Higher-Order Markov Chain Models for Categorical Data Sequences*

Wai Ki Ching, Eric S. Fung, Michael K. Ng

Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong, People's Republic of China

Received 12 March 2002; revised 21 October 2003; accepted 28 January 2004

DOI 10.1002/nav.20017

Abstract: In this paper we study higher-order Markov chain models for analyzing categorical data sequences. We propose an efficient estimation method for the model parameters. Data sequences such as DNA and sales demand are used to illustrate the predicting power of our proposed models. In particular, we apply the developed higher-order Markov chain model to the server logs data. The objective here is to model the users' behavior in accessing information and to predict their behavior in the future. Our tests are based on a realistic web log and our model shows an improvement in prediction. © 2004 Wiley Periodicals, Inc. Naval Research Logistics 51: 557–574, 2004.

Keywords: higher-order Markov model; categorical data; linear programming

1. INTRODUCTION

Data sequences (or time series) occur frequently in many real world applications. The most important step in analyzing a data sequence (or time series) is the selection of an appropriate mathematical model for the data. Because it helps in predictions, hypothesis testing, and rule discovery. A data sequence X can be logically represented as a vector $(X_1, X_2, \ldots, X_T)$, where T is the length of the sequence, and $X_i \in DOM(A)$ $(1 \leq i \leq T)$, associated with a defined semantic and a data type. In this paper, we consider and assume other types used can be mapped to one of these two types. The domains of attributes associated with these two types are called numeric and categorical respectively. A numeric domain consists of real numbers. A domain $DOM(A)$ is defined as categorical if it is finite and unordered, e.g., for any $a, b \in DOM(A)$, either $a = b$ or $a \neq b$ (see, e.g., [8]). Numerical data sequences have been studied in detail (see, e.g., [5]). Mathematical tools such as Fourier transform and spectral analysis are employed

* The research of this project is supported by RGC Grant Nos. HKU 7130/02P, HKU 7126/02P, and 7046/03P and HKU CRCG Grant Nos. 10203501, 10203907, 10204436, and 10204437

Correspondence to: M. K. Ng (kkpong@hkusua.hku.hk)

frequently in the analysis of numerical data sequences. Different time sequences models are proposed and developed in the literatures [5].

For categorical data sequences, there are many situations that one would like to employ higher-order Markov chain models as a mathematical tool (see, e.g., [2, 11, 13–15]). A number of applications can be found in the literatures [9, 14, 16, 18]. For example, in sales demand prediction, products are classified into several states such as very high sales volume, high sales volume, standard, low sales volume, and very low sales volume (categorical type: ordinal data). A higher-order Markov chain model is then used to fit the observed data and apply in the wind turbine design. Alignment of sequences (categorical type: nominal data) is an important topic in DNA sequence analysis [18]. It involves searching of patterns in a DNA sequence of huge size. In these applications and many others, one would like to (i) characterize categorical data sequences for the purpose of comparison and classification process or (ii) model categorical data sequences and hence to make predictions in the control and planning process. It has been shown higher-order Markov chain models can be a promising approach for these purposes [9, 15, 16, 18].

For simplicity in discussion, in the following we assume that each data point X_t in a categorical data sequence takes values in

$$\mathcal{M} \equiv \{1, 2, \ldots, m\}$$

and m is finite, i.e., it has m possible categories or states. The conventional model for a nth order Markov chain has $(m - 1)m^n$ model parameters. The major problem in using such kind of model is that the number of parameters (the transition probabilities) increases exponentially with respect to the order of the model. This large number of parameters discourages people from using a higher-order Markov chain directly. In [15], Raftery proposed a higher-order Markov chain model which involves only one additional parameter for each extra lag. The model can be written as follows:

$$P(X_t = k_0 \mid X_{t-1} = k_1, \ldots, X_{t-n} = k_n) = \sum_{i=1}^{n} \lambda_i q_{k_0 k_i}, \tag{1}$$

where

$$\sum_{i=1}^{n} \lambda_i = 1$$

and $Q = [q_{ij}]$ is a transition matrix with column sums equal to one, such that

$$0 \leq \sum_{i=1}^{n} \lambda_i q_{k_0 k_i} \leq 1, \quad k_0, k_i \in \mathcal{M}. \tag{2}$$

The constraint in (2) is to guarantee that the right handside of (1) is a probability. The total number of independent parameters in his model is of $n + m^2$. Raftery proved that (1) is analogous to the standard AR(n) model in the sense that each additional lag, after the first is specified by a single parameter and the autocorrelations satisfy a system of linear equations similar to the Yule-Walker equations. Moreover, the parameters $q_{k_0 k_i}$, λ_i can be estimated numerically by maximizing the log-likelihood of (1) subjected to the constraints (2). However, this approach involves solving a highly nonlinear optimization problem (a coded program for solving the maximum log-likelihood problem can be found at http://lib.stat.cmu.edu/general/mtd). The proposed method neither guarantees convergence nor a global maximum. The main contribution of this paper is to generalize the Raftery model by allowing Q to vary with different lags. Numerical examples are given to demonstrate that our generalized model has a better prediction power than the Raftery model does. This means that our model is not overparameterized in general. We also develop an efficient method to estimate the model parameters.

The rest of the paper is organized as follows. In Section 2, we propose our higher-order Markov chain models and discuss some properties of the proposed model. In Section 3, we propose an estimation method for the model parameters required in our higher-order Markov chain model. In Section 4, numerical examples on DNA sequence and the sales demand data are given to demonstrate the predicting power of our model. In Section 5, we apply our higher-order Markov chain models to a real data set for web prediction. Finally, concluding remarks are given to conclude the paper in Section 6.

2. HIGHER-ORDER MARKOV CHAIN MODELS

In this section we extend the Raftery model [15] to a more general higher-order Markov model by allowing Q to vary with different lags. Here we assume that the weight λ_i is nonnegative such that

$$\sum_{i=1}^{n} \lambda_i = 1. \tag{3}$$

We first notice that (1) can be rewritten as

$$\mathbf{X}_{t+n+1} = \sum_{i=1}^{n} \lambda_i Q \mathbf{X}_{t+n+1-i}, \tag{4}$$

where $\mathbf{X}_{t+n+1-i}$ is the probability distribution of the states at time $(t + n + 1 - i)$. Using (3) and the fact that Q is a transition probability matrix, we note that each entry of $\mathbf{X}_{t+n+1}$ is in between 0 and 1, and the sum of all entries is also equal to 1. We remark that the Raftery model does not assume λ is nonnegative and therefore the additional constraints (2) should be added to guarantee that $\mathbf{X}_{t+n+1}$ is the probability distribution of the states.

The Raftery model in (4) can be generalized as follows:

$$\mathbf{X}_{t+n+1} = \sum_{i=1}^{n} \lambda_i Q_i \mathbf{X}_{t+n+1-i}. \tag{5}$$

The total number of independent parameters in the new model is $n + nm^2$. We note that if $Q_1 = Q_2 = \cdots = Q_n$ then (5) is just the Raftery model in (4).

In our model we assume that $\mathbf{X}_{t+n+1}$ depends on $\mathbf{X}_{t+i}$ $(i = 1, 2, \ldots, n)$ via the matrix Q_i and weight λ_i. One may relate Q_i to the ith step transition matrix of the process and we will use this idea to estimate Q_i. Here we assume that each Q_i is a nonnegative stochastic matrix with column sums equal to 1. Before we present our estimation method for the model parameters, we first discuss some properties of our proposed model in the following proposition.

PROPOSITION 1: If Q_n is irreducible and $\lambda_n > 0$ such that

$$0 \le \lambda_i \le 1 \quad \text{and} \quad \sum_{i=1}^{n} \lambda_i = 1,$$

then the model in (5) has a stationary distribution $\bar{\mathbf{X}}$ when $t \to \infty$ independent of the initial state vectors $\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_{n-1}$. The stationary distribution $\bar{\mathbf{X}}$ is also the unique solution of the following linear system of equations:

$$\left(I - \sum_{i=1}^{n} \lambda_i Q_i\right)\bar{\mathbf{X}} = \mathbf{0} \quad \text{and} \quad \mathbf{1}^T\bar{\mathbf{X}} = 1.$$

Here I is the m-by-m identity matrix (m is the number of possible states taken by each data point) and $\mathbf{1}$ is an m-vector of 1's.

PROOF: We first note that if $\lambda_n = 0$, then this is not an nth-order Markov chain. Therefore, $\lambda_n > 0$ is a reasonable assumption. Secondly, if Q_n is not irreducible, then we consider the case that $\lambda_n = 1$, and, in this case, clearly there is no unique stationary distribution for the system. Therefore, Q_n is irreducible is a necessary condition for the existence of a unique stationary distribution.

Now we let

$$\mathbf{Y}_{t+n+1} = (\mathbf{X}_{t+n+1}, \mathbf{X}_{t+n}, \ldots, \mathbf{X}_{t+2})^T$$

be an nm-by-1 vector. Then one may write

$$\mathbf{Y}_{n+1} = R\mathbf{Y}_n,$$

where

$$R = \begin{pmatrix} \lambda_1 Q_1 & \lambda_2 Q_2 & \cdots & \lambda_{n-1}Q_{n-1} & \lambda_n Q_n \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & 0 & & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & I & 0 \end{pmatrix} \tag{6}$$

is an nm-by-nm square matrix. We then define

$$\tilde{R} = \begin{pmatrix} \lambda_1 Q_1 & I & 0 & \cdots & \cdots & 0 \\ \vdots & 0 & I & 0 & & \vdots \\ \vdots & 0 & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ \lambda_{n-1} Q_{n-1} & \vdots & & \ddots & \ddots & I \\ \lambda_n Q_n & 0 & \cdots & 0 & 0 & 0 \end{pmatrix}. \tag{7}$$

We note that R and $\tilde{R}$ have the same characteristic polynomial in τ:

$$\det\left[(-1)^{n-1}\left((\lambda_1 Q_1 - \tau I)\tau^{n-1} + \sum_{i=2}^{n} \lambda_i Q_i \tau^{n-i}\right)\right].$$

Thus R and $\tilde{R}$ have the same set of eigenvalues.

It is clear that $\tilde{R}$ is an irreducible stochastic matrix with column sums equal to 1. Then from the Perron-Frobenius Theorem [4, p. 134], all the eigenvalues of $\tilde{R}$ (or equivalently R) lie in the interval (0, 1] and there is exactly one eigenvalue equal to one. This implies that

$$\lim_{t\to\infty} \overbrace{R \cdots R}^{t} = \lim_{t\to\infty} (R)^t = \mathbf{V}\mathbf{U}^T$$

is a positive rank-1 matrix as R is irreducible. Therefore, we have

$$\lim_{t\to\infty} \mathbf{Y}_{t+n+1} = \lim_{t\to\infty} (R)^t \mathbf{Y}_{n+1} = \mathbf{V}(\mathbf{U}^T \mathbf{Y}_{n+1}) = \alpha \mathbf{V}.$$

Here α is a positive number because $\mathbf{Y}_{n+1} \neq \mathbf{0}$ and is nonnegative. This implies that X_t also tends to a stationary distribution as t goes to infinity. Hence we have

$$\lim_{t\to\infty} \mathbf{X}_{t+n+1} = \lim_{t\to\infty} \sum_{i=1}^{n} \lambda_i Q_i \mathbf{X}_{t+n+1-i},$$

and therefore we have

$$\bar{\mathbf{X}} = \sum_{i=1}^{n} \lambda_i Q_i \bar{\mathbf{X}}.$$

The stationary distribution vector $\bar{\mathbf{X}}$ satisfies

$$\left(I - \sum_{i=1}^{n} \lambda_i Q_i\right)\bar{\mathbf{X}} = \mathbf{0} \quad \text{with} \quad \mathbf{1}^T \bar{\mathbf{X}} = 1. \tag{8}$$

The normalization constraint is necessary as the matrix

$$\left(I - \sum_{i=1}^{n} \lambda_i Q_i\right)$$

has a 1-dimensional null space. The result is then proved. □

We remark that if some λ_i are equal to zero, we can rewrite the vector $\mathbf{Y}_{t+n+1}$ in terms of $\mathbf{X}_i$, where λ_i are nonzero. Then the model in (5) still has a stationary distribution $\bar{\mathbf{X}}$ when t goes to infinity independent of the initial state vectors, and the stationary distribution $\bar{\mathbf{X}}$ can be obtained by solving the corresponding linear system of equations with the normalization constraint.

3. PARAMETERS ESTIMATION

In this section, we present two efficient methods to estimate the parameters Q_i and λ_i for $i = 1, 2, \ldots, n$. To estimate Q_i, we regard Q_i as the ith step transition matrix of the categorical data sequence $\{X_t\}$. Given the categorical data sequence $\{X_t\}$, one can count the transition frequency $f_{jk}^{(i)}$ in the sequence from state k to state j in the ith step. Hence one can construct the ith step transition matrix for the sequence $\{X_t\}$ as follows:

$$F^{(i)} = \begin{pmatrix} f_{11}^{(i)} & \cdots & \cdots & f_{m1}^{(i)} \\ f_{12}^{(i)} & \cdots & \cdots & f_{m2}^{(i)} \\ \vdots & \vdots & \vdots & \vdots \\ f_{1m}^{(i)} & \cdots & \cdots & f_{mm}^{(i)} \end{pmatrix}. \tag{9}$$

From $F^{(i)}$, we get the estimates for $Q_i = [q_{kj}^{(i)}]$ as follows:

$$\hat{Q}_i = \begin{pmatrix} \hat{q}_{11}^{(i)} & \cdots & \cdots & \hat{q}_{m1}^{(i)} \\ \hat{q}_{12}^{(i)} & \cdots & \cdots & \hat{q}_{m2}^{(i)} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{q}_{1m}^{(i)} & \cdots & \cdots & \hat{q}_{mm}^{(i)} \end{pmatrix}, \tag{10}$$

where

$$\hat{q}_{kj}^{(i)} = \begin{cases} \dfrac{f_{kj}^{(i)}}{\sum_{k=1}^{m} f_{kj}^{(i)}} & \text{if } \sum_{k=1}^{m} f_{kj}^{(i)} \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

We note that the computational complexity of the construction of $F^{(i)}$ is of $O(L^2)$ operations, where L is the length of the given data sequence. Hence the total computational complexity of the construction of $\{F^{(i)}\}_{i=1}^{n}$ is of $O(nL^2)$ operations. Here n is the number of lags.

The following proposition shows that these estimators are unbiased.

PROPOSITION 2: The estimators in (11) satisfies $E(f_{kj}^{(i)}) = q_{kj}^{(i)}E\ (\sum_{j=1}^{m} f_{kj}^{(i)})$.

PROOF: Let T be the length of the sequence, $[q_{kj}^{(i)}]$ be the ith step transition probability matrix and $\bar{X}_l$ be the steady state probability that the process is in state l. Then we have

$$E(f_{kj}^{(i)}) = T \cdot \bar{X}_k \cdot q_{kj}^{(i)}$$

and

$$E\left(\sum_{j=1}^{m} f_{kj}^{(i)}\right) = T \cdot \bar{X}_k \cdot \left(\sum_{j=1}^{m} q_{kj}^{(i)}\right) = T \cdot \bar{X}_k.$$

Therefore we have

$$E(f_{kj}^{(i)}) = q_{kj}^{(i)} \cdot E\left(\sum_{j=1}^{m} f_{kj}^{(i)}\right). \qquad \square$$

In some situations, if the sequence is too short so that $\hat{Q}_i$ (especially $\hat{Q}_n$) contains a lot of zeros (therefore $\hat{Q}_n$ may not be irreducible). We remark that this did not occur in our tested examples. Here we propose the second method. Let $\mathbf{W}^{(i)}$ be the distribution of the ith transition sequence; then another possible estimation for Q_i can be $\mathbf{W}^{(i)}\mathbf{1}^T$. We note that if $\mathbf{W}^{(i)}$ is a positive vector, then $\mathbf{W}^{(i)}\mathbf{1}^T$ will be a positive matrix and hence an irreducible matrix.

3.1. Linear Programming Formulation for Estimation of λ_i

Proposition 1 gives a sufficient condition for the sequence $\mathbf{X}_t$ to converge to a stationary distribution $\mathbf{X}$. Suppose $\mathbf{X}_t \to \bar{\mathbf{X}}$ as t goes to infinity then $\bar{\mathbf{X}}$ can be estimated from the sequence $\{X_t\}$ by computing the proportion of the occurrence of each state in the sequence and let us denote it by $\hat{\mathbf{X}}$. From (8) one would expect

$$\sum_{i=1}^{n} \lambda_i \hat{Q}_i \hat{\mathbf{X}} \approx \hat{\mathbf{X}}. \tag{12}$$

This suggests one possible way to estimate the parameters $\lambda = (\lambda_1, \ldots, \lambda_n)$ as follows. We consider the following optimization problem:

$$\min_{\lambda} \max_{k} \left| \left[\sum_{i=1}^{n} \lambda_i \hat{Q}_i \hat{\mathbf{X}} - \hat{\mathbf{X}} \right]_k \right|,$$

subject to

$$\sum_{i=1}^{n} \lambda_i = 1 \quad \text{and} \quad \lambda_i \geq 0, \ \forall\, i.$$

Here $[\cdot]_k$ denotes the kth entry of the vector. The constraints in the optimization problem guarantee the existence of the stationary distribution $\mathbf{X}$. Next we see that the above optimization problem formulate a linear programming problem:

$$\min_{\lambda} w$$

subject to

$$\begin{pmatrix} w \\ w \\ \vdots \\ w \end{pmatrix} \geq \hat{\mathbf{X}} - [\hat{Q}_1\hat{\mathbf{X}}|\hat{Q}_2\hat{\mathbf{X}}| \cdots |\hat{Q}_n\hat{\mathbf{X}}] \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix},$$

$$\begin{pmatrix} w \\ w \\ \vdots \\ w \end{pmatrix} \geq -\hat{\mathbf{X}} + [\hat{Q}_1\hat{\mathbf{X}}|\hat{Q}_2\hat{\mathbf{X}}| \cdots |\hat{Q}_n\hat{\mathbf{X}}] \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix},$$

$$w \geq 0, \quad \sum_{i=1}^{n} \lambda_i = 1, \quad \text{and} \quad \lambda_i \geq 0, \ \forall\, i.$$

We can solve the above linear programming problem efficiently and obtain the parameters λ_i. In the next subsection, we demonstrate the estimation method by a simple example.

Instead of solving a min-max problem, we remark that we can also formulate the following optimization problem:

$$\min_{\lambda} \sum_{k=1}^{n} \left| \left[\sum_{i=1}^{n} \lambda_i \hat{Q}_i \hat{\mathbf{X}} - \hat{\mathbf{X}} \right]_k \right|$$

subject to

$$\sum_{i=1}^{n} \lambda_i = 1 \quad \text{and} \quad \lambda_i \geq 0, \ \forall\, i.$$

The corresponding linear programming problem is given as follows:

$$\min_{\lambda} \sum_{k=1}^{m} w_k$$

subject to

$$\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix} \geq \hat{\mathbf{X}} - [\hat{Q}_1\hat{\mathbf{X}}|\hat{Q}_2\hat{\mathbf{X}}|\cdots|\hat{Q}_n\hat{\mathbf{X}}] \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix},$$

$$\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix} \geq -\hat{\mathbf{X}} + [\hat{Q}_1\hat{\mathbf{X}}|\hat{Q}_2\hat{\mathbf{X}}|\cdots|\hat{Q}_n\hat{\mathbf{X}}] \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix},$$

$$w_i \geq 0, \qquad \forall\, i, \qquad \sum_{i=1}^{n} \lambda_i = 1, \qquad \text{and} \qquad \lambda_i \geq 0, \qquad \forall\, i.$$

In the above linear programming formulation, the number of variables is equal to n and the number of constraints is equal to $2m + 1$. The order of the linear programming is linear in the number of lags and in the number of states. Therefore, the expected computational complexity of solving the above linear programming problem is of $O(m^2 n)$ [7, p. 96].

3.2. An Example

We consider a sequence $\{X_t\}$ of three states ($m = 3$) given by

$$\{1, 1, 2, 2, 1, 3, 2, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 1, 2\}. \tag{13}$$

The sequence $\{X_t\}$ can be written in vector form

$$\mathbf{X}_1 = (1, 0, 0)^T, \qquad \mathbf{X}_2 = (1, 0, 0)^T, \qquad \mathbf{X}_3 = (0, 1, 0)^T, \qquad \ldots, \qquad \mathbf{X}_{20} = (0, 1, 0)^T.$$

We consider $n = 2$, then from (13) we have the transition frequency matrices

$$F^{(1)} = \begin{pmatrix} 1 & 3 & 3 \\ 6 & 1 & 1 \\ 1 & 3 & 0 \end{pmatrix} \quad \text{and} \quad F^{(2)} = \begin{pmatrix} 1 & 4 & 1 \\ 3 & 2 & 3 \\ 3 & 1 & 0 \end{pmatrix}. \tag{14}$$

Therefore from (14) we have the i-step transition matrices ($i = 1, 2$) as follows:

$$\hat{Q}_1 = \begin{pmatrix} 1/8 & 3/7 & 3/4 \\ 3/4 & 1/7 & 1/4 \\ 1/8 & 3/7 & 0 \end{pmatrix} \quad \text{and} \quad \hat{Q}_2 = \begin{pmatrix} 1/7 & 4/7 & 1/4 \\ 3/7 & 2/7 & 3/4 \\ 3/7 & 1/7 & 0 \end{pmatrix} \tag{15}$$

and

$$\hat{\mathbf{X}} = (\tfrac{2}{5}, \tfrac{2}{5}, \tfrac{1}{5})^T.$$

Hence we have

$$\hat{Q}_1\hat{\mathbf{X}} = (\tfrac{13}{35}, \tfrac{57}{140}, \tfrac{31}{140})^T \quad \text{and} \quad \hat{Q}_2\hat{\mathbf{X}} = (\tfrac{47}{140}, \tfrac{61}{140}, \tfrac{8}{35})^T.$$

To estimate λ_i we consider the optimization problem:

$$\min_{\lambda_1,\lambda_2} w$$

subject to

$$\begin{cases} w \geq \frac{2}{5} - \frac{13}{35}\lambda_1 - \frac{47}{140}\lambda_2, \\ w \geq -\frac{2}{5} + \frac{13}{35}\lambda_1 + \frac{47}{140}\lambda_2, \\ w \geq \frac{2}{5} - \frac{57}{140}\lambda_1 - \frac{61}{140}\lambda_2, \\ w \geq -\frac{2}{5} + \frac{57}{140}\lambda_1 + \frac{61}{140}\lambda_2, \\ w \geq \frac{1}{5} - \frac{31}{140}\lambda_1 - \frac{8}{35}\lambda_2, \\ w \geq -\frac{1}{5} + \frac{31}{140}\lambda_1 + \frac{8}{35}\lambda_2, \\ w \geq 0, \quad \lambda_1 + \lambda_2 = 1, \quad \lambda_1, \lambda_2 \geq 0. \end{cases}$$

The optimal solution is

$$(\lambda_1^*, \lambda_2^*, w^*) = (1, 0, 0.0286),$$

and we have the model

$$\mathbf{X}_{t+1} = \hat{Q}_1\mathbf{X}_t. \tag{16}$$

We remark that if we do not specify the nonnegativity of λ_1 and λ_2, the optimal solution becomes

$$(\lambda_1^{**}, \lambda_2^{**}, w^{**}) = (1.80, -0.80, 0.0157),$$

the corresponding model is

$$\mathbf{X}_{t+1} = 1.80\hat{Q}_1\mathbf{X}_t - 0.80\hat{Q}_2\mathbf{X}_{t-1}. \tag{17}$$

Although w^{**} is less than w^*, the model (17) is not suitable. It is easy to check that

$$1.80\hat{Q}_1\begin{pmatrix}1\\0\\0\end{pmatrix} - 0.80\hat{Q}_2\begin{pmatrix}0\\1\\0\end{pmatrix} = \begin{pmatrix}-0.2321\\1.1214\\0.1107\end{pmatrix};$$

therefore, λ_1^{**} and λ_2^{**} are not valid parameters.

We note that if we consider the optimization problem:

$$\min_{\lambda_1,\lambda_2} w_1 + w_2 + w_3,$$

subject to

$$\begin{cases} w_1 \geq \frac{2}{5} - \frac{13}{35}\lambda_1 - \frac{47}{140}\lambda_2, \\ w_1 \geq -\frac{2}{5} + \frac{13}{35}\lambda_1 + \frac{47}{140}\lambda_2, \\ w_2 \geq \frac{2}{5} - \frac{57}{140}\lambda_1 - \frac{61}{140}\lambda_2, \\ w_2 \geq -\frac{2}{5} + \frac{57}{140}\lambda_1 + \frac{61}{140}\lambda_2, \\ w_3 \geq \frac{1}{5} - \frac{31}{140}\lambda_1 - \frac{9}{35}\lambda_2, \\ w_3 \geq -\frac{1}{5} + \frac{31}{140}\lambda_1 + \frac{9}{35}\lambda_2, \\ w_1, w_2, w_3 \geq 0, \qquad \lambda_1 + \lambda_2 = 1, \qquad \lambda_1, \lambda_2 \geq 0. \end{cases}$$

The optimal solution is the same as the previous min-max formulation and is equal to

$$(\lambda_1^*, \lambda_2^*, w_1^*, w_2^*, w_3^*) = (1, 0, 0.0286, 0.0071, 0.0214).$$

4. SOME PRACTICAL EXAMPLES

In this section we apply our model to some data sequences. The data sequences are the DNA sequence and the sales demand data sequence. Given the state vectors $\mathbf{X}_i$, $i = t - n, t - n + 1, \ldots, t - 1$, the state probability distribution at time t can be estimated as follows:

$$\hat{\mathbf{X}}_t = \sum_{i=1}^{n} \lambda_i \hat{Q}_i \mathbf{X}_{t-i}.$$

In many applications, one would like to make use of the higher-order Markov models for the purpose of prediction. According to the this state probability distribution, the prediction of the next state $\hat{X}_t$ at time t can be taken as the state with the maximum probability, i.e.,

$$\hat{X}_t = j, \qquad \text{if } [\hat{\mathbf{X}}_t]_i \leq [\hat{\mathbf{X}}_t]_j, \qquad \forall\, 1 \leq i \leq m.$$

To evaluate the performance and effectiveness of our higher-order Markov chain model, a prediction result is measured by the prediction accuracy r defined as

Table 1. Prediction accuracy in the DNA sequence.

	2-State model	3-State model	4-State model
New model	0.57	0.49	0.33
Raftery's model	0.57	0.47	0.31
Random chosen	0.50	0.33	0.25

$$r = \frac{\sum_{t=n+1}^{T} \delta_t}{T},$$

where T is the length of the data sequence and

$$\delta_t = \begin{cases} 1, & \text{if } \hat{X}_t = X_t \\ 0, & \text{otherwise.} \end{cases}$$

Using the example in the previous section, there are two possible prediction rules:

$$\begin{cases} \hat{X}_{t+1} = 2, & \text{if } X_t = 1, \\ \hat{X}_{t+1} = 1, & \text{if } X_t = 2, \\ \hat{X}_{t+1} = 1, & \text{if } X_t = 3 \end{cases}$$

or

$$\begin{cases} \hat{X}_{t+1} = 2, & \text{if } X_t = 1, \\ \hat{X}_{t+1} = 3, & \text{if } X_t = 2, \\ \hat{X}_{t+1} = 1, & \text{if } X_t = 3. \end{cases}$$

The prediction accuracy r for the sequence in (13) is equal to 12/19 for both prediction rules. We note that the prediction accuracies of other rules for the sequence in (13) are less than 12/19.

Next the test results on different data sequences are discussed. In the following tests, we solve min-max optimization problems to determine the parameters λ_i of higher-order Markov models. However, we remark that the results of using the 1-norm optimization problem as discussed in the previous section are about the same as that of using the min-max formulation. All the computations here are done by MATLAB with a PC.

4.1. The DNA Sequence

In order to determine whether certain short DNA sequence (a categorical data sequence of four possible categories) occurred more often than would be expected by chance, Avery [3] examined the Markovian structure of introns from several other genes in mice. Here we apply our model to the introns from the mouse αA-crystallin gene see for instance [16]. We compare our second-order model with the Raftery second-order model. The model parameters of the Raftery model are given in [16]. The results are reported in Table 1 below. The comparison is made with different grouping of states as suggested in [16]. In grouping states 1 and 3, and states 2 and 4 we have a 2-state model. Our model gives

$$\hat{Q}_1 = \begin{pmatrix} 0.5568 & 0.4182 \\ 0.4432 & 0.5818 \end{pmatrix}, \qquad \hat{Q}_2 = \begin{pmatrix} 0.4550 & 0.5149 \\ 0.5450 & 0.4851 \end{pmatrix},$$

$$\hat{\mathbf{X}} = (0.4858, 0.5142)^T, \qquad \lambda_1 = 0.7529, \qquad \text{and} \quad \lambda_2 = 0.2471.$$

In grouping states 1 and 3 we have a 3-state model. Our model gives

$$\hat{Q}_1 = \begin{pmatrix} 0.5568 & 0.3573 & 0.4949 \\ 0.2571 & 0.3440 & 0.2795 \\ 0.1861 & 0.2987 & 0.2256 \end{pmatrix}, \quad \hat{Q}_2 = \begin{pmatrix} 0.4550 & 0.5467 & 0.4747 \\ 0.3286 & 0.2293 & 0.2727 \\ 0.2164 & 0.2240 & 0.2525 \end{pmatrix},$$

$$\hat{\mathbf{X}} = (0.4858, 0.2869, 0.2272)^T, \quad \lambda_1 = 1.0, \quad \text{and} \quad \lambda_2 = 0.0.$$

If there is no grouping, we have a 4-state model. Our model gives

$$\hat{Q}_1 = \begin{pmatrix} 0.2268 & 0.2987 & 0.2274 & 0.1919 \\ 0.2492 & 0.3440 & 0.2648 & 0.2795 \\ 0.3450 & 0.0587 & 0.3146 & 0.3030 \\ 0.1789 & 0.2987 & 0.1931 & 0.2256 \end{pmatrix},$$

$$\hat{Q}_2 = \begin{pmatrix} 0.1891 & 0.2907 & 0.2368 & 0.2323 \\ 0.3814 & 0.2293 & 0.2773 & 0.2727 \\ 0.2532 & 0.2560 & 0.2305 & 0.2424 \\ 0.1763 & 0.2240 & 0.2555 & 0.2525 \end{pmatrix},$$

$$\hat{\mathbf{X}} = (0.2395, 0.2869, 0.2464, 0.2272)^T, \qquad \lambda_1 = 0.253, \quad \text{and} \quad \lambda_2 = 0.747.$$

When using the expected errors (assuming that the next state is randomly chosen with equal probability for all states) as a reference, the percentage gain in effectiveness of using higher-order Markov chain models is in the 3-state model. In this case, our model also gives a better estimation compared with the Raftery model. Raftery [15] refers to using BIC to weight efficiency gained in terms of extra parameters used. This is important in his approach since his method requires to solve a highly nonlinear optimization problem. The complexity of solving the optimization problem increases when there are many parameters to be estimated. We remark that our estimation method is quite efficient. The main cost is to solve a linear programming problem and the expected computational complexity of solving the above linear programming problem is of $O(m^2n)$, where m is the number of states and n is the order of the model (see Section 3).

4.2. The Sales Demand Data

A large soft-drink company in Hong Kong presently faces an in-house problem of production planning and inventory control. A pressing issue that stands out is the storage space of its central warehouse, which often finds itself in the state of overflow or near capacity. The company is thus in urgent needs to study the interplay between the storage space requirement and the overall growing sales demand. There are product states due to the level of sales volume. The states include:

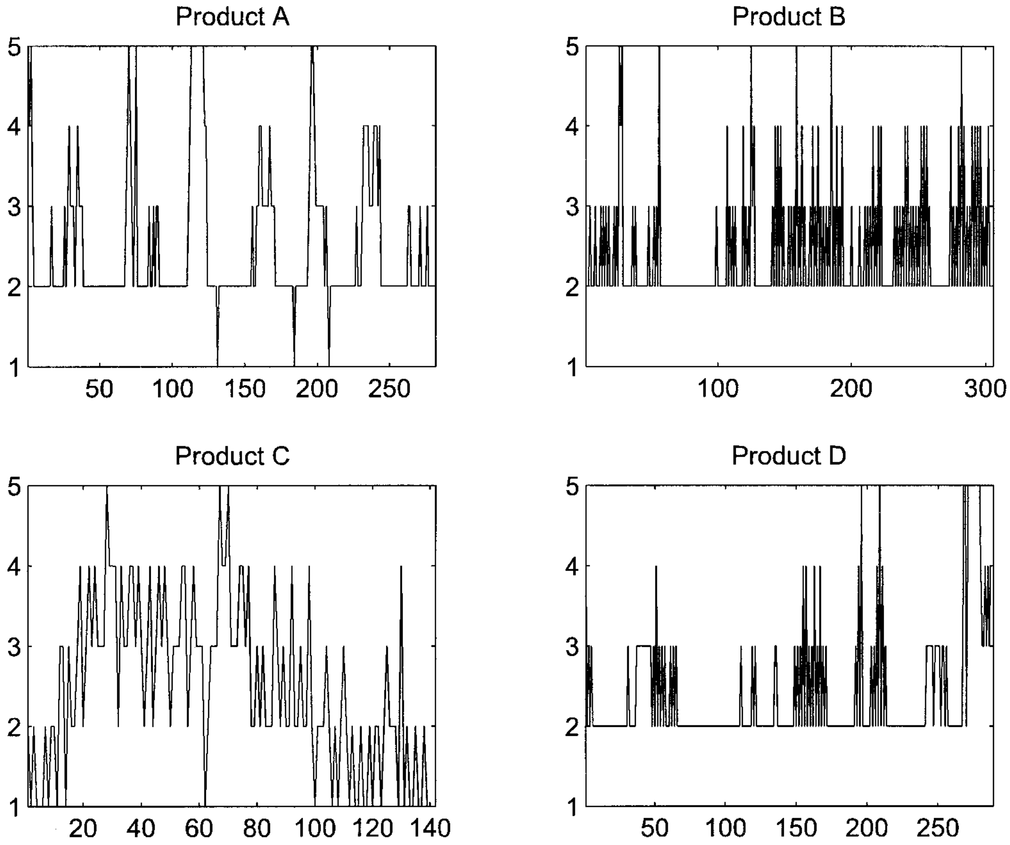

Figure 1. The states of four products A, B, C, and D.

State 1: very slow-moving (very low sales volume)
State 2: slow-moving
State 3: standard
State 4: fast-moving
State 5: very fast-moving (very high sales volume).

Such labelings are useful from both marketing and production planning points of view. For instance, in the production planning, the company develops a dynamic programming (DP) model to recommend better production planning so as to minimize its inventory build-up, and to maximize the demand satisfaction as well. Since the number of alternatives at each stage (each day in the planning horizon) are very large (the number of products raised to the power of the number of production lines), the computational complexity of the DP model is enormous. A priority scheme based on the state (the level of sales volume) of the product is introduced to tackle this combinatorial problem, and therefore an effective and efficient production plan can be obtained. It is obvious that the accurate prediction of state (the level of sales volume) of the product is important in the production planning model.

In Figure 1, we show that the states of four products of the soft-drink company for some sales periods. Here we employ higher-order Markov models to predict categories of these four

Table 2. Prediction accuracy in the sales demand data.

	Product A	Product B	Product C	Product D
First-order Markov model	0.76	0.70	0.39	0.74
Second-order Markov model	0.79	0.78	0.51	0.83
New model ($n = 2$)	0.78	0.76	0.43	0.78
Random chosen	0.20	0.20	0.20	0.20

products separately. For our new model, we consider the second-order ($n = 2$) model and use the data to estimate $\hat{Q}_i$ and λ_i ($i = 1, 2$). The results are reported in Table 2. For comparison, we also study the first-order and the second-order full Markov chain model. Results show the effectiveness of our new model. We also see from Figure 1 that the change of the states of the products A, B, and D is more regular than that of the product C. We find in Table 2 that the prediction results for the products A, B, and D are better than that of C.

5. APPLICATIONS TO WEB PREDICTION

The Internet provides a rich environment for users to retrieve information. However, it is easy for a user to get lost in the sea of information. One way to assist the user with their informational need is to predict a user's future request and use the prediction for recommendation. Recommendation systems reply on a prediction model to make inferences on users' interests based upon which to make recommendations. Examples are the WebWatcher [10] system and Letzia [12] system. Accurate prediction can potentially shorten the users' access times and reduce network traffic when the recommendation is handled correctly. In this section, we use a higher-order Markov chain model to exploit the information from Web server logs for predicting users' actions on the web.

Our higher-order Markov chain model is built on a Web server log file. We consider the Web server log file to be preprocessed into a collection of user sessions. Each session is indexed by a unique user id and starting time [17]. Each session is a sequence of requests where each request corresponds to a visit to a web page. For simplicity, we represent each request as a state. Then each session is just a categorical data sequence. For simplicity, we denote each Web page (state) by an integer.

5.1. Web Log Files and Preprocessing

Experiments were conducted on a real Web log file taken from the Internet. We first implemented a data preprocessing program to extract sessions from the log file. We downloaded two web log files from the Internet. The data set was a Web log file from the EPA WWW server located at Research Triangle Park, NC. This log contained 47748 transactions generated in 24 hours from 23:53:25 EDT, August 29, to 23:53:07, August 30, 1995. In preprocessing, we removed all the invalid requests and the requests for images. We used Host id to identify visitors and a 30-min time threshold to identify sessions. 428 sessions of lengths between 16 and 20 were identified from the EPA log file. The total number of web pages (states) involved is 3753.

5.2. Prediction Models

By exploring the session data from the Web log file, we have observed that a large number of similar sessions rarely exist. This is because in a complex Web site with variety of pages, and

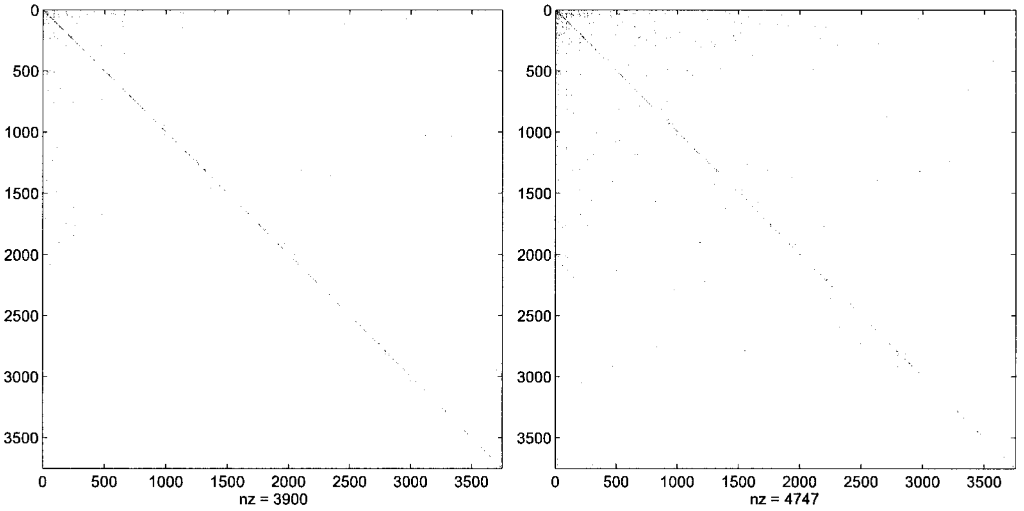

Figure 2. The first (left) and the second (right) step transition matrices of all sessions.

many paths and links, one should not expect that, in a given time period, a large number of visitors follow only a few paths. If this was true, it would mean that the structure and contents of the Web site had a serious problem because only a few pages and paths were interested by the visitors. In fact, most Web site designers expect that the majority of their pages, if not every one, are visited and paths followed (equally) frequently.

In Figure 2, we depict the first and the second step transition matrices of all sessions. It is clear that these matrices are very sparse. There are 3900 and 4747 entries in the first and the second step transition matrices respectively. Nonzero entries only contain about 0.033% in the total elements of the first and the second step transition matrices.

Based on these observations, if we directly use these transition matrices to build prediction models, they may not be effective. Since the number of pages (states) are very large, the prediction probability for each page may be very low. Moreover, the computational work for solving the linear programming problem in the estimation of λ_i are also high since the number of constraints in the linear programming problem depends on the number of pages (states). Here we propose to use clustering algorithms [9] to cluster the sessions. The idea is to form a transition probability matrix for each session, to construct the distance between two sessions based on the Frobenius norm of the difference of their transition probability matrices, and then to use k-means algorithm to cluster the sessions. As a result of the cluster analysis, the Web page cluster can be used to construct a higher-order Markov chain model. Then we prefetch those web documents that are close to a user-requested document in a Markov chain model.

We find that there is a clear similarity among these sessions in each cluster for the EPA log file. As an example, we show in Figure 3 that the first, the second, and the third step transition probability matrices of a cluster in EPA log file. There are 70 pages involved in this cluster. Nonzero entries contain about 1.92%, 2.06%, and 2.20%, respectively, in the total elements of the first, the second, and the third step transition matrices. Usually, the prediction of the next Web page is based on the current page and the previous few pages [1]. Therefore, we use a third-order model ($n = 3$) and consider the first, the second, and the third transition matrices in the construction of the Markov model. After we find the transition matrices, we determine λ_i and build our new higher-order Markov chain model for each cluster. For the above mentioned

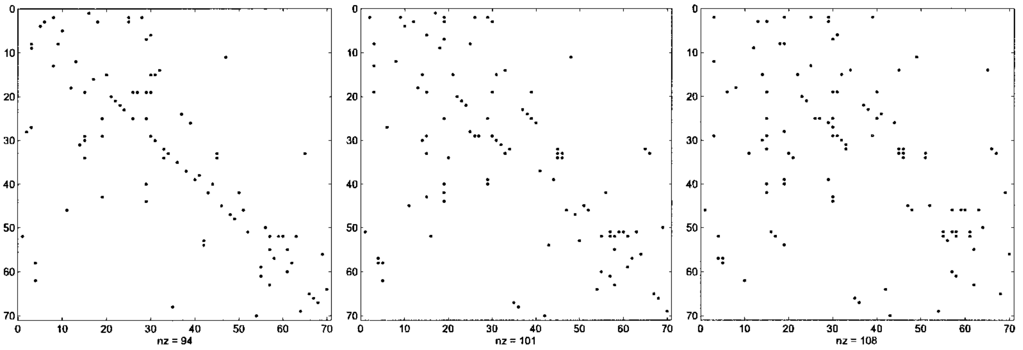

Figure 3. The first (left), the second (middle), and the third (right) step transition matrices of a cluster.

cluster, its corresponding λ_1, λ_2, and λ_3 are 0.4984, 0.4531, and 0.0485, respectively. The parameters show that the prediction of the next Web page strongly depends on the current and the previous pages.

Below we present the prediction results for the EPA log file. We perform clustering based on their transition matrices and parameters. Sixteen clusters are found experimentally based on their average within-cluster distance, and therefore 16 third-order Markov chain model for these clusters are determined for the prediction of user-request documents. For comparison, we also compute the first-order Markov chain model for each cluster. Totally, there are 6255 web documents for the prediction test. We find the prediction accuracy of our method is about 0.77, but the prediction accuracy of using the first-order full Markov chain model is only 0.75. Results show an improvement in the prediction. We have applied these prediction results to the problem of integrated web caching and prefetching [19]. The slight increase of the prediction accuracy can power a prefetching engine. Experimental results in [19] show that the resultant system outperforms Web systems that are based on caching alone.

6. CONCLUDING REMARKS

In this paper, we proposed and developed a higher-order Markov chain model for categorical data sequences. The number of model parameters increases linearly with respect to the number of lags. Efficient estimation methods for the model parameters are also proposed by making use of the observed transition frequencies and the steady state distribution. The expected computational complexity of our estimation methods is of $(nL^2 + nm^2)$, where n is the number of lags, m is the number of states and L is the length of sequence. Numerical examples in the DNA sequences and sales demand are given to demonstrate the predicting power of our model. We also apply the developed higher-order Markov chain model to the server logs data. Our tests are based on a realistic Web log and our model has shown an improvement in the prediction of the users' behavior in accessing information.

We conclude the paper by giving the following possible extensions of our model in future research:

- For the problem of modeling sales demands, we have assumed that the products are independent. We then constructed a higher-order Markov chain model for each product individually. However, the demands of the products can be correlated. Therefore, it is natural to further develop Markov models for modeling multiple categorical data sequences together, and to get better prediction rules.

- It is possible to extend our model to the case of Hidden Markov Models (HMMs) [14]. It is well known that the HMMs are first order Markov models. It is interesting to develop a higher-order HMM based on our proposed approach.

REFERENCES

[1] D. Albrecht, I. Zukerman, and A. Nicholson, Pre-sending documents on the WWW: A comparative study, Proc Sixteenth Int Joint Conf Artif Intell IJCAI99, 1999, pp. 1274–1279.
[2] S. Adke and D. Deshmukh, Limit distribution of a high order Markov chain, J Roy Statist Soc Ser B 50 (1988), 105–108.
[3] P. Avery, The analysis of intron data and their use in the detection of short signals, J Mol Evol 26 (1987), 335–340.
[4] O. Axelsson, Iterative solution methods, Cambridge University Press, Cambridge, 1996.
[5] P. Brockwell and R. Davis, Time series: Theory and methods, Springer-Verlag, New York, 1991.
[6] W. Craig, The song of the wood pewee, University of the State of New York, Albany, 1943.
[7] S. Fang and S. Puthenpura, Linear optimization and extensions, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[8] K. Gowda and E. Diday, Symbolic clustering using a new dissimilarity measure, Pattern Recognition 24(6) (1991), 567–578.
[9] J. Huang, M. Ng, W. Ching, D. Cheung, and J. Ng, A cube model for Web access sessions and cluster analysis, WEBKDD 2001, Workshop on Mining Web Log Data Across All Customer Touch Points, Seventh ACM SIGKDD Int Conf Knowledge Discovery Data Mining, August 2001, pp. 47–58.
[10] T. Joachims, D. Freitag, and T. Mitchell, Web Watch: A tour guide for the World Wide Web, Proc Fifteenth Int Joint Conf Artif Intell IJCAI 97, 1997, pp. 770–775.
[11] W. Li and M. Kwok, Some results on the estimation of a higher order Markov chain, Department of Statistics, The University of Hong Kong, Hong Kong, 1989.
[12] H. Lieberman, Letizia: An agent that assists Web browsing, Proc Fourteenth Int Joint Conf Artif Intell IJCAI 95, 1995, pp. 924–929.
[13] J. Logan, A structural model of the higher-order Markov process incorporating reversion effects, J Math Sociol 8 (1981), 75–89.
[14] I. MacDonald and W. Zucchini, Hidden Markov and other models for discrete-valued time series, Chapman & Hall, London, 1997.
[15] A. Raftery, A model for high-order Markov chains, J Roy Statist Soc Ser B 47 (1985), 528–539.
[16] A. Raftery and S. Tavare, Estimation and modelling repeated patterns in high order Markov chains with the mixture transition distribution model, Appl Statist 43 (1994), 179–199.
[17] C. Shahabi, A. Faisal, F. Kashani, and J. Faruque, INSITE: A tool for real time knowledge discovery from users Web navigation, Proc VLDB2000, Cairo, Egypt, 2000, pp. 635–638.
[18] M. Waterman, Introduction to computational biology, Chapman & Hall, Cambridge, 1995.
[19] Q. Yang, Z. Huang, and M. Ng, A data cube model for prediction-based Web prefetching, J Intell Inform Syst 20 (2003), 11–30.