathbf{k}), \mathbf{k})| \overset{a.s.}{\to} 0$ as $\min\{n_1, \ldots, n_J\} \to \infty$, where $\hat{\boldsymbol{\phi}}(\mathbf{k}) = (\hat{\boldsymbol{\theta}}_1(k_1), \ldots, \hat{\boldsymbol{\theta}}_J(k_J))$. Using the assumption in the theorem and Fact 2, we have $\hat{\boldsymbol{\phi}}(\mathbf{k}) \overset{a.s.}{\to} \boldsymbol{\phi}^\star(\mathbf{k})$ as $\min\{n_1, \ldots, n_J\} \to \infty$. Note that for all $\mathbf{k} \ge \mathbf{k}^\star$, $\delta(\mathbf{x}, \boldsymbol{\phi}^\star(\mathbf{k}), \mathbf{k})$ is a likelihood based Bayes classifier, and this implies that $\Delta(\hat{\boldsymbol{\phi}}(\mathbf{k}), \mathbf{k}) \overset{a.s.}{\to} \Delta(\boldsymbol{\phi}^\star(\mathbf{k}), \mathbf{k}) = \Delta_B$ as $\min\{n_1, \ldots, n_J\} \to \infty$.

Rewrite the expression for the V-fold CV estimate of the misclassification rate as

$$\widehat{\Delta}_{CV}(\mathbf{k}) = \frac{1}{V}\sum_{v=1}^{V} \widehat{\Delta}_v(\hat{\boldsymbol{\phi}}^{(-v)}(\mathbf{k}), \mathbf{k}).$$

Note that $\hat{\boldsymbol{\phi}}^{(-v)}(\mathbf{k})$ is based on data points in the training sample excluding those in the vth fold, and $\widehat{\Delta}_v(\hat{\boldsymbol{\phi}}^{(-v)}(\mathbf{k}), \mathbf{k}) = \sum_{j=1}^{J} \pi_j/n_j(v) \sum_{i\,:\,\mathbf{x}_{ji}\in\, v\text{th fold}} I(\delta(\mathbf{x}_{ji}, \hat{\boldsymbol{\phi}}^{(-v)}(\mathbf{k}), \mathbf{k}) \neq j)$. Now, for any fixed v, we get $\left|\widehat{\Delta}_v(\hat{\boldsymbol{\phi}}^{(-v)}(\mathbf{k}), \mathbf{k}) - \Delta(\hat{\boldsymbol{\phi}}^{(-v)}(\mathbf{k}), \mathbf{k})\right| \overset{a.s.}{\to} 0$ as $\min\{n_1(v), \ldots, n_J(v)\} \to \infty$ as at (A.2).

Fix $1 \leq v \leq V$, $\mathbf{k} > \mathbf{k}^\star$ and let $\min\{n_1, \ldots, n_J\} \to \infty$, which implies $\min\{n_1(v), \ldots, n_J(v)\} \to \infty$. Then, in view of the assumption in the theorem and Fact 2, we have $\hat{\boldsymbol{\phi}}^{(-v)}(\mathbf{k}) \overset{a.s.}{\to} \boldsymbol{\phi}^\star(\mathbf{k})$. Since $\Delta(\boldsymbol{\phi}^\star(\mathbf{k}), \mathbf{k}) = \Delta_B =$ the Bayes risk, $\Delta(\hat{\boldsymbol{\phi}}^{(-v)}(\mathbf{k}), \mathbf{k}) \overset{a.s.}{\to} \Delta(\boldsymbol{\phi}^\star(\mathbf{k}), \mathbf{k})$, and consequently, $\widehat{\Delta}_v(\hat{\boldsymbol{\phi}}^{(-v)}(\mathbf{k}), \mathbf{k}) \overset{a.s.}{\to} \Delta(\boldsymbol{\phi}^\star(\mathbf{k}), \mathbf{k})$. So, for any $\mathbf{k} \geq \mathbf{k}^\star$, $\lim\,[\widehat{\Delta}_{CV}(\mathbf{k}) - \widehat{\Delta}_{CV}(\mathbf{k}^\star)] = \lim\,[\widehat{\Delta}_{CV}(\mathbf{k}) - \Delta_B] = 0$ almost surely as $\min\{n_1, \ldots, n_J\} \to \infty$. On the other hand for $\mathbf{k} \ngeq \mathbf{k}^\star$, one can verify that as $\min\{n_1, \ldots, n_J\} \to \infty$, $\liminf\,[\widehat{\Delta}_{CV}(\mathbf{k}) - \widehat{\Delta}_{CV}(\mathbf{k}^\star)] > 0$ almost surely in view of the assumption that $\Delta(\boldsymbol{\phi}^\star(\mathbf{k}), \mathbf{k}) > \Delta_B$. As $\hat{\mathbf{k}}_{CV}$ is obtained by minimizing $\widehat{\Delta}_{CV}(\mathbf{k})$ over $\mathbf{k}$, this implies that $P(\hat{\mathbf{k}}_{CV} \ngeq \mathbf{k}^\star) \to 0$ as $\min\{n_1, \ldots, n_J\} \to \infty$. If we consider B independent splits of the training data, $\widehat{\Delta}_{CV}(\mathbf{k})$ is an average obtained over these splits. Hence, all the arguments and the results here continue to hold.

The conditional misclassification probability, $\Delta(\hat{\boldsymbol{\phi}}(\hat{\mathbf{k}}_{CV}), \hat{\mathbf{k}}_{CV})$, given the training sample, can be expressed as a finite sum

$$\Delta(\hat{\boldsymbol{\phi}}(\hat{\mathbf{k}}_{CV}), \hat{\mathbf{k}}_{CV}) = \sum_{\mathbf{k}} \Delta(\hat{\boldsymbol{\phi}}(\mathbf{k}), \mathbf{k})\; I(\hat{\mathbf{k}}_{CV} = \mathbf{k}).$$

Since $\Delta(\hat{\boldsymbol{\phi}}(\mathbf{k}), \mathbf{k}) \overset{a.s.}{\to} \Delta_B$ for any $\mathbf{k} \geq \mathbf{k}^\star$, and $P(\hat{\mathbf{k}}_{CV} \geq \mathbf{k}^\star) \to 1$ as $\min\{n_1, \ldots, n_J\} \to \infty$, we must have $\Delta(\hat{\boldsymbol{\phi}}(\hat{\mathbf{k}}_{CV}), \hat{\mathbf{k}}_{CV}) \overset{a.s.}{\to} \Delta_B$ as $\min\{n_1, \ldots, n_J\} \to \infty$. As the misclassification probability of a classifier is the expected value of the conditional misclassification probability given the training sample, the proof of the theorem is now complete by a simple application of the Dominated Convergence Theorem.

Proof of Theorem 2. Note that when the components of the vector $\boldsymbol{\gamma}(k)$ vary over $\mathbb{R}$, the family of sets $\{\mathbf{x} : \boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}}(k) \geq 0\}$ form a VC class with finite VC index (see Examples 19.17 and 19.18 in van der Vaart (2000)). From the arguments in the proof of Theorem 3.1 in Ghosh and Chaudhuri (2005), we have

$$\sup_{\boldsymbol{\gamma}(k)} \left|\widehat{\Delta}(\boldsymbol{\gamma}(k), k) - \Delta(\boldsymbol{\gamma}(k), k)\right| \overset{a.s.}{\to} 0 \text{ as } \min\{n_1, n_2\} \to \infty.$$

Recall the expression for the V-fold CV estimate with a fixed V in the proof of Theorem 1, and define

$$\widehat{\Delta}_R(k) = \frac{1}{V}\sum_{v=1}^{V} \widehat{\Delta}_v(\hat{\boldsymbol{\gamma}}_R^{(-v)}(k), k).$$

Note that $\hat{\boldsymbol{\gamma}}_R^{(-v)}(k)$ is based on data points in the training sample excluding those in the vth fold, and $\widehat{\Delta}_v$ is constructed using the observations in the vth fold. Using arguments similar to those used in proving Theorem 3.1 in Ghosh and Chaudhuri (2005), one can show that for any fixed k, $\widehat{\Delta}_R(k) \overset{a.s.}{\to} \Delta^{\mathcal{L}}(k)$ as $\min\{n_1, n_2\} \to \infty$. Recall that $\Delta^{\mathcal{L}}(k) > \Delta(k^{\circ})$ for $k < k^{\circ}$, and $\Delta^{\mathcal{L}}(k) = \Delta(k^{\circ})$ for any $k^{\circ} \leq k \leq \bar{k}$. Since $\hat{k}_R = \arg\min_k \widehat{\Delta}_R(k)$, we now get $P(\hat{k}_R \geq k^{\circ}) \to 1$ as $\min\{n_1, n_2\} \to \infty$.

Conditional on the training sample, consider the decomposition

$$\Delta(\hat{\boldsymbol{\gamma}}_R(\hat{k}_R), \hat{k}_R) = \sum_k \Delta(\hat{\boldsymbol{\gamma}}_R(k), k) I(\hat{k}_R = k).$$

For any fixed k, the convergence of $\Delta(\hat{\boldsymbol{\gamma}}_R(k), k)$ to $\Delta^{\mathcal{L}}(k)$ follows from Ghosh and Chaudhuri (2005). Since $\Delta^{\mathcal{L}}(k) = \Delta^{\mathcal{L}}(k^{\circ})$ for all $k \geq k^{\circ}$ and $P(\hat{k}_R \geq k^{\circ}) \to 1$ as $\min\{n_1, n_2\} \to \infty$, the proof of the convergence of the misclassification probability of the classifier, based on RD and $\hat{k}_R$, to $\Delta^{\mathcal{L}}(k^{\circ})$ now follows from similar arguments as those at the end of the proof of Theorem 1.

Proof of Theorem 3. For DWD, using the fact that $W(\cdot)$ is a Lipschitz continuous function and Example 19.7 in van der Vaart (2000), it follows that as $\boldsymbol{\gamma}(k)$ varies over the set $\{\boldsymbol{\gamma}(k) : \|\boldsymbol{\gamma}(k)\| \leq 1\}$, the functions $W(\boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}}(k))$ form a VC class with finite VC index. Now, as $n_j/n \to \pi_j$ for $j = 1$ and 2, arguments using Hoeffding's inequality and Borel-Cantelli lemma in a similar way as in the proofs of (A.1) and (A.2) yield

$$\sup_{\boldsymbol{\gamma}(k)} \left| \widehat{D}(\boldsymbol{\gamma}(k), k) - D(\boldsymbol{\gamma}(k), k) \right| \overset{a.s.}{\to} 0, \tag{A.3}$$

where $D(\boldsymbol{\gamma}(k), k) = \pi_1 E_{G_1}[W(\boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}}(k))] + \pi_2 E_{G_2}[W(-\boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}}(k))]$.

Note that for SVM, the function $[1-z]_+$ is Lipschitz continuous in z. Hence, the facts that $\|\boldsymbol{\gamma}(k)\| \leq 1$ and each component of $\mathbf{T}_{\mathbf{X}}(k)$ is bounded above, imply that the functions $[1 - \boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}}(k)]_+$ form a VC class with finite VC index (see Example 19.7 in van der Vaart (2000)). Consequently, arguments similar to those used in the case of DWD lead to

$$\sup_{\boldsymbol{\gamma}(k)} \left| \widehat{S}(\boldsymbol{\gamma}(k), k) - S(\boldsymbol{\gamma}(k), k) \right|$$
$$\leq \sup_{\boldsymbol{\gamma}(k)} \left| \frac{1}{n}\sum_{i=1}^{n_1} [1 - \boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}_{1i}}(k)]_+ + \frac{1}{n}\sum_{i=1}^{n_2} [1 + \boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}_{2i}}(k)]_+ - S(\boldsymbol{\gamma}(k), k) \right| + |\lambda|$$

$\overset{a.s.}{\to} 0$ as $\lambda \to 0$ and $n \to \infty$, where $S(\boldsymbol{\gamma}(k), k) = \pi_1 E_{G_1}[1 - \boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}}(k)]_+ + \pi_2 E_{G_2}[1 + \boldsymbol{\gamma}(k)\mathbf{T}_{\mathbf{X}}(k)]_+$.

Under the assumption that the class boundary of the Bayes classifier $\delta_B(\mathbf{x})$ is a linear function of $\mathbf{T}_{\mathbf{X}}(k^\circ)$, we get $\Delta^{\mathcal{L}}(k^\circ) = \Delta_B$. Now, due to *Fisher consistency* of DWD and SVM (see, e.g., Qiao et al. (2010); Lin (2002)), the classifiers obtained by minimizing $D(\boldsymbol{\gamma}(k), k)$ and $S(\boldsymbol{\gamma}(k), k)$ over $\boldsymbol{\gamma}(k)$ will be a Bayes classifier for any $k^\circ \leq k \leq \bar{k}$, and it has the property that any observation on the class boundary of this classifier can be classified into any of the two classes without altering its misclassification probability. This implies that both $\Delta(\widehat{\boldsymbol{\gamma}}_D(k), k)$ (as well as $\Delta(\widehat{\boldsymbol{\gamma}}_D^{(-v)}(k), k)$ for a fixed v) and $\Delta(\widehat{\boldsymbol{\gamma}}_S(k), k)$ (as well as $\Delta(\widehat{\boldsymbol{\gamma}}_S^{(-v)}(k), k)$ for a fixed v) converge almost surely to Δ_B for $k^\circ \leq k \leq \bar{k}$ as $n \to \infty$. Note that here we need to use a sub-sequence argument because, unlike the maximum likelihood estimates for the parameters of the Markov models, we may not have direct convergence of $\widehat{\boldsymbol{\gamma}}_D(k)$ (or $\widehat{\boldsymbol{\gamma}}_D^{(-v)}(k)$) and $\widehat{\boldsymbol{\gamma}}_S(k)$ (or $\widehat{\boldsymbol{\gamma}}_S^{(-v)}(k)$). Note also that the limits of $\widehat{\boldsymbol{\gamma}}_D(k)$ (or $\widehat{\boldsymbol{\gamma}}_D^{(-v)}(k)$) and $\widehat{\boldsymbol{\gamma}}_S(k)$ (or $\widehat{\boldsymbol{\gamma}}_S^{(-v)}(k)$) along suitable sub-sequences are minimizers of the continuous functions $D(\boldsymbol{\gamma}(k), k)$ and $S(\boldsymbol{\gamma}(k), k)$, respectively. Hence, in view of the *Fisher consistency* of DWD and SVM, those limits yield Bayes classifiers with class boundaries that are linear functions of $\mathbf{T}_{\mathbf{X}}(k)$ for $k^\circ \leq k \leq \bar{k}$. Now, using arguments similar to those used in the proof of Theorem 1, one can verify that $\lim[\widehat{\Delta}_D(k) - \widehat{\Delta}_D(k^\circ)] = \lim[\widehat{\Delta}_D(k) - \Delta_B] = \lim[\widehat{\Delta}_S(k) - \widehat{\Delta}_S(k^\circ)] = \lim[\widehat{\Delta}_S(k) - \Delta_B] = 0$ almost surely for $k \geq k^\circ$ as $n \to \infty$. On the other hand, for $k < k^\circ$, since $\Delta^{\mathcal{L}}(k) > \Delta^{\mathcal{L}}(k^\circ) = \Delta_B$, similar arguments as those used in the proof of Theorem 1, imply that $\liminf[\widehat{\Delta}_D(k) - \widehat{\Delta}_D(k^\circ)] > 0$ and $\liminf[\widehat{\Delta}_S(k) - \widehat{\Delta}_S(k^\circ)] > 0$ almost surely as $n \to \infty$. Hence, $P(\hat{k}_D < k^\circ) \to 0$ and $P(\hat{k}_S < k^\circ) \to 0$ as $n \to \infty$.

Finally, given the training sample, we have the following decompositions into finite sums of the conditional misclassification probabilities

$$\Delta(\hat{\boldsymbol{\gamma}}_D(\hat{k}_D), \hat{k}_D) = \sum_k \Delta(\hat{\boldsymbol{\gamma}}_D(k), k) I(\hat{k}_D = k),$$

$$\Delta(\hat{\boldsymbol{\gamma}}_S(\hat{k}_S), \hat{k}_S) = \sum_k \Delta(\hat{\boldsymbol{\gamma}}_S(k), k) I(\hat{k}_S = k).$$

The proof of the theorem is now complete in view of arguments as at the proofs of Theorems 1 and 2.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Trans. Automat. Control* **19**, 716-723.

Basu, S., Burma, D. P. and Chaudhuri, P. (2003). Words in DNA sequences : some case studies based on their frequency statistics. *J. Math. Biol.* **46**, 479-503.

Baum, L. E., Petrie, T., Soules, G. and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.* **41**, 164-171.

Billingsley, P. (1961a). *Statistical Inference for Markov Processes.* Univ. of Chicago Press, U.S.A.

Billingsley, P. (1961b). Statistical methods in Markov chains. *Ann. Math. Statist.* **32**, 12-40.

Christmann, A. and Rousseeuw, P. (2001). Measuring overlap in binary regression. *Comput. Statist. Data Anal.* **37**, 65-75.

Christmann, A., Fischer, P. and Joachims, T. (2002). Comparison between various regression depth methods and the support vector machine to approximate the minimum number of misclassification. *Comput. Statist.* **17**, 273-287.

Cios, K. J., Wedding, D. K. and Liu, N. (1997). CLIP3 : Cover learning using integer programming. *Kybernetes* **26**, 513-536.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D. and Weingessel, A. (2011). e1071 : Misc functions of the department of statistics (e1071), TU Wien. R package version 1.5-27. *http://CRAN.R-project.org/package=e1071*

Ghosh, A. K. and Chaudhuri, P. (2005). On data depth and distribution free discriminant analysis using separating surfaces. *Bernoulli* **11**, 1-27.

Harley, C. and Reynolds, R. (1987). Analysis of *E. Coli* promoter sequences. *Nucleic Acids Research* **15**, 2343-2361.

Hastie, T., Tibshirani, R. and Friedman, J. H. (2009). *Elements of Statistical Learning Theory.* Springer, New York.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.* **58**, 13-30.

Hsu, C. W., Chang, C. C. and Lin, C. J. (2010). A practical guide to support vector classification. Available at `www.csie.ntu.edu.tw/~cjlin/papers/guide`.

Juang, B. H. and Rabiner, L. R. (1991). Hidden Markov models for speech recognition. *Technometrics* **33**, 251-272.

Karatzoglou, A., Smola, A., Hornik, K. and Zeileis A. (2004). kernlab - An S4 Package for kernel methods in R. *J. Statist. Soft.* **11**, 1-20.

Katz, R. W. (1981). On some criteria for estimating the order of a Markov chain. *Technometrics* **23**, 243-249.

Kurgan, L. A., Cios, K., Tadeusiewicz, R., Ogiela, M. and Goodenday, L. (2001). Knowledge discovery approach to automated cardiac SPECT diagnosis. *Art. Intell. Medicine* **23**, 149-169.

Lin, Y. (2002). Support vector machines and the Bayes rule in classification. *Data Mining and Knowledge Discovery* **6**, 259-275.

Marron, J. S., Todd, M. J. and Ahn, J. (2007). Distance weighted discrimination. *J. Amer. Statist. Assoc.* **102**, 1267-1271.

Noordewier, M., Towell, G. and Shavlik, J. (1991). Training knowledge-based neural networks to recognize genes in DNA sequences. *Adv. Neural Info. Proc. Sys.* **3**, Morgan Kaufmann.

Qiao, X., Zhang, H. H., Liu, Y., Todd, M. J. and Marron, J. S. (2010). Weighted distance weighted discrimination and its asymptotic properties. *J. Amer. Statist. Assoc.* **105**, 401-414.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **77**, 257-286.

Reinert, G., Schbath, S. and Waterman, M. S. (2000). Probabilistic and statistical properties of words : an overview. *J. Comput. Biol.* **7**, 1-46.

Ripley, B. (2011). tree: Classification and regression trees. R package version 1.0-29. `http://CRAN.R-project.org/package=tree`

Rousseeuw, P. J. and Hubert, M. (1999). Regression depth (with discussions). *J. Amer. Statist. Assoc.* **94**, 388-402.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.* **6**, 461-464.

Sonnenburg, S., Rätsch, G. and Schölkopf, B. (2005). Large scale genomic sequence SVM classifiers. *In Proceedings of the 22nd Int. Conf. Mach. Learn. (L. D. Raedt and S. Wrobel ed.)* 849-856, ACM Press, New York.

Tong, H. (1975). Determination of the order of a Markov chain by Akaike's information criterion. *J. Appl. Probab.* **12**, 488-497.

Towell, G. G. and Shavlik, J. W. (1994). Knowledge based artificial neural networks. *Artificial Intell.* **70**, 119-165.

van der Vaart, A. W. (2000). *Asymptotic Statistics.* Cambridge, United Kingdom.

Vapnik, V. (1998). *Statistical Learning Theory.* Wiley, New York.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S.* Springer, New York.

Waterman, M. S. (1995). *Introduction to Computational Biology.* Chapman and Hall, New York.

King Abdullah University of Science and Technology, CEMSE Division, Thuwal 23955-6900, Saudi Arabia.

E-mail: tijahbus@gmail.com

Theoretical Statistics and Mathematics Unit, Indian Statistical Institute, 203, B. T. Road, Kolkata 700108, India.

E-mail: probal@isical.ac.in

Theoretical Statistics and Mathematics Unit, Indian Statistical Institute, 203, B. T. Road, Kolkata 700108, India.

E-mail: akghosh@isical.ac.in

(Received July 2012; accepted February 2013)