Linear Discriminant Analysis of Character Sequences Using Occurrences of Words

Subhajit Dutta, Probal Chaudhuri and Anil K. Ghosh

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

Summary. In this article, we investigate certain methods for classification of character sequences, where the characters belong to a finite set. These methods are based on extraction of some appropriate features from the data, and they use suitable linear functions of these features to construct the classifiers. It is observed that in the case of linear classifiers based on Markov models with unknown orders, if the orders are estimated using BIC (Schwarz, 1978) or cross-validation, the resulting classifier has certain optimal asymptotic properties, and such classifiers have good finite sample performance in various simulated data sets. Even when Markov models are not valid for the data, it is observed that linear classifiers based on counts of words, when the word length is chosen adaptively by cross-validation, have good asymptotic properties, and they perform well in various simulated and real data sets. Further, those linear classifiers turn out to be comparable or slightly better than model specific nonlinear classifiers like those constructed using the Baum-Welch algorithm in hidden Markov models when data are generated from such parametric models.

Keywords: Bayes classifier, distance weighted discrimination, Markov and hidden Markov models, regression depth, support vector machines, V-fold cross-validation.

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. We first consider an example related to different segments of the genomic sequence (i.e., DNA sequence) of the organism *Escherichia coli*. In such a sequence, we have segments that code for proteins, and those 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 it is not translated into a protein. On the other hand, *exons* are the 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 (known as the *splice sites*) into *introns* and *exons* (see Figure 1). *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. Thus the messenger-RNA carries the information about the basic building blocks required for the synthesis of a protein.

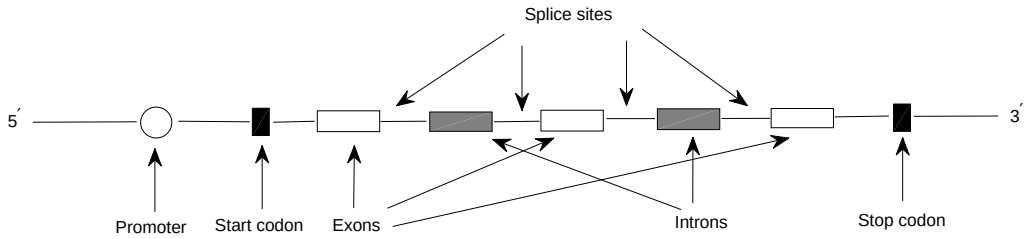

Figure 1. A typical segment of a DNA sequence

Associated with the above 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 set, our 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 EI (i.e., *exon to intron*) site, an IE (i.e., *intron to exon*) site 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.

In the next example, we consider a classification problem involving a game between two individuals called Connect-4. The data set is available at the UCI database, and it contains "all legal 8-ply positions in the game of connect-4 in which neither player has won yet, and in which the next move is not forced". Each board is a 6×7 rectangle, and in each position, one player puts a 'x' or the other player puts a 'o' or it is left blank (denoted by 'b'). So, each sequence involves the characters 'x', 'o' and 'b' and is of length 42. The game results in *a win* or a *a loss* or a *a draw*, and hence this is a classification problem involving three classes. In the data set, 44473 cases are classified as *wins*, 16635 cases as *losses* and 6449 cases as *draws*.

As a third 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 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 two investigations are carried out, one under stress and the other under rest. Each investigation yields a three-dimensional image, which 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. Further, 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 of which is recorded as 0 or 1, are generated using the CLIP3 algorithm (see, e.g., Kurgan, 2002). Based on these partial diagnoses, each of the patients is classified into two classes, *normal* and *abnormal*. The SPECT database contains 267 binary sequences each with length 22. 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 of the preceding three examples, we have a supervised classification problem with a training data set of labeled sequences, and we need to develop a decision rule $\delta(\mathbf{x})$ for assigning a future observation $\mathbf{x} = (x_1, \cdots, x_d) \in S^d$ to one of J competing classes. Here S^d is the collection of all sequences of length d over a common finite state space S. In the training sample, the sequences in the i-th ($1 \leq i \leq J$) class will be denoted as $\mathbf{x}_{i1}, \cdots, \mathbf{x}_{in_i}$, where $\mathbf{x}_{ij} = (x_{ij1}, \ldots, x_{ijd})$ is a sequence of length d, and $x_{ijk} \in S$ for all $i \leq 1 \leq J$, $j \leq 1 \leq n_i$ and $1 \leq k \leq d$. For the first example, $S =$\{A, T, C, G\}. In the first problem in that example, we have $d = 57$, while in the second problem, $d = 60$. In the second example, $S = \{0, 1, 2\}$, and $d = 42$. In the third example, we have $S = \{0, 1\}$ and $d = 22$.

In any of the preceding examples, we can consider the sequences in the i-th class as independent realizations of some stochastic sequence $\mathbf{X} = (X_1, \cdots, X_d)$ generated from the finite state space S according to a probability distribution G_i. It is well known that the optimal Bayes rule $\delta_B(\mathbf{x})$ is given by

$$\delta_B(\mathbf{x}) = \delta_B(x_1, \cdots, x_d) = \arg \max_{1 \leq i \leq J} P_{G_i}(X_1 = x_1, \cdots, X_d = x_d).$$

The construction of the Bayes rule requires estimates of $s^d - 1$ probabilities for each of the

J classes, where s is the cardinality of S. However, in all 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, for those probabilities, one needs to look for some parsimonious models involving fewer parameters.

For any fixed positive integer k, we will refer to the elements of S^k as k-words, and the counts (frequencies) of these k-words can be used as some simple and natural features for discriminating among sequences that belong to different classes. In the literature of molecular biology, several authors (see, e.g., Waterman, 1995; Reinert, Schabath and Waterman, 2000; Basu, Burma and Chaudhuri, 2003 for a detailed review) have used such word frequencies (popularly known as oligonucleotide frequencies in the genetics literature) to analyze DNA sequences. For any fixed k, and a fixed k-word $(m_1, \cdots, m_k) \in S^k$, let $I(x_{ijl} = m_1, \cdots, x_{ij(l+k-1)} = m_k)$ be the indicator variable that denotes the occurrence (or non-occurrence) of the event $\{x_{ijl} = m_1, \cdots, x_{ij(l+k-1)} = m_k\}$ for $l = 1, \cdots, (d-k+1)$. So,

$$T_{\mathbf{x}_{ij}}(m_1, \cdots, m_k) = \sum_{l=1}^{d-k+1} I(x_{ijl} = m_1, x_{ij(l+1)} = m_2, \cdots, x_{ij(l+k-1)} = m_k)$$

gives the count for the k-word $(m_1, \cdots, m_k)$ in $\mathbf{x}_{ij}$. If the sequences in the i-th class are independent realizations of a stationary stochastic sequence $\mathbf{X} = (X_1, X_2, \cdots, X_d)$ with distribution G_i, $\{n_i(d-k+1)\}^{-1} \sum_{j=1}^{n_i} T_{\mathbf{x}_{ij}}(m_1, \cdots, m_k)$ is a natural estimate for the k-dimensional marginal probability $P_{G_i}(X_1 = m_1, \cdots, X_k = m_k)$.

The organization of rest of the article is as follows. In Section 2, we assume the sequences to satisfy Markov models and develop some likelihood based classifiers, which turn out to be linear classifiers based on word occurrences. In Section 3, we investigate some linear classifiers based on word counts when Markov models may not hold for the observed sequences. Orders of the Markov models in Section 2, and the lengths of the words in Section 3 are chosen based on the training data. Asymptotic properties of misclassification rates of these classifiers are also studied in these sections along with some finite sample simulation studies. In Section 4, we compare the performance of these classifiers using the real data sets described above. A brief summary of the main ideas and the results is given in Section 5 along with some concluding remarks. All proofs and mathematical details are presented in the Appendix.

2. Linear Classifiers based on Markov Models

Note that a k-th order Markov model with s states involves only $s^{k+1} - 1$ (which may be substantially smaller than $s^d - 1$) parameters. If an observation $\mathbf{x} = (x_1, \cdots, x_d)$ from

the i-th ($1 \le i \le J$) class follows a Markov model of order k_i with stationary transition probabilities, the likelihood associated with $\mathbf{x}$ is given by

$$q_i(x_1, \cdots, x_{k_i}) \prod_{l=1}^{d-k_i} p_i(x_{l+k_i} | x_l, \cdots, x_{l+k_i-1})$$

$$= \prod_{(m_1,\cdots,m_{k_i}) \in S^{k_i}} q_i(m_1, \cdots, m_{k_i})^{I(x_1=m_1,\cdots,x_{k_i}=m_{k_i})} \prod_{(m_1,\cdots,m_{k_i+1}) \in S^{k_i+1}} p_i(m_{k_i+1} | m_1, \cdots, m_{k_i})^{T_{\mathbf{X}}(m_1,\cdots,m_{k_i+1})},$$

where $\{p_i(m_{k_i+1}|m_1, \cdots, m_{k_i}); (m_1, \cdots, m_{k_i+1}) \in S^{k_i+1}\}$ are the elements of the transition probability matrix (t.p.m.), and $\{q_i(m_1, \cdots, m_{k_i}); (m_1, \cdots, m_{k_i}) \in S^{k_i}\}$ are the elements of the initial probability distribution (i.p.d.). So, the Bayes classifier based on these Markov likelihoods is given by

$$\delta_B(\mathbf{x}) = \arg \max_{1 \le i \le J} \{\log \pi_i + \sum_{(m_1,\cdots,m_{k_i}) \in S^{k_i}} I(x_1 = m_1, \cdots, x_{k_i} = m_{k_i}) \log q_i(m_1, \cdots, m_{k_i}) + \sum_{(m_1,\cdots,m_{k_i+1}) \in S^{k_i+1}} T_{\mathbf{X}}(m_1, \cdots, m_{k_i+1}) \log p_i(m_{k_i+1} | m_1, \cdots, m_{k_i})\}, \quad (*)$$

where π_i is the prior probability of the i-th class ($1 \le i \le J$). Note that Markov models with different orders form a nested family in the sense that for any $k \ge 0$, a Markov model of order k is also a Markov model of order $k' \ge k$. Hence, all these Markov models corresponding to different classes can be viewed as Markov models of a common order $K = \max_{1 \le i \le J} k_i$, and we have the following result.

Result 1 : *Assume that the G_is are Markov with stationary transition probabilities. If the initial probability vector $\{q_i(m_1, \cdots, m_K) : (m_1, \cdots, m_{K+1}) \in S^{K+1}\}$ is the same for all $1 \le i \le J$, the Bayes classifier will be a linear function of the variables $T_{\mathbf{X}}(m_1, \cdots, m_{K+1})$, where $(m_1, \cdots, m_{K+1}) \in S^{K+1}$.*

Recall now from classical discriminant analysis of continuous multivariate data that the Bayes classifier is actually a linear classifier when the probability models for different classes are Gaussian with a common dispersion matrix. The preceding result can be viewed as an analogue of that in the context of discriminant analysis of character sequences satisfying Markov models. Apart from being computationally and conceptually simple, the linear classifier in $(*)$ provides good lower dimensional views of class separability and helps to detect important discriminating features.

If there are J competing Markov models, in order to build the classifier, one needs to estimate the orders of these Markov models $k_1, \ldots, k_J$ and their corresponding parameters $\boldsymbol{\theta}_{1k_1}, \boldsymbol{\theta}_{2k_2}, \ldots, \boldsymbol{\theta}_{Jk_J}$ (say) that include the elements of the i.p.d. and the t.p.m. Note that the elements of $\boldsymbol{\theta}_{ik_i}$ consist of the $q_i(\cdot)$'s and the $p_i(\cdot)$'s, which are given by

$$q_i(m_1, \cdots, m_{k_i}) = P_{G_i}(X_1 = m_1, \cdots, X_{k_i} = m_{k_i}) \text{ and}$$

$p_i(m_{k_i+1}|m_1,\cdots,m_{k_i}) = P_{G_i}(X_1 = m_1,\cdots,X_{k_i+1} = m_{k_i+1})/P_{G_i}(X_1 = m_1,\cdots,X_{k_i} = m_{k_i})$.

If k_i is known, it is easy to verify that the m.l.e. $\hat{\boldsymbol{\theta}}_{ik_i,n_i}$ of $\boldsymbol{\theta}_{ik_i}$ based on the training sample corresponding to the i-th class can be obtained as follows.

$$\hat{q}_i(m_1,\cdots,m_{k_i}) = \sum_{j=1}^{n_i} T_{\mathbf{X}_{ij}}(m_1,\cdots,m_{k_i})/n_i(d-k_i+1) \text{ and}$$

$$\hat{p}_i(m_{k_i+1}|m_1,\cdots,m_{k_i}) = \sum_{j=1}^{n_i} T_{\mathbf{X}_{ij}}(m_1,\cdots,m_{k_i},m_{k_i+1})/\sum_{j=1}^{n_i}\sum_{m_0\in S} T_{\mathbf{X}_{ij}}(m_1,\cdots,m_{k_i},m_0).$$

An appealing property of $\hat{\boldsymbol{\theta}}_{ik_i,n_i}$ is that since this estimate is based on simple averages of bounded random variables, the following result holds in view of the strong law of large numbers and the central limit theorem.

Result 2 : *Suppose that G_i is Markov with stationary transition probabilities such that all words of length $\leq d$ have positive probability. Then we have the following.*
(a) $\hat{\boldsymbol{\theta}}_{ik_i,n_i} \overset{a.s.}{\to} \boldsymbol{\theta}_{ik_i}$ *as* $n_i \to \infty$.
(b) $\sqrt{n_i}(\hat{\boldsymbol{\theta}}_{ik_i,n_i} - \boldsymbol{\theta}_{ik_i}) \overset{D}{\to} N_{s^{k_i+1}-1}(0,\Sigma_{k_i})$, *for some* $(s^{k_i+1}-1)\times(s^{k_i+1}-1)$ *dispersion matrix* Σ_{k_i}.

In practice, one has to estimate the order of the Markov model k_i as well, and this leads us to a problem in model selection. In the past, several authors formulated this as a multiple hypothesis testing problem and investigated related likelihood ratio tests (see, e.g., Billingsley, 1961a; b for detailed reviews). But, 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. Hence, as an alternative to this approach, he proposed AIC (see Akaike, 1974) for selecting the optimal order of the Markov chain from a class of competing Markov models. However, this procedure based on AIC was based on heuristics only as pointed out by Katz (1984), who proved it to be asymptotically inconsistent. It will be appropriate to note here that procedures based on likelihood ratio tests will also be inconsistent if some fixed positive levels of significance are used. As an alternative to AIC, Katz (1984) proposed to choose the order of the Markov chain using BIC (see Schwarz, 1978) and proved its consistency. Following Katz (1984), here we estimate the order of the Markov chain using BIC as follows

$$k_{in_i}^{BIC} = \arg\min_{k_i}\left\{-2\ \ln\left\{\frac{L_i(\mathbf{x}_{i1},\ldots,\mathbf{x}_{in_i};\hat{\boldsymbol{\theta}}_{ik_i,n_i},k_i)}{L_i(\mathbf{x}_{i1},\ldots,\mathbf{x}_{in_i};\hat{\boldsymbol{\theta}}_{im,n_i},m)}\right\} - (\ln n_i)\ (s^m - s^{k_i})(s-1)\right\},$$

where $L_i(\mathbf{x}_{i1},\ldots,\mathbf{x}_{in_i};\hat{\boldsymbol{\theta}}_{ik_i,n_i},k_i)$ denotes the joint likelihood of the training sample corresponding to the i-th class, which is based on a k_i-th order Markov model ($0 \leq k_i \leq m$) and

evaluated at the m.l.e. $\hat{\boldsymbol{\theta}}_{ik_i,n_i}$. Here m is some suitable upper bound for the order of the Markov model.

However, in the context of classification, a more relevant and natural way to estimate the joint order of the Markov models is by minimizing an estimate of the misclassification rate. In this article, along with the above mentioned likelihood based method, we consider another method for model selection, where the k_i's are estimated by minimizing the cross-validation (CV) estimate of the misclassification rate. Compared to BIC, this choice of the joint order of the Markov models may lead to a better error rate. In cross-validation type methods (see, e.g., Devroye, Györfi and Lugosi, 1996), one chooses the values of $k_1, \ldots, k_J$ by minimizing an estimate of the misclassification rate. For any fixed $\mathbf{n} = (n_1, \ldots, n_J)$, let $[u_1, \cdots, u_V]$ be a partition of the training sample into V folds, where the numbers of observations from the i-th class $(1 \le i \le J)$ in $u_1, \cdots, u_V$ are as close as possible. For any fixed $\mathbf{k} = (k_1, \ldots, k_J)$ let $\hat{\boldsymbol{\theta}}^{-u_v}_{ik_i,n_i}$ be the m.l.e. of $\boldsymbol{\theta}_{ik_i}$ $(1 \le i \le J)$ obtained from the training sample of the i-th class, when the v-th $(1 \le v \le V)$ fold is left out. Now, consider the classifier $\delta(\mathbf{x}, \hat{\boldsymbol{\theta}}^{-u_v}_{\mathbf{k}}, \mathbf{k}) = \arg\max_{1 \le i \le J} \{\pi_i \, L_i(\mathbf{x}; \hat{\boldsymbol{\theta}}^{-u_v}_{ik_i,n_i}, k_i)\}$ based on Markov likelihoods for the J classes and for all $1 \le v \le V$. We consider B such partitions of the training sample, and hence the V-fold cross-validation estimate of the misclassification rate is given by

$$\Delta^{CV}_{\mathbf{n}}(\mathbf{k}) = \sum_{j=1}^{J} \frac{\pi_j}{n_j} \sum_{t=1}^{B} \sum_{v_t=1}^{V} \sum_{\mathbf{X}_{ij} \in u_{v_t}} I\{\delta(\mathbf{x}_{ij}, \hat{\boldsymbol{\theta}}^{-u_{v_t}}_{\mathbf{k}}, \mathbf{k}) \ne j\},$$

where $\hat{\boldsymbol{\theta}}^{-u_{v_t}}_{\mathbf{k}} = (\hat{\boldsymbol{\theta}}^{-u_{v_t}}_{1k_1,n_1}, \cdots, \hat{\boldsymbol{\theta}}^{-u_{v_t}}_{Jk_J,n_J})$ for all $1 \le t \le T$. The optimum orders of the Markov models $\mathbf{k}^{CV}_{\mathbf{n}} = (k^{CV}_{1n_1}, \ldots, k^{CV}_{Jn_J})$ are obtained by minimizing $\Delta^{CV}_{\mathbf{n}}(\mathbf{k})$ with respect to $\mathbf{k}$. Unlike BIC, CV needs simultaneous optimization w.r.t. the J variables $k_1, \cdots, k_J$, and this makes CV computationally quite expensive, especially when there are several competing classes. For $J > 2$, it would be computationally faster to adopt the pairwise classification approach. Also, for higher values of V, this method becomes computationally expensive, especially when the training sample is large.

Note that, if $G_1, \cdots, G_J$ are Markov with $\mathbf{k}_T = (k^T_1, \cdots, k^T_J)$ being the vector of true orders of the Markov models, and $\boldsymbol{\theta}^{\star}_{\mathbf{k}_T} = (\boldsymbol{\theta}^{\star}_{1k^T_1}, \ldots, \boldsymbol{\theta}^{\star}_{Jk^T_J})$ being the vector of true model parameters, the misclassification rate of the Bayes classifier $\delta_B(\mathbf{x})$ is given by

$$\Delta_B = \Delta(\boldsymbol{\theta}^{\star}_{\mathbf{k}_T}, \mathbf{k}_T) = \sum_{i=1}^{J} \pi_i \; P\{\arg\max_{1 \le j \le J} \pi_j \; L_j(\mathbf{X}; \boldsymbol{\theta}^{\star}_{jk^T_j}, k^T_j) \ne i \mid \mathbf{X} \text{ is from the } i\text{-th class}\}.$$

Since a Markov model of order $\mathbf{k}_T$ can always be viewed as a higher order Markov model with appropriate model parameters, we have $\Delta(\boldsymbol{\theta}^{\star}_{\mathbf{k}}, \mathbf{k}) = \Delta(\boldsymbol{\theta}^{\star}_{\mathbf{k}_T}, \mathbf{k}_T) = \Delta_B$ for all $\mathbf{k} \ge \mathbf{k}_T$,

where we write $\mathbf{k} \geq \mathbf{k}_T$ if $k_i \geq k_i^T$ for all $1 \leq i \leq J$. This leads to the following fact.

$(P1)$: *There exists a* $\mathbf{k}^\circ = (k_1^\circ, \cdots, k_J^\circ) \in \{0, \cdots, m\}^J$ *such that* $\mathbf{k}^\circ \leq \mathbf{k}_T$ *and*

$$\Delta(\boldsymbol{\theta}^\star_{\mathbf{k}}, \mathbf{k}) \begin{cases} = \Delta_B & \text{if } k_i^\circ \leq k_i \leq m \text{ for all } 1 \leq i \leq J, \\ > \Delta_B & \text{if } k_i < k_i^\circ \text{ for some } i. \end{cases}$$

Since $\mathbf{k}_T$ is the true joint order of the Markov chains, any $\mathbf{k} > \mathbf{k}_T$ leads to a Bayes classifier with appropriate choice of parameters. The following theorem shows that irrespective of whether we use BIC or CV, the error rate of the resulting classifier converges to the Bayes risk.

Theorem 1 : *Assume that* $G_1, \ldots, G_J$ *are Markov, and the conditions assumed in Results 1 and 2 hold. Under this assumption, a classifier based on the Markov model constructed using BIC or CV described above will have misclassification rate that converges to the Bayes risk as the training sample size goes to infinity.*

Now, we carry out simulation studies based on some Markov as well as non-Markov models to compare the performance of classifiers based on these two model selection procedures. In our simulation studies, for the V-fold CV method, we tried different values for V, and they yielded similar results. Here we report the results for $V = 2$ (with $B = 2$) only. We use eight examples in this section. The first three are based on Markov models, and the fourth one is based on variable order Markov models (VMM), which are finite mixtures of fixed order Markov models. The last four examples are based on hidden Markov models (HMM) (see, e.g., Rabiner, 1989). In each example, we generated sequences of length 100 and formed training and test samples with 200 and 300 observations from each class, respectively. This procedure was repeated 100 times, and the average test set misclassification rates of the classifiers based on Markov likelihood over these 100 simulations are reported in Table 1. Average error rates of the Bayes classifiers are also reported to facilitate comparison. Throughout this section, the priors of the competing classes are taken to be equal.

In the first example, we consider two collections of sequences generated from two 1st order Markov models each with state space $S = \{0, 1\}$. The transition probability matrices (t.p.m.) (*which we shall write row-wise for all matrices*) for the two classes are $(.52, .48)$, $(.48, .52)$ and $(.47, .53)$, $(.53, .47)$, respectively. Note that for a k-th order Markov model with s states, the t.p.m. has s^k rows and s columns. In our examples involving Markov models, the i.p.d. has taken to be a uniform distribution over s^k states. In Example-2, the first class is generated from a 1st order Markov model with t.p.m. $(.51, .49)$, $(.49, .51)$, and the other class is generated from a 5th order Markov model with t.p.m. $(.47, .53)$, $(.54, .46)$,

$(.40, .60)$, $(.50, .50)$, $(.72, .28)$, $(.45, .55)$, $(.40, .60)$, $(.35, .65)$, $(.71, .29)$, $(.50, .50)$, $(.40, .60)$, $(.91, .09)$, $(.25, .75)$, $(.28, .72)$, $(.40, .60)$, $(.35, .65)$, $(.70, .30)$, $(.54, .46)$, $(.40, .60)$, $(.09, .91)$, $(.72, .28)$, $(.45, .55)$, $(.40, .60)$, $(.35, .65)$, $(.16, .84)$, $(.50, .50)$, $(.43, .57)$, $(.91, .09)$, $(.25, .75)$, $(.81, .19)$, $(.40, .60)$, $(.35, .65)$. Example-3 deals with a four class problem with the i-th class being a Markov model of order i for $i = 1, 2, 3, 4$. The t.p.m. for different classes are : $(.52, .48)$, $(.48, .52)$ (for the 1st class)**;** $(.32, .68)$, $(.68, .32)$, $(.52, .48)$, $(.48, .52)$ (for the 2nd class)**;** $(.05, .95)$, $(.95, .05)$, $(.51, .49)$, $(.49, .51)$, $(.35, .65)$, $(.65, .35)$, $(.25, .75)$, $(.75, .25)$ (for the 3rd class) and $(.47, .53)$, $(.54, .46)$, $(.40, .60)$, $(.50, .50)$, $(.72, .28)$, $(.45, .55)$, $(.40, .60)$, $(.35, .65)$, $(.71, .29)$, $(.50, .50)$, $(.40, .60)$, $(.91, .09)$, $(.25, .75)$, $(.28, .72)$, $(.40, .60)$, $(.62, .38)$ (for the 4th class). In all of these three examples, BIC and CV had comparable error rates with BIC performing marginally better than CV. We observed the same phenomenon in Example-4, where there are two classes, and we first generate a Markov chain $(z_1, \cdots, z_{d-2})$ with state space $\{0, 1, 2\}$, i.p.d. $(1/3, 1/3, 1/3)$ and t.p.m $(.1, .2, .7)$, $(.2, .3, .5)$, $(.3, .4, .3)$ in order to make a random choice of the order of a Markov model. We then generate (X_1, X_2) from a uniform distribution over S^2. For $3 \leq r \leq d$, the r-th element of the sequence, X_r is generated from a Markov model of order z_{r-2}. For the 1st class, the sequences were generated from a mixture of three Markov models with t.p.m.s $(.6, .4)$ (when order is 0), $(.55, .45)$, $(.45, .55)$ (when order is 1) and $(.35, .65)$, $(.45, .55)$, $(.51, .49)$, $(.4, .6)$ (when order is 2), and for the second class they were generated from a mixture of Markov models with t.p.m.s $(.6, .4)$ (when order is 0), $(.42, .58)$, $(.58, .42)$ (when order is 1); and $(.52, .48)$, $(.55, .45)$ $(.35, .65)$, $(.51, .49)$ (when order is 2).

Next, we consider some examples, where data are generated from hidden Markov models (HMM). An HMM is characterized by two sets of parameters; the t.p.m. of the *hidden process* and the *emission probabilities* (e.p.) from the unobserved to the observed sequence (see, e.g., Fink, 2007). In Example-5, we consider two classes of binary sequences generated from two HMMs, with each of the hidden processes having binary state space. For the first class, the t.p.m. is $(.55, .45)$, $(.45, .55)$ and the e.p. are $(.6, .4)$, $(.4, .6)$, whereas those for class-2 are $(.45, .55)$, $(.55, .45)$ and $(.5, .5)$, $(.2, .8)$, respectively. Example-6 deals with two classes of binary sequences generated from two HMMs, where the hidden process in each case has three states $\{0, 1, 2\}$. The t.p.m. and the e.p. for the 1st class are $(.8, .1, .1)$, $(.1, .8, .1)$, $(.1, .1, .8)$ and $(.6, .4)$, $(.4, .6)$, $(.3, .7)$, respectively, whereas those for class-2 are $(.1, .1, .8)$, $(.1, .8, .1)$, $(.8, .1, .1)$ and $(.6, .4)$, $(.4, .6)$, $(.3, .7)$, respectively. In Example-7, we consider a four class problem combining Examples 5 and 6 stated above. In the last example (Example-8), we consider sequences with state space $\{0, 1, 2\}$, which are generated from an

Table 1. Error rates of classifiers constructed using likelihoods based on Markov models (MM).

Model →	Ex-1	Ex-2	Ex-3	Ex-4	Ex-5	Ex-6	Ex-7	Ex-8
Method ↓	(MM)	(MM)	(MM)	(VMM)	(HMM)	(HMM)	(HMM)	(HMM)
Bayes	0.3122	0.0165	0.0628	0.2458	0.0626	0.3238	0.4133	0.0397
BIC	0.3273	0.0186	0.0727	0.2752	0.0714	0.3430	0.4365	0.1493
CV	0.3287	0.0202	0.0782	0.2823	0.0712	0.3446	0.4443	0.1237

HMM. The hidden process for each class has three states, and the t.p.m. corresponding to the two classes are $(.8,.1,.1)$, $(.1,.6,.3)$, $(.1,.3,.6)$ and $(.1,.1,.8)$, $(.1,.8,.1)$, $(.8,.1,.1)$, respectively. The e.p. for the two classes are generated from mixture distributions of the form $.36*Bin(2,.6)+.48*Bin(2,.4)+.16*Bin(2,.3)$ and $.64*Bin(2,.6)+.32*Bin(2,.4)+.04*Bin(2,.3)$, where $Bin(n,p)$ denotes the binomial distribution with parameters n and p. In Examples 5-7, classifiers based on Markov likelihood had error rates close to the Bayes risk (see Table 1), but in Example-8 their performance was not satisfactory. In Example-8, BIC had somewhat worse performance than CV, but the error rate of CV was also much worse than the Bayes risk.

3. Other linear classifiers

In Section 2, we have seen that if the underlying sequences are not Markov, classifiers based on the Markov likelihood sometimes may not have satisfactory performance. For instance, in Example-8, the classifiers based on Markov likelihood had empirical error rates more than thrice the Bayes risk. This clearly shows the necessity to develop some better classifiers for the classification of sequence data where Markov models are inadequate. Note that the classifiers based on Markov likelihood uses initial word occurrences and counts of words of suitable length. So, it is natural to construct some new classifiers using these features. Here we use suitable linear functions of these features for this purpose. However, it is our empirical experience that if we use only word counts ignoring initial word occurrences, the error rates of the resulting classifiers, if not better, usually remains the same. Moreover, it cuts down the computing cost by reducing the dimension of the measurement (i.e., feature) vector, and this helps to overcome the problem of certain software and hardware limitations.

Let us now consider linear classifier based on word counts, where we do not assume any parametric structure (e.g., Markov model) for the underlying distributions. In some sense, this linear classification can be viewed as a generalization of the earlier method based on Markov likelihood. In the case of a two-class problem, for a fixed $k \in \{1, \cdots, m\}$, a linear

classifier based on counts of k-words is of the form

$$\delta(\mathbf{x}, \boldsymbol{\alpha}_k, \beta_k, k) = \begin{cases} 1 & \text{if } \boldsymbol{\alpha}_k' \mathbf{T}_{\mathbf{X},k} + \beta_k \geq 0 \\ 2 & \text{otherwise,} \end{cases}$$

where $\mathbf{T}_{\mathbf{X},k} = \{T_{\mathbf{X}}(m_1, \ldots, m_k) : (m_1, \cdots, m_k) \in S^k\}$ is a $s^k (= |S|^k)$ dimensional vector containing counts of all k-words in $\mathbf{x}$. Here $\boldsymbol{\alpha}_k$ is a s^k dimensional vector, and β_k is a scalar. To get a good linear classifier, we need to choose suitable values of k, $\boldsymbol{\alpha}_k$ and β_k using some appropriate criteria. For any fixed k, we can estimate $\boldsymbol{\alpha}_k$ and β_k from the training data using well-known methods like support vector machines (SVM) (see, e.g., Vapnik, 1998; Hastie, Tibshirani and Friedman, 2009), distance weighted discrimination (DWD) (see Marron, Todd and Ahn, 2007) or regression depth (RD) (see, e.g., Christmann, Fischer and Joachims, 2002; Ghosh and Chaudhuri, 2005). Here we consider counts of k-words as the measurement variables. Since we are not assuming any parametric model, BIC is not applicable any more in this case for choosing the value of k. In each of these three methods, first we choose the optimum value of k (denoted by $k_{\mathbf{n}}$) by minimizing the V-fold cross-validation estimate of the error rate, and then using this optimal value of k, we estimate $\boldsymbol{\alpha}$ and β (denoted by $\hat{\boldsymbol{\alpha}}_{k_{\mathbf{n}}}$ and $\hat{\beta}_{k_{\mathbf{n}}}$). The asymptotic performance of such classifiers is described in the following theorem.

Theorem 2 : *Consider a classification problem with two classes. The misclassification rate of the classifier based on RD converges to the misclassification rate of the best linear classifier based on counts of k-words with $k \leq m$. Further, if the Bayes discriminating surface is a linear function of frequencies of such k-words, then, for RD and DWD, the error rate of the resulting classifier converges to the optimal Bayes risk as $min\{n_1, n_2\} \to \infty$. If $C \to \infty$, where C is the parameter involved in the cost function used in SVM, the same convergence result (Bayes risk consistency) holds for SVM as well.*

Recall that the classifier based on RD is a linear classifier based on frequencies of words of an appropriate length, that is constructed by minimizing the error rate estimated from the training data. *If the underlying sequences are k-step Markov, and we have equal initial distributions, the best linear classifier based on counts of $(k+1)$-words is the Bayes classifier.* So, in that case, the error rates of all these linear classifiers with data driven choices of k converge to the Bayes risk. However, even when the Bayes classifier is not linear, the error rate of the RD based classifier converges to that of the best linear classifier. If there are $J(> 2)$ competing classes, we can perform $\binom{J}{2}$ binary classifications taking one pair of classes at a time and combine the results of these pairwise classifications using the well-known method of majority voting (see, e.g., Friedman, 1996; Hastie *et. al.*, 2009).

Table 2. Misclassification rates of linear classifiers based on word counts, and an adaptive version of the Baum-Welch algorithm.

Model →	Ex-1	Ex-2	Ex-3	Ex-4	Ex-5	Ex-6	Ex-7	Ex-8
Method ↓	(MM)	(MM)	(MM)	(VMM)	(HMM)	(HMM)	(HMM)	(HMM)
Bayes	0.3122	0.0165	0.0628	0.2458	0.0626	0.3238	0.4133	0.0397
RD	0.3284	0.0221	0.0748	0.2850	0.0712	0.3469	0.4406	0.0426
DWD	0.3270	0.0268	0.0785	0.2823	0.0720	0.3474	0.4498	0.0481
SVM	0.3364	0.0235	0.0772	0.2941	0.0723	0.3526	0.4493	0.0427
AdpBW	-	-	-	-	0.0722	0.3596	0.4663	0.0482

To study the performance of these linear classifiers, we considered the same data sets generated in Section 2 (Examples 1-8), and error rates are reported in Table 2. For SVM, we used programs available in R. For DWD, we used MATLAB codes available at `http://www.unc.edu/~marron/marron_software.html`, and we used our own codes in R for RD. Among the linear classifiers considered in this section, the overall performance of the classifier based on RD was marginally better than DWD and SVM. In Examples 1-4, when the underlying models are Markov or variable order Markov, these linear classifiers had error rates comparable to that of the classifier based on Markov likelihood after choosing the order by BIC. Even in Examples 5-7, all these linear classifiers had similar error rates. But, Example-8 clearly demonstrates the advantage of the classifiers based on RD, DWD and SVM. In this example, classifiers based on Markov model had error rates higher than 12%, but these three linear classifiers yielded misclassification rates close to the Bayes risk (3.97%). In Examples 5-8, we also tried an adaptive version of the well-known Baum-Welch algorithm (see, e.g., Baum *et. al.*, 1970; Welch, 2003) (AdpBW) for parameter estimation in HMMs based on training data and chose the order (or equivalently the cardinality of the state space) of the hidden process using BIC. In all these four examples, the linear classifiers based on RD, DWD and SVM yielded slightly better or comparable error rates than the model specific classifiers constructed using Baum-Welch algorithm. It will be appropriate to note here that the Baum-Welch algorithm is an iterative algorithm, and the initial starting point, the choice of which is not straight forward in practice, plays a major role in the performance of the algorithm. Besides the convergence of the algorithm is also an issue in practice, and when one needs to estimate the order of the hidden Markov process, it adds to the difficulty in the implementation of the method.

4. Results from the analysis of real data sets

In this section, we analyze four real data sets taken from the UCI machine learning repository. Description of each of the data sets, namely the **Promoter Gene data**, the **Splice Junction data**, the **Connect-4 data** and the **SPECT Heart data**, was given in Section 1. Only in the SPECT Heart data set, we have specific training and test sets. In all other cases, we carried out our analysis based on training and test sets formed by randomly partitioning the data. The sizes of the training and the test sets in each case are reported in Table 3 along with the average error rates (over all random partitions) of different classifiers. In the case of first two data sets, this random partitioning was done 100 times to form 100 training and test sets. However, in the case of Connect-4 data, which contains more than 67,000 observations, we considered a single partition only. For methods based on V-fold CV, we have reported the error rates for $V = 2$ only though other values of V were also tried, and the results did not turn out to be very different for different values of V. Throughout this section, we have taken the prior probabilities of different classes to be proportional to their training sample sizes.

In the Promoter Gene data set, linear classifiers based on RD, SVM and DWD worked well, and they outperformed all classifiers based on Markov models. In the website of the UCI repository, the best reported error rate for the Promoter Gene data set is 0.0377 using KBANN (Knowledge Based Artificial Neural Net) and the worst is 0.1792, based on a classifier using ID3 (Quinlan's decision-tree builder). We obtained the best error rate of 0.0638 using the DWD classifier. In the Splice Junction data set, we omitted the 15 sequences which had missing characters and worked with the rest of the sequences. For this data set, the website of the UCI repository reports the best error rate of 0.0632 using KBANN, and the worst reported error rate is 0.2074 using a nearest neighbor classifier. However, in this example, our best error rate was 0.2547, and it was obtained using the linear classifier based on SVM. For both of these data sets involving genomic sequences, KBANN seems to be an effective knowledge-based learning algorithm. However KBANN is specifically designed for these specific problems, and it lacks generalization.

In the Connect-4 data set, linear classifier based on SVM yielded the best error of 0.2512. The linear classifier based on Markov models with the orders chosen by BIC had an error rates of 31%. Using the nearest neighbor classifier based on Bayesian networks, Kontkanen *et. al.* (2000) reports error rates varying between 0.30 and 0.40 for this data set. Note that all our linear classifiers had error rates less than 35%.

In the SPECT Heart data set, Markov model with a cross-validated choice of the order

Table 3. Misclassification rates of different classifiers on real data sets.

Data set	Promoter Gene	Splice Junction	Connect-4	SPECT Heart
Seq. Length	57	60	42	22
Training	40+40	700+700+1400	22236+8317+3224	40+40
Test	13+13	62+65+248	22237+8318+3225	15+172
Markov+BIC	0.2038	0.3621	0.3109	0.2843
Markov+CV	0.1362	0.3620	0.3438	0.2501
RD	0.0746	0.4183	0.3416	0.2976
DWD	0.0638	0.3884	–	0.3114
SVM	0.0827	0.2547	0.2512	0.3221

of the Markov model led to the best error rate. The performance of the RD based linear classifier as well as that of the classifier based on Markov likelihood with orders of the Markov models chosen using BIC was also satisfactory. But all other classifiers had much higher error rates. In the website of the UCI repository, the classifier based on CLIP3 algorithm has been reported to yield an error rate of 0.16, but its performance is highly volatile and Kurgan (2002) reported error rates varying from 0.16 to 0.5, when this algorithm was applied to various other data sets.

5. Concluding remarks

This paper deals with classification of sequence data, where we use frequencies of some words of appropriate length as discriminating features and use linear functions of those frequencies to develop the classifiers. We have studied classifiers based on Markov likelihood, where the order of the Markov model is chosen by two different methods, namely BIC and CV. The procedure based on CV that we have investigated here is based on the misclassification error rate, while BIC is based on likelihoods associated with the models. In all our numerical studies, BIC had nearly comparable performance with CV. Further, BIC is computationally much more efficient than CV, and as it will be evident from the statement and the proof of Proposition 1 in the Appendix, $\mathbf{k}_{\mathbf{n}}^{BIC} = (k_{1n_1}^{BIC}, \ldots, k_{Jn_J}^{BIC})$ is a consistent estimate for $\mathbf{k}_T$ while $\mathbf{k}_{\mathbf{n}}^{CV}$ has a positive probability of asymptotically over-estimating $\mathbf{k}_T$. So, when we have a reasonably large training sample and Markov models are valid for the data, BIC selects a parsimonious model. On the other hand, one cannot use BIC in the absence of specified parametric models. We could, however, use CV for adaptive choice of the length of the words in the construction of linear classifiers in Section 3, where no parametric model

for the data was assumed.

The performance of different linear classifiers based on counts of words having length adaptively determined from the data turn out to be quite good in our simulation study irrespective of whether the Markov model is valid for the simulated data or not. The empirical misclassification rates for linear classifiers based on RD, SVM and DWD are found to be comparable, and the classifier based on RD performed slightly better than the other two linear classifiers in most of the simulated data sets. In Example-8, where all linear classifiers based on Markov model performed poorly, the classifier based on RD had error rate close to the Bayes risk. In all the simulated examples involving HMM, the linear classifier based on RD yielded the best error rates among the linear classifiers considered here, and the overall performance of all three linear classifiers was slightly better than that of the classifier constructed using the Baum-Welch algorithm, which is an algorithm specifically developed for HMMs. The performance of all three linear classifiers was comparable in the Promoter Gene and the SPECT Heart data. However, the performance of SVM was significantly better than all of the other classifiers considered in the Splice Junction data set and the Connect-4 data. In the case of Connect-4 data, the Matlab code for DWD could not be run due to hardware limitations.

Appendix : Proofs and Mathematical details

Before proving Theorem 1, we will first state and prove a proposition. Note that for a k-th order Markov model with state space $S = \{1, \cdots, s\}$, since $\sum_{(m_1,\ldots,m_k)\in S^k} q(m_1, \cdots, m_k) = 1$ and $\sum_{m_{k+1}\in S} p(m_{k+1}|\ m_1, \cdots, m_k) = 1$ for all $(m_1, \ldots, m_k) \in S^k$, $\boldsymbol{\theta}_k$ contains $(s^{k+1}-1)$ free parameters, and here we assume that it lies in $\Theta_k \subset [0,1]^{s^{k+1}-1}$, the parameter space.

Proposition 1 : *Recall the definition of* $\mathbf{k}^\circ$ *given in* $(P1)$*. If the sequences in the* i*-th class satisfy a Markov model with order* k_i^T*, for all* $1 \le i \le J$*, we have* $P(\mathbf{k}_{\mathbf{n}}^{BIC} = \mathbf{k}_T)$ *and* $P(\mathbf{k}_{\mathbf{n}}^{CV} \ge \mathbf{k}^\circ)$ *both converging to 1 as* $min\{n_1, \cdots, n_J\} \to \infty$ *(recall that* $\mathbf{k}^\circ \le \mathbf{k}_T$*).*

Proof of Proposition 1 : From Katz (1981), for any fixed class i $(1 \le i \le J)$, we have $k_{in_i}^{BIC} \xrightarrow{P} k_i^T$ as $n_i \to \infty$. This implies that $P(\mathbf{k}_{\mathbf{n}}^{BIC} = \mathbf{k}_T) \to 1$ as $\min\{n_1, \cdots, n_J\} \to \infty$.

We next derive the limiting behavior of $\mathbf{k}_{\mathbf{n}}^{CV}$. For each fixed $\mathbf{k} \in \{0, \cdots, m\}^J$ and $\boldsymbol{\theta}_{\mathbf{k}} \in \Theta_{\mathbf{k}}$, first we want to show that $\sup_{\boldsymbol{\theta}_{\mathbf{k}}} |\Delta_{\mathbf{n}}(\boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k}) - \Delta(\boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k})| \xrightarrow{a.s.} 0$ as $\min\{n_1, \cdots, n_J\} \to \infty$, where $\Delta_{\mathbf{n}}(\boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k}) = \sum_{i=1}^J \frac{\pi_i}{n_i} \sum_{j=1}^{n_i} I(\delta(\mathbf{X}_{ij}, \boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k}) \ne i)$ and $\Delta(\boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k}) = \sum_{i=1}^J \pi_i$ $P(\delta(\mathbf{X}, \boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k}) \ne i \mid \mathbf{X}$ is from the i-th class). Note that from Hoeffding's inequality (see

Hoeffding, 1963), for all $i = 1, \ldots, J$ and every $\epsilon > 0$, we have

$$P\left\{\Big|\frac{1}{n_i}\sum_{j=1}^{n_i} I(\delta(\mathbf{X}_{ij}, \boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k}) \neq i) - P(\delta(\mathbf{X}, \boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k}) \neq i)\Big| > \epsilon\right\} < 2e^{-2n_i\epsilon^2}$$

$$\Rightarrow P\left\{\Big|\Delta_{\mathbf{n}}(\boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k}) - \Delta(\boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k})\Big| > \epsilon\right\} < 2\sum_{i=1}^{J} e^{-2n_i\epsilon^2} \text{ (using triangle inequality).}$$

Now, we will use some arguments based on Vapnik-Chervonenkis (VC) index (dimension) of hyperplanes in Euclidean spaces. We know that the VC index of hyperplanes in $\mathbb{R}^d$ is $(d+1)$, and the hyperplanes form a VC class (for a discussion on VC index, VC class and related matters see, e.g., van der Vaart, 2000). Now, define

$$R^{\mathbf{k}}_{i_1,\cdots,i_p}(\boldsymbol{\theta}_{\mathbf{k}}) = \Big\{\mathbf{x} : \pi_{i_1}\ L_{i_1}(\mathbf{x}; \boldsymbol{\theta}_{i_1 k_{i_1}}, k_{i_1}) = \cdots = \pi_{i_p}\ L_{i_p}(\mathbf{x}; \boldsymbol{\theta}_{i_p k_{i_p}}, k_{i_p}) = \max_{1\leq l\leq J}\{\pi_l\ L_l(\mathbf{x}; \boldsymbol{\theta}_{lk_l}, k_l)\} \text{ and } \pi_j\ L_j(\mathbf{x}; \boldsymbol{\theta}_{jk_j}, k_j) < \max_{1\leq l\leq J}\{\pi_l\ L_l(\mathbf{x}; \boldsymbol{\theta}_{lk_l}, k_l)\} \text{ if } j \notin \{i_1, \cdots, i_p\}\Big\}$$

and $\mathbb{P}_p$ as the set of p distinct, ordered integers from $\{1, \cdots, J\}$. Also, define $P_{jl}(\boldsymbol{\theta}_{\mathbf{k}}) = \{\mathbf{x} : \pi_j\ L_j(\mathbf{x}; \boldsymbol{\theta}_{jk_j}, k_j) \geq \pi_l\ L_l(\mathbf{x}; \boldsymbol{\theta}_{lk_l}, k_l)\}$ and $E_{jl}(\boldsymbol{\theta}_{\mathbf{k}}) = P_{jl}(\boldsymbol{\theta}_{\mathbf{k}}) \cap P_{lj}(\boldsymbol{\theta}_{\mathbf{k}}) = \{\mathbf{x} : \pi_j\ L_j(\mathbf{x}; \boldsymbol{\theta}_{jk_j}, k_j) = \pi_l\ L_l(\mathbf{x}; \boldsymbol{\theta}_{lk_l}, k_l)\}$. For each fixed $(i_1, \cdots, i_p) \in \mathbb{P}_p$, one can check that $R^{\mathbf{k}}_{i_1,\cdots,i_p}(\boldsymbol{\theta}_{\mathbf{k}}) = \left\{\bigcap_{j\in(i_2,\cdots,i_p)} E_{i_1 j}(\boldsymbol{\theta}_{\mathbf{k}})\right\} \bigcap \left\{\bigcap_{j\notin(i_1,\cdots,i_p)} P_{i_1 j}(\boldsymbol{\theta}_{\mathbf{k}})\right\}$.

Recall that the loglikelihood based on a Markov model of order k is a linear function of initial occurrences of k-words and frequencies of $(k+1)$-words. Therefore, for any fixed $1 \leq j, l \leq J$ and as $\boldsymbol{\theta}_{\mathbf{k}}$ varies, the family of sets $P_{jl}(\boldsymbol{\theta}_{\mathbf{k}})$ and $E_{jl}(\boldsymbol{\theta}_{\mathbf{k}})$ form a VC class with a finite VC index. So, for any fixed $(i_1, \cdots, i_p)$, $R^{\mathbf{k}}_{i_1,\cdots,i_p}(\boldsymbol{\theta}_{\mathbf{k}})$ is also a member of a class with a finite VC index. Now, for the time being and the sake of simplicity, assume that in case of ties, we classify an observation to the class with the minimum index (i.e., to the class i_1 if there is a tie among p classes $i_1, \cdots, i_p$ with $i_1 < \cdots < i_p$). So, for the i-th class, we have

$$\{\mathbf{x} : \delta(\mathbf{x}, \boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k}) \neq i\} = \left\{\mathbf{x} : \mathbf{x} \in \bigcup_{i_1=1,\ i_1\neq i}^{J} R^{\mathbf{k}}_{i_1}(\boldsymbol{\theta}_{\mathbf{k}})\right\} \bigcup \left\{\mathbf{x} : \mathbf{x} \in \bigcup_{p=2}^{J} \bigcup_{(i_1,\cdots,i_p)\in\mathbb{P}_p, i\notin\{i_1,\cdots,i_p\}} R^{\mathbf{k}}_{i_1,\cdots,i_p}(\boldsymbol{\theta}_{\mathbf{k}})\right\} \bigcup \left\{\mathbf{x} : \mathbf{x} \in \bigcup_{p=2}^{J} \bigcup_{(i_1,\cdots,i_p)\in\mathbb{P}_p, i\in\{i_1,\cdots,i_p\}, i_1\neq i} R^{\mathbf{k}}_{i_1,\cdots,i_p}(\boldsymbol{\theta}_{\mathbf{k}})\right\}.$$

Note that the set $\{\mathbf{x} : \delta(\mathbf{x}, \boldsymbol{\theta}, \mathbf{k}) \neq i\}$ is also a member of a VC class with finite VC index, being a finite disjoint union of the $R^{\mathbf{k}}_{i_1,\cdots,i_p}(\boldsymbol{\theta}_{\mathbf{k}})$'s. Therefore, combining the above facts, we get

$$A_{\mathbf{n},\mathbf{k}}(\epsilon) = P\left\{\sup_{\boldsymbol{\theta}_{\mathbf{k}}}\Big|\Delta_{\mathbf{n}}(\boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k}) - \Delta_G(\boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k})\Big| > \epsilon\right\} < 2N^{D(\mathbf{k})}\sum_{i=1}^{J} e^{-2n_i\epsilon^2},$$

where $N = \sum_{i=1}^{J} n_i$, and $D(\mathbf{k})$ is a constant depending on $\mathbf{k}$. Since $\sum_{\mathbf{n}} A_{\mathbf{n},\mathbf{k}}(\epsilon) < \infty$ for any $\epsilon > 0$, from Borel-Cantelli lemma, it follows that $\sup_{\boldsymbol{\theta}_{\mathbf{k}}} |\Delta_{\mathbf{n}}(\boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k}) - \Delta(\boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k})| \overset{a.s.}{\to} 0$

as $\min\{n_1, \ldots, n_J\} \to \infty$. Note at this point that in the case of ties, interchanging the labels of the tied classes does not make any change in the error rate $\Delta(\boldsymbol{\theta}_{\mathbf{k}}, \mathbf{k})$. The above convergence result holds if instead of i_1, we choose any of the tied classes from $\{i_2, \ldots, i_p\}$. Hence, if we use randomization to break ties among the classes, the convergence result will continue to hold by a simple averaging over all such interchanges. Hence, $|\Delta_{\mathbf{n}}(\hat{\boldsymbol{\theta}}_{\mathbf{k}}, \mathbf{k}) - \Delta(\hat{\boldsymbol{\theta}}_{\mathbf{k}}, \mathbf{k})| \stackrel{a.s.}{\to} 0$, $|\Delta_{\mathbf{n}}(\boldsymbol{\theta}^{\star}_{\mathbf{k}}, \mathbf{k}) - \Delta(\boldsymbol{\theta}^{\star}_{\mathbf{k}}, \mathbf{k})| \stackrel{a.s.}{\to} 0$, where $\hat{\boldsymbol{\theta}}_{\mathbf{k}} = (\hat{\boldsymbol{\theta}}_{1k_1,n_1}, \ldots, \hat{\boldsymbol{\theta}}_{Jk_J,n_J})$ and consequently, $\Delta(\hat{\boldsymbol{\theta}}_{\mathbf{k}}, \mathbf{k}) \stackrel{a.s.}{\to} \Delta(\boldsymbol{\theta}^{\star}_{\mathbf{k}}, \mathbf{k})$ as $\min\{n_1, \ldots, n_J\} \to \infty$ in view of the continuity of Δ as a function of $\boldsymbol{\theta}_{\mathbf{k}}$. Also, for our V-fold CV estimate of the error rate as defined in Section 2, we have, using similar arguments (see Corollary 8.1 in Devroye *et. al.*, 1996) $\left|\Delta^{CV}_{\mathbf{n}}(\hat{\boldsymbol{\theta}}_{\mathbf{k}}, \mathbf{k}) - \Delta(\hat{\boldsymbol{\theta}}_{\mathbf{k}}, \mathbf{k})\right| \stackrel{P}{\to} 0$, and $\Delta^{CV}_{\mathbf{n}}(\hat{\boldsymbol{\theta}}_{\mathbf{k}}, \mathbf{k}) \stackrel{P}{\to} \Delta(\boldsymbol{\theta}^{\star}_{\mathbf{k}}, \mathbf{k})$. So, for any $\mathbf{k} < \mathbf{k}^{\circ}$, one can verify that $\Delta^{CV}_{\mathbf{n}}(\hat{\boldsymbol{\theta}}_{\mathbf{k}^{\circ}}, \mathbf{k}^{\circ}) - \Delta^{CV}_{\mathbf{n}}(\hat{\boldsymbol{\theta}}_{\mathbf{k}}, \mathbf{k}) \stackrel{P}{\to} \Delta(\boldsymbol{\theta}^{\star}_{\mathbf{k}^{\circ}}, \mathbf{k}^{\circ}) - \Delta(\boldsymbol{\theta}^{\star}_{\mathbf{k}}, \mathbf{k}) > 0$ (due to $(P1)$). This implies that $P(\mathbf{k}^{CV}_{\mathbf{n}} < \mathbf{k}^{\circ}) \to 0$ as $\min\{n_1, \cdots, n_J\} \to \infty$. □

Proof of Theorem 1 : For C = BIC or CV, $\Delta(\hat{\boldsymbol{\theta}}_{\mathbf{k}^C_{\mathbf{n}}}, \mathbf{k}^C_{\mathbf{n}})$, the conditional misclassification probability given the training sample can be expressed as

$$\Delta(\hat{\boldsymbol{\theta}}_{\mathbf{k}^C_{\mathbf{n}}}, \mathbf{k}^C_{\mathbf{n}}) = \sum_{\mathbf{k} \in \{0, \cdots, m\}^J} \Delta(\hat{\boldsymbol{\theta}}_{\mathbf{k}}, \mathbf{k})\ I(\mathbf{k}^C_{\mathbf{n}} = \mathbf{k}).$$

From $(P1)$ and the part of the proof of Proposition 1 concerning CV, we have for any $\mathbf{k} \geq \mathbf{k}^{\circ}$, $\Delta(\hat{\boldsymbol{\theta}}_{\mathbf{k}}, \mathbf{k}) \stackrel{a.s.}{\to} \Delta_B$ as $\min\{n_1, \cdots, n_J\} \to \infty$. From Proposition 1, we have $P(\mathbf{k}^C_{\mathbf{n}} \geq \mathbf{k}^{\circ}) \to 1$ as $\min\{n_1, \cdots, n_J\} \to \infty$. Using the decomposition of $\Delta(\hat{\boldsymbol{\theta}}_{\mathbf{k}^C_{\mathbf{n}}}, \mathbf{k}^C_{\mathbf{n}})$ into a finite sum as stated above, the fact that the misclassification probability of a classifier can be obtained as the expected value of the conditional misclassification probability conditioned on the training sample, and the fact that the misclassification probabilities are bounded by one, the proof is now complete by a simple application of dominated convergence theorem. □

Proof of Theorem 2 : Given any fixed $k, \boldsymbol{\alpha}_k, \beta_k$, define the empirical error rate of the linear classifier with class boundary $\boldsymbol{\alpha}'_k T_{\mathbf{X},k} + \beta_k = 0$, as

$$\Delta^{\mathcal{L}}_{\mathbf{n}}(\boldsymbol{\alpha}_k, \beta_k, k) = \frac{\pi_1}{n_1} \sum_{j=1}^{n_1} I\{\boldsymbol{\alpha}'_k \mathbf{T}_{\mathbf{X}_{1j},k} + \beta_k < 0\} + \frac{\pi_2}{n_2} \sum_{j=1}^{n_2} I\{\boldsymbol{\alpha}'_k \mathbf{T}_{\mathbf{X}_{2j},k} + \beta_k \geq 0\},$$

where n_1 and n_2 are training sample sizes of the two classes, and π_1 and π_2 are their respective prior probabilities. Its population analogue is then defined as follows

$$\Delta^{\mathcal{L}}(\boldsymbol{\alpha}_k, \beta_k, k) = \pi_1\ P\{\boldsymbol{\alpha}'_k \mathbf{T}_{\mathbf{X},k} + \beta_k < 0 \mid \mathbf{X} \sim G_1\} + \pi_2\ P\{\boldsymbol{\alpha}'_k \mathbf{T}_{\mathbf{X},k} + \beta_k \geq 0 \mid \mathbf{X} \sim G_2\}.$$

From Theorem 3.1 (pp. 21-22) in Ghosh and Chaudhuri (2005), we have

$$\sup_{\boldsymbol{\alpha}_k, \beta_k} \left|\Delta^{\mathcal{L}}_{\mathbf{n}}(\boldsymbol{\alpha}_k, \beta_k, k) - \Delta^{\mathcal{L}}(\boldsymbol{\alpha}_k, \beta_k, k)\right| \stackrel{a.s.}{\to} 0 \text{ as } \min\{n_1, n_2\} \to \infty.$$

For a two class problem, let Y_i (± 1) denote the label of the i-th observation. For DWD (see Qiao *et. al.*, 2010) we define $D(\boldsymbol{\alpha}_k, \beta_k, k) = E[V(Y(\boldsymbol{\alpha}_k' \mathbf{T}_{\mathbf{X},k} + \beta_k))]$ and denote its empirical version by $D_{\mathbf{n}}(\boldsymbol{\alpha}_k, \beta_k, k)$, where

$$V(yf) = \begin{cases} 2\sqrt{C} - C & yf \leq \frac{1}{\sqrt{C}}, \\ \frac{1}{yf} & \text{otherwise.} \end{cases}$$

Now, using Hoeffding's inequality and arguments based on VC index for hyperplanes, we have $\sup_{\boldsymbol{\alpha}_k, \beta_k} |D_{\mathbf{n}}(\boldsymbol{\alpha}_k, \beta_k, k) - D(\boldsymbol{\alpha}_k, \beta_k, k)| \stackrel{a.s.}{\to} 0$. Similarly, for SVM (see Lin, 2002) we define $S_{\mathbf{n}}(\boldsymbol{\alpha}_k, \beta_k, k) = \frac{1}{n_1+n_2} \sum_{i=1}^{n_1+n_2} [1 - Y_i(\boldsymbol{\alpha}_k' \mathbf{T}_{\mathbf{X}_i,k} + \beta_k)]_+ + \lambda \|\boldsymbol{\alpha}_k\|^2$ and $S(\boldsymbol{\alpha}_k, \beta_k, k) = E[(1 - Y(\boldsymbol{\alpha}_k' \mathbf{T}_{\mathbf{X},k} + \beta_k))_+]$. Now using the triangle inequality, and the fact that $\|\boldsymbol{\alpha}_k\| \leq M$, since each component of $\mathbf{T}_{\mathbf{X}_i,k}$'s is bounded above, arguments similar to that in the case of DWD, lead to

$$\begin{aligned} &\sup_{\boldsymbol{\alpha}_k, \beta_k} |S_{\mathbf{n}}(\boldsymbol{\alpha}_k, \beta_k, k) - S(\boldsymbol{\alpha}_k, \beta_k, k)| \\ &\leq \sup_{\boldsymbol{\alpha}_k, \beta_k} \left| \frac{1}{n_1 + n_2} \sum_{i=1}^{n_1+n_2} [1 - Y_i(\boldsymbol{\alpha}_k' \mathbf{T}_{\mathbf{X}_i,k} + \beta_k)]_+ - S(\boldsymbol{\alpha}_k, \beta_k, k) \right| + |\lambda| M^2 \stackrel{a.s.}{\to} 0, \end{aligned}$$

as $\lambda \to 0$ and $\min\{n_1, n_2\} \to \infty$. So, we have the uniform a.s. consistency results for both SVM and DWD.

Consider now a V-fold CV method, and let $[u_1, \ldots, u_V]$ be the V folds of the training sample. For any fixed k, after leaving out the v-th fold u_v, we estimate $(\boldsymbol{\alpha}_k, \beta_k)$ based on the remaining observations. For RD, let $(\hat{\boldsymbol{\alpha}}_{k,-v}^R, \hat{\beta}_{k,-v}^R)$ denote the estimate of $(\boldsymbol{\alpha}_k, \beta_k)$ obtained by minimizing the empirical error rate. For this fixed k, define the V-fold CV estimate of the error rate based on $(\hat{\boldsymbol{\alpha}}_{k,-v}^R, \hat{\beta}_{k,-v}^R)$ for all $1 \leq v \leq V$, as follows

$$\Delta_{\mathbf{n}}^{R,V}(k) = \frac{\pi_1}{n_1} \sum_{v=1}^{V} \sum_{\mathbf{X}_{1j} \in u_v} I\{\hat{\boldsymbol{\alpha}}_{k,-v}^{R\prime} \mathbf{T}_{\mathbf{X}_{1j},k} + \hat{\beta}_{k,-v}^R < 0\} + \frac{\pi_2}{n_2} \sum_{v=1}^{V} \sum_{\mathbf{X}_{2j} \in u_v} I\{\hat{\boldsymbol{\alpha}}_{k,-v}^{R\prime} \mathbf{T}_{\mathbf{X}_{2j},k} + \hat{\beta}_{k,-v}^R \geq 0\}.$$

In the same way we can define $\Delta_{\mathbf{n}}^{S,V}(k)$ and $\Delta_{\mathbf{n}}^{D,V}(k)$ to be the V-fold CV estimates of the error rate for the procedures based on SVM and DWD, respectively. Let $\Delta^{\mathcal{L}}(k)$ be the error rate of the best linear classifier based on k-words with $k \leq m$. Following arguments of Ghosh and Chaudhuri (2005), for any finite V, one can show that $\Delta_{\mathbf{n}}^{R,V}(k) \stackrel{a.s.}{\to} \Delta^{\mathcal{L}}(k)$ as $\min\{n_1, n_2\} \to \infty$ (see also Corollary 8.1 in Devroye *et. al.*, 1996). Define $k_{\mathbf{n}}^R = \arg\min_k \Delta_{\mathbf{n}}^{R,V}(k)$ and $S_{\mathcal{L}} = \{k : \Delta^{\mathcal{L}}(k) = \min_{1 \leq k \leq m} \Delta^{\mathcal{L}}(k)\}$ to be the collection of all optimal values of k. Using arguments similar to those used in the proof of Proposition 1, we get $P(k_{\mathbf{n}}^R \in S_{\mathcal{L}}) \to 1$ as $\min\{n_1, n_2\} \to \infty$. Now, conditional on the training sample, consider the decomposition $\Delta(\hat{\boldsymbol{\alpha}}_{k_{\mathbf{n}}^R}^R, \hat{\beta}_{k_{\mathbf{n}}^R}^R, k_{\mathbf{n}}^R) = \sum_{k=0}^{m} \Delta(\hat{\boldsymbol{\alpha}}_k^R, \hat{\beta}_k^R, k)\ I(k_{\mathbf{n}}^R = k)$. For any fixed k, the convergence of $\Delta(\hat{\boldsymbol{\alpha}}_k^R, \hat{\beta}_k^R, k)$ to $\Delta^{\mathcal{L}}(k)$ follows from Ghosh and Chaudhuri

(2005). Now, since $P(k_{\mathbf{n}}^R \in S_{\mathcal{L}}) \to 1$, the proof of the convergence of the error rate of the RD based classifier to the error rate of the best linear classifier based on k-words ($k \leq m$), follows from the argument as in the proof of Theorem 1.

On the other hand, in DWD and SVM, we estimate $(\boldsymbol{\alpha}_k, \beta_k)$ by optimizing the functions $D_{\mathbf{n}}(\cdot)$ and $S_{\mathbf{n}}(\cdot)$, respectively. So, under the assumption that the Bayes classifier is a linear function of k-words with $k \leq m$, and using the Fisher consistency of the linear classifiers based on SVM and DWD (see, e.g., Lin, 2002; Qiao *et. al.*, 2010), one can show that for $k \in S_{\mathcal{L}}^B$, $\Delta_{\mathbf{n}}^{S,V}(k) \overset{a.s.}{\to} \Delta_B$ and $\Delta_{\mathbf{n}}^{D,V}(k) \overset{a.s.}{\to} \Delta_B$ as $\min\{n_1, n_2\} \to \infty$, where $S_{\mathcal{L}}^B = \{k : \Delta^{\mathcal{L}}(k) = \Delta_B\}$. Note that $S_{\mathcal{L}}^B$ denotes the collection of those values of k such that a linear classifier based on frequencies of k-words achieves the Bayes risk (Δ_B). From the Fisher consistency (see, e.g., Lin, 2002; Qiao *et. al.*, 2010) of the linear classifiers based on SVM and DWD, we again have $P(k_{\mathbf{n}}^C \in S_{\mathcal{L}}^B) \to 1$ as $\min\{n_1, n_2\} \to \infty$, for $C =$ SVM and DWD. Now, considering the decomposition $\Delta(\hat{\boldsymbol{\alpha}}_{k_{\mathbf{n}}^C}^C, \hat{\beta}_{k_{\mathbf{n}}^C}^C, k_{\mathbf{n}}^C) = \sum_{k=0}^m \Delta(\hat{\boldsymbol{\alpha}}_k^C, \hat{\beta}_k^C, k) I(k_{\mathbf{n}}^C = k)$, for $C =$ SVM and DWD, we have proofs of the Bayes risk consistency for linear classifiers constructed using SVM and DWD. □

References

Akaike, H. (1974) A new look at the statistical model identification. *IEEE Trans. Aut. Cont.*, **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.* The University of Chicago Press, U.S.A.

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

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.

Devroye, L., Györfi, L. and Lugosi, G. (1996) *A Probabilistic Theory of Pattern Recognition.* Springer, New York.

Fink, G. A. (2007) *Markov Models for Pattern Recognition.* Springer, New York.

Friedman, J.H. (1996) Another approach to polychotomous classification. *Tech. Rep., Dept. of Stat., Stanford University.*

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.

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

Kontkanen, P., Lahtinen, J., Myllymäki, P., Silander, T. and Tirri, H. (2000) Supervised model-based visualization of high-dimensional data. *Intell. Data Anal.*, **4**, 213-227.

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.

Kurgan, L. A. (2002) The ensemble of classifiers to improve accuracy of SPECT heart image analysis system. *Univ. Colorado Cen. Comput. Biol., Denver, U.S.A., Poster session.*

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.

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

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

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

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

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

Welch, L. R. (2003) Hidden Markov models and the Baum-Welch algorithm. *IEEE Inf. Soc. News (Shannon lecture)*, **53**, 10-13.