We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

19

A Hierarchical Bayesian Hidden Markov Model for Multi-Dimensional Discrete Data

Shigeru Motoi [1], Yohei Nakada [1], Toshie Misu [2], Tomohiro Yazaki [1]
Takashi Matsumoto [1] and Nobuyuki Yagi [2]
[1] *Faculty of Science and Engineering, Waseda University, Tokyo, Japan*
[2] *Science and Technical Research Laboratories, NHK (Japan Broadcasting Corporation), Tokyo, Japan*

1. Introduction

1.1 Motivation

A fundamental problem encountered in many fields is to model data o_t given a discrete time-series data sequence $y := (o_1, \cdots, o_T)$. This problem is found in diverse fields, such as control systems, robotics, event detection (Motoi et al., 2007), handwriting recognition (Yasuda et al., 2000 ; Funada et al., 2005), and protein structure prediction (Krogh et al., 2001 ; Tusnady & Simon, 1998 ; Kaburagi et al., 2007). The data o_t can often be a multi-dimensional variable exhibiting stochastic activity. A powerful tool for solving such problems is multi-dimensional discrete Hidden Markov Models (HMMs), and the effectiveness of this approach has been demonstrated in numerous studies (Motoi et al., 2007 ; Yasuda et al., 2000 ; Funada et al., 2005 ; Kaburagi et al., 2007). The hidden states of the HMMs are treated as hidden factors for emission of the observed data o_t. However, if redundant components having low dependencies on the hidden states are contained in the data o_t, these components often have a negative impact on the HMM performance. Overcoming this problem requires a method of quantifying the redundancy (state independence) of these components and/or reducing their influence.

In this chapter, we describe an extension of the HMM for these kinds of data sequences within the framework of a hierarchical Bayesian scheme. In this extended model, we introduce *commonality hyperparameters* to describe the degree of commonality of the emission probabilities among different hidden states (that is, hidden factors of the data o_t). Additionally, there is a one-to-one relationship between each hyperparameter and a component of the data o_t. This allows us to identify low-dependency components and to minimize their negative impact.

Like other Bayesian HMMs, the extended model requires complicated integrations in the learning and prediction processes, usually involving a posterior distribution. Analytic solutions of these integrations are often intractable or non-trivial due to their inherent

complexity. In this chapter, therefore, we also describe an implementation based on a Markov Chain Monte Carlo (MCMC) method (Scott, 2002).

1.2 Related work

In one detailed study, several feature selection methods were considered, such as discriminant feature analysis, principal component analysis, and the sequential search method (Nouza, 1996). In addition, that study also described a fast feature selection algorithm. Our approach described in this chapter may be regarded as a Bayesian feature selection scheme based on the dependencies of the hidden states.

There have been a number of studies examining Bayesian HMMs and their implementations, such as (Funada et al., 2005 ; Motoi et al., 2007 ; Huo et al., 1995 ; MacKay, 1997 ; Scott, 2002). Reference (Huo et al., 1995) describes a Maximum A Posteriori (MAP) estimation for Bayesian HMMs, and reference (MacKay, 1997) describes a Variational Bayesian method (so-called ensemble learning). In addition, references (Funada et al., 2005 ; Motoi et al., 2007 ; Scott, 2002) discuss Bayesian HMMs using MCMC. The model that we describe here is an extension of such Bayesian HMMs for discrete multi-dimensional data containing redundant components.

There is a well-known successful method to determine redundant components of multi-dimensional (input) data in the field of Bayesian Neural Networks (BNNs), called Automatic Relevance Determination (ARD) (MacKay, 1992 ; Neal, 1996 ; Qi et al., 2004 ; Tipping, 2000 ; Matsumoto et al., 2001 ; Nakada et al., 2005). ARD was first described in (MacKay, 1992); that method used a Laplace approximation. Reference (Neal, 1996) described another ARD using MCMC, and reference (Qi et al., 2004) discusses a variant based on Expectation Propagation. Several studies have also described extensions of the BNN using the ARD method, including, for example, the Relevance Vector Machine (Tipping, 2000) and BNNs for nonlinear time-series data (Matsumoto et al., 2001 ; Nakada et al., 2005). The structure of the extended HMM is completely different from that of such BNNs;nevertheless, the fundamental hierarchical Bayesian concepts show a number of underlying similarities.

2. Model specification

In this section, we describe the extended Bayesian HMM. The setting of hyperparameters is the principal difference between our extended model and the conventional Bayesian HMMs (see Sec. 2.5).

2.1 HMM Topology

The HMM structure depends on the particular topology employed and the number of states N. Topologies commonly employed include “ergodic” and “left-to-right”. Here we describe only the ergodic topology, since we employed that topology in our experiments, described later.

2.2 Data and hidden variables

In the HMM framework, we must consider the time-series data sequence (observation data sequence) $y := (o_1, \cdots, o_T)$ and the hidden variable sequence $z := (q_1, \cdots, q_T)$. The terms o_t and

q_t represent the time-series data and the hidden variable at time t, and T is the sequence length. The hidden variable q_t is a one-dimensional variable that takes finite values among the available N states (that is, $q_t \in \{1,\cdots,N\}$), whereas the data o_t is a multi-dimensional discrete variable defined by $o_t := (o_{1,t},\cdots,o_{D,t})$. Here, D represents the dimension of the data o_t, the variable $o_{k,t}$ the k-th component of o_t, and M_k the number of symbols for $o_{k,t}$ (in other words, $o_{k,t} \in \{1,\cdots,M_k\}$).

2.3 Observation model

Consider the complete parameter set θ of an HMM. The probability of the data y_t is

$$P(y \mid \theta) := \sum_z P(y \mid z,b)P(z \mid a,c), \quad \theta := (a,b,c), \tag{1}$$

Here,

$$P(y \mid z,b) := \prod_{t=1}^{T} P(o_t \mid q_t,b), \tag{2}$$

$$P(z \mid a,c) := P(q_1 \mid c)\prod_{t=2}^{T} P(q_t \mid q_{t-1},a). \tag{3}$$

The emission probability of the data o_t in (2) is

$$P(o_t \mid q_t,b) := \prod_{k=1}^{D} P(o_{k,t} \mid q_t,b_k), \tag{4}$$

where $b := (b_1,\cdots,b_D)$. The probability $P(o_{k,t} \mid q_t,b_k)$ in Eqn. (4) represents the emission probability of the k-th component $o_{k,t}$. It is defined as

$$P(o_{k,t} = j \mid q_t = i,b_k) := b_{k,ij}, \tag{5}$$

where $b_k := (b_{k,1},\cdots,b_{k,N})$, $b_{k,i} := (b_{k,i1},\cdots,b_{k,iM_k})$, $\sum_{j=1}^{M_k} b_{k,ij} = 1$, and $0 \le b_{k,ij} \le 1$.

The hidden variable transition probability and the initial hidden variable probability in Eqn. (3) are

$$P(q_t = j \mid q_{t-1} = i,a) := a_{ij}, \quad i,j > 1, \tag{6}$$

$$P(q_1 = i \mid c) := c_i, \tag{7}$$

Here, $a := (a_1, \cdots, a_N)$, $a_i := (a_{i1}, \cdots, a_{iN})$, $\sum_{j=1}^{N} a_{ij} = 1$, $0 \le a_{ij} \le 1$, $c := (c_1, \cdots, c_N)$, $\sum_{i=1}^{N} c_i = 1$, and $0 \le c_i \le 1$.

2.4 Prior distribution for parameters

Within a Bayesian framework, both the observation model (the likelihood function) and the prior distribution of the parameter set are defined. For the sake of simplicity, many Bayesian HMMs assume parameter independency in the prior distribution. That is to say:

$$P(\theta \mid \phi) = P(a \mid \alpha) P(b \mid \beta) P(c \mid \gamma), \tag{8}$$

$$P(a \mid \alpha) := \prod_{i=1}^{N} P(a_i \mid \alpha_i), \tag{9}$$

$$P(b \mid \beta) = \prod_{k=1}^{D} \prod_{i=1}^{N} P(b_{k,i} \mid \beta_{k,i}), \tag{10}$$

where

$$\phi := (\alpha, \beta, \gamma), \quad \alpha := (\alpha_1, \cdots, \alpha_N)$$
$$\beta := (\beta_1, \cdots, \beta_D), \quad \beta_k := (\beta_{k,1}, \cdots, \beta_{k,N}).$$

The prior distributions of a_i, $b_{k,i}$ and c in Eqns. (8)-(10) are also defined using the "natural conjugate" Dirichlet prior distribution:

$$P(a_i \mid \alpha_i) := \mathcal{D}(a_i; \alpha_i), \tag{11}$$

$$P(b_{k,i} \mid \beta_{k,i}) := \mathcal{D}(b_{k,i}; \beta_{k,i}), \tag{12}$$

$$P(c \mid \gamma) := \mathcal{D}(c; \gamma), \tag{13}$$

where $\mathcal{D}(\cdot; \chi)$ is the Dirichlet distribution with the parameter vector χ, and $\alpha_i := (\alpha_{i1}, \cdots, \alpha_{iN})$, $\alpha_{ij} > 0$, $\beta_{k,i} := (\beta_{k,i1}, \cdots, \beta_{k,iN})$, $\beta_{k,ij} > 0$, $\gamma := (\gamma_1, \cdots, \gamma_N)$, $\gamma_i > 0$.

2.5 Settings for hyperparameter set

As in a number of conventional Bayesian HMMs, for example, (Funada et al., 2005 ; Huo et

al., 1995), all components of the hyperparameter vectors are fixed at 1.0, except for $\beta_{k,i}$.[1] With our approach on the other hand, we consider a reparameterization of the hyperparameter vectors $\{\beta_{k,i}\}_{i=1}^{N}$, and the prior distribution of the reparameterized hyperparameters in order to identify components having low dependency on the states (redundant components).

A. Reparameterization of $\beta_{k,i}$

We define the hyperparameter vector $\beta_{k,i}$ as:

$$\beta_{k,i} := \lambda_k \eta_k, \quad i = 1, \cdots, N, \tag{14}$$

where $\lambda_k (\in \mathbf{R}) > 0$, $\eta_k := (\eta_{k,1}, \cdots, \eta_{k,M_k})$, $0 < \eta_{k,i} < 1$, and $\sum_{i=1}^{M_k} \eta_{k,i} = 1$. Here, λ_k is the *commonality hyperparameter* describing the degree of commonality for the emission probabilities of $\{o_{k,t}\}_{t=1}^{T}$: $P(o_{k,t} \mid q_t, b_k)$ among different hidden states.[2] The hyperprameter η_k is a *common shape hyperparameter* that described the average shape of the emission probabilities $P(o_{k,t} \mid q_t, b_k)$ for different hidden states.

Here, we examine the effect of the commonality hyperparameter λ_k on the emission probability $b_{k,i}$. The shapes of the prior distribution (10) for various values of λ_k are shown in Figure 1. Fig. 1 (c) shows a case where λ_k is large. Here, the parameter vectors $\{b_{k,i}\}_{i=1}^{N}$, exhibit only small differences, i.e., $b_{k,1} \approx b_{k,2} \approx \cdots \approx b_{k,M_k} \approx \eta_k$, meaning that there is low dependency of $\{o_{k,t}\}_{t=1}^{T}$ on the states. For smaller λ_k on the other hand (Fig. 1 (a) or (b)), the diversity of $\{b_{k,i}\}_{i=1}^{N}$ among each state is greater; in other words, the dependency of $\{o_{k,t}\}_{t=1}^{T}$ on the states is not low.

B. Prior distribution for λ_k and η_k

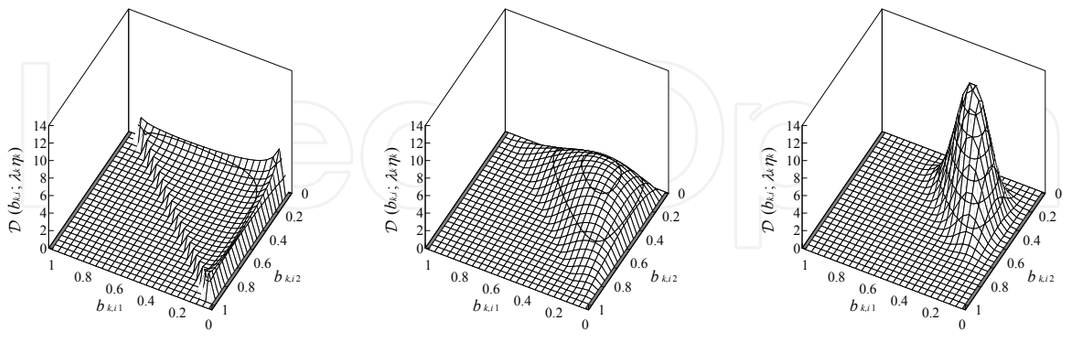
Here we describe the prior distribution of the hyperparameters λ_k and η_k used for learning these hyperparameters in a Bayesian learning method described later.

The commonality hyperparameter λ_k has no well-known "natural conjugate" prior distribution. Therefore, the prior distribution for λ_k is defined using only information in the

[1] This basic setting of the Dirichlet prior distribution makes it equivalent to a non-informative uniform prior distribution.

[2] The diversity of $P(o_{k,t} \mid q_t, b_k)$ among the states corresponds to that of $\{b_{k,i}\}_{i=1}^{N}$ because the emission probability of $o_{k,t}$: $P(o_{k,t} \mid q_t, b_k)$ is defined by using $\{b_{k,i}\}_{i=1}^{N}$, as shown in equation (5).

range $\lambda_k \in (0,\infty)$. Although there are a number of alternative prior distributions for a positive continuous variable (for example, the log-normal prior distribution), the prior distribution of λ_k is given by the following gamma prior distribution:



(a) λ_k = 2. (b) λ_k = 6. (c) λ_k = 18.

Fig. 1. Dirichlet prior distribution for $b_{k,i}$, for various values of the commonality hyperparameter λ_k. The parameters $\{b_{k,i}\}_{i=1}^{N}$ are 3D variables $b_{k,i} = (b_{k,i1}, b_{k,i2}, b_{k,i3})$, and the common shape hyperparameter η_k is constant, $\eta_k = (0.3, 0.3, 0.4)$. The component $b_{k,i3}$ is omitted because it can be determined from $b_{k,i3} = 1 - b_{k,i1} - b_{k,i2}$. This figure clearly shows that, for larger λ_k, the parameters $\{b_{k,i}\}_{i=1}^{N}$ concentrate more around the average η_k.

$$P(\lambda_k) := \mathcal{G}(\lambda_k; \kappa, \omega), \tag{15}$$

where $\mathcal{G}(\cdot; \kappa, \omega)$, is the gamma distribution having shape parameter κ and scale parameter ω.[3] These hyperhyperparameters are set to $\kappa = 1.0$ and $\omega = 100$ in the experiments described in Sec. 4, which allows λ_k to be widely distributed within in its available range.

There is also no known "natural conjugate" prior distribution for η_k. However, there are a limited number of options for the prior distribution because of the constraints of η_k, namely, $\sum_{i=1}^{M_k} \eta_{k,i} = 1$ and $0 < \eta_{k,i} < 1$. Therefore, we use the Dirichlet distribution as the prior distribution for η_k:

$$P(\eta_k) := \mathcal{D}(\eta_k; \eta_0), \tag{16}$$

where η_0 denotes the hyperhyperparameter vector. By considering a non-informative

[3]The gamma distribution is defined as $\mathcal{G}(x; \kappa, \omega) := \frac{x^{\kappa-1} \exp(-\omega^{-1} x)}{\omega^{\kappa} \Gamma(\phi)}$, where $\Gamma(\cdot)$ is the gamma function.

setting for η_k, the vector η_0 is set to $\eta_0 = (1.0, \cdots, 1.0)$ in the experiments described later.

Fig. 2 graphically summarizes the model specifications described in this section.

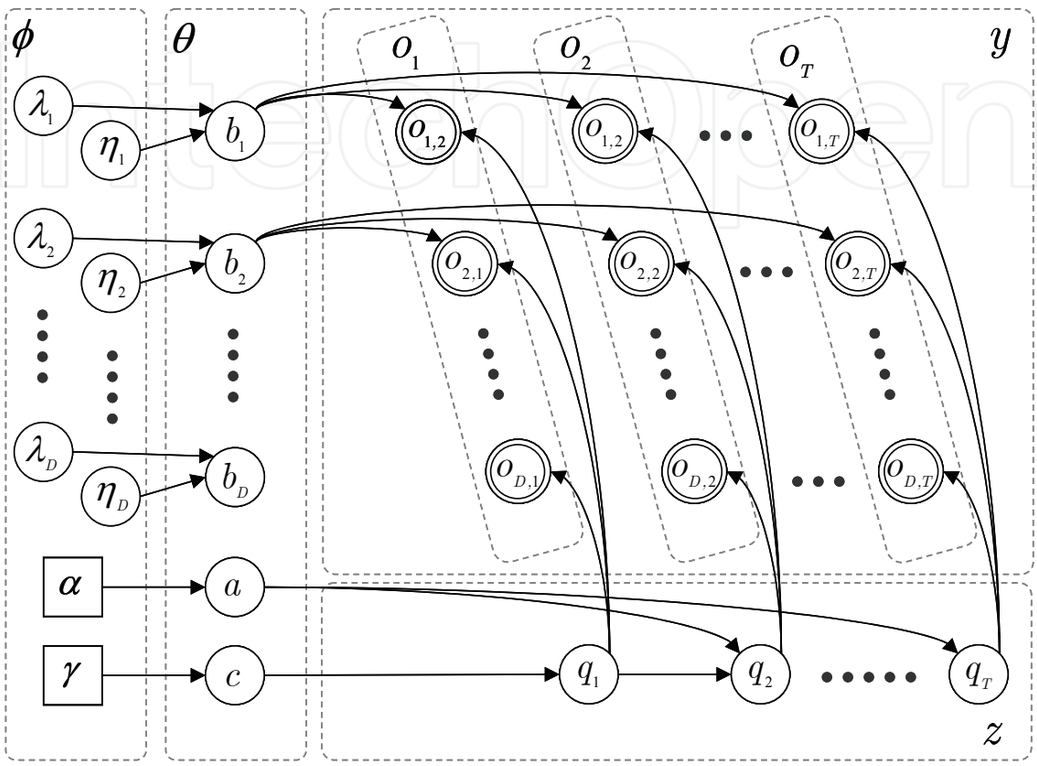


Fig. 2. Graphical representation of the model. The double circles are observable probabilistic variables, and the single circles are unobservable probabilistic variables. The squares are the fixed variables, the arrows probabilistic dependencies between variables, and the dashed lines groups of variables. Hyperhyperparameters and their dependencies are omitted for clarity.

3. Bayesian learning for the model

We define a training dataset Y as the set of time-series data sequences $\{y_l\}_{l=1}^{L}$, where L is the number of sequences and l is the index of the sequence. The goal of Bayesian learning is, given the training dataset Y and the above model, to evaluate the (joint) posterior distribution for θ and ϕ:

$$P(\theta, \phi \mid Y) = \sum_{Z} P(\theta, \phi, Z \mid Y), \tag{17}$$

where

$$P(\theta,\phi,Z\mid Y)=\frac{P(Y\mid Z,\theta)P(Z\mid\theta)P(\theta\mid\phi)P(\phi)}{\sum_{Z}\int\int P(Y\mid Z,\theta)P(Z\mid\theta)P(\theta\mid\phi)P(\phi)d\theta d\phi}, \tag{18}$$

and Z is the set of hidden variable sequences $\{z_l\}_{l=1}^{L}$, corresponding to the dataset Y.

3.1 Implementation with MCMC

The integrations in equation (18) have no closed-form analytical solution, because of their complexity. Monte Carlo methods can generate samples from the posterior distribution (17), and we therefore adopt this approach. [4]Once the samples $\{(\theta^{(r)},\phi^{(r)})\}_{r=1}^{R}$ are generated, it is an easy matter to approximate the posterior distribution (17) with

$$P(\theta,\phi\mid Y)\approx\frac{1}{R}\sum_{r=1}^{R}\delta((\theta,\phi)-(\theta^{(r)},\phi^{(r)})), \tag{19}$$

where $\delta(\cdot)$ is the Dirac delta function, R is the number of samples, and r is the index of the sample. Fig. 3 summarizes the procedure used in our implementation.

3.2 Model evaluation

We introduce a fitness score as a metric to evaluate the degree of fitness between a set of test data sequences Y_{NEW} and the trained model:

$$\mathrm{Score}(Y_{NEW}):=\log P(Y_{NEW}\mid Y), \tag{20}$$

Here, $P(Y_{NEW}\mid Y)$ is the (conditional) marginal likelihood, that is, the likelihood function $P(Y_{NEW}\mid\theta)$ averaged over the posterior distribution $P(\theta,\phi\mid Y)$:

$$P(Y_{NEW}\mid Y)=\int\int P(Y_{NEW}\mid\theta)P(\theta,\phi\mid Y)d\theta d\phi. \tag{21}$$

Using the Monte Carlo approximation (19), we can approximate this marginal likelihood as

$$P(Y_{NEW}\mid Y)\approx\frac{1}{R}\sum_{r=1}^{R}P(Y_{NEW}\mid\theta^{(r)}). \tag{22}$$

[4] Specifically, we consider the joint posterior distribution (18) for generating the samples using an MCMC technique based on that in (Scott, 2002). By discarding samples of Z after taking the samples of (θ,ϕ,Z) from the joint posterior distribution (18), it becomes relatively straightforward to obtain samples of θ and ϕ from the posterior distribution (17).

Implementation using MCMC methods

(a) Initialization step:
Initialize $\theta^{(0)}$ and $\phi^{(0)}$ by sampling. $\psi^{(0)}$ is generated from the prior distribution, whereas $\theta^{(0)}$ is generated uniformly within the range of θ.

(b) MCMC step:
For g = 1 to G, repeat the following:
(i) Generate the g-th sample of Z by with the forward-backward sampling method (Scott, 2002).
(ii) Generate the g-th sample of θ using the Gibbs sampling method (Scott, 2002 ; Geman & Geman, 1984).
(iii) Generate the g-th sample of ϕ using the Metropolis-Hastings method (Hastings, 1970). [5]

(c) Selection step:
For the Monte Carlo approximation (19), select the sample set $\{\theta^{(r)}\}_{r=1}^{R}$ from $\{\theta^{(g)}\}_{g=1}^{G}$. [6]

Fig. 3. MCMC implementation.

4. Experiments

4.1 Artificial dataset experiment

We conducted an experiment using artificial datasets to evaluate our extended model. These datasets contain state-independent variables serving as redundant components.

A. Target HMM

In this experiment, we used multi-dimensional data sequences, each data component having 5 symbols. We generated these sequences from a 5-state ergodic HMM in which the hidden variable transition parameter a^* and the initial hidden variable parameter π^* were retained:

[5] In actual implementation, a well-known strategy to improve the acceptance rates is to apply the Metropolis-Hastings method separately to each hyperparameter λ_k and hyperparameter vector η_k. We use proposal distributions designed on the basis of information from the model, because this approach also improves the efficiency of the Metropolis-Hastings method in many cases.
We show details of the designed proposal distributions in the appendix.

[6] In the MCMC method, it is usually necessary to discard the initial samples. In the experiments described in Sec. 4, we generated 1000 samples in the MCMC step (b) (G = 1000), and we used the last 500 samples for the Monte Carlo approximation (R = 500).

$$a^* = \begin{bmatrix} 0.70 & 0.10 & 0.05 & 0.05 & 0.10 \\ 0.10 & 0.70 & 0.10 & 0.05 & 0.05 \\ 0.05 & 0.10 & 0.70 & 0.10 & 0.05 \\ 0.05 & 0.05 & 0.10 & 0.70 & 0.10 \\ 0.10 & 0.05 & 0.05 & 0.10 & 0.70 \end{bmatrix}, \quad \pi^* = \begin{bmatrix} 0.20 \\ 0.20 \\ 0.20 \\ 0.20 \\ 0.20 \end{bmatrix}^T.$$

The column and row numbers of the matrix representing parameter a^* are the next and current values of the hidden variable. The column number in the matrix representing parameter π^* is equivalent to the index of the initial hidden variable. We explain the emission probabilities of the target HMM in detail in the following.

B. State-dependent and state-independent components

In this experiment, we considered the following two probability matrices, b^*_{DEP} and b^*_{IND}, for the emission probability parameter of the k-th data component, b^*_k:

$$b^*_{DEP} = \begin{bmatrix} 0.50 & 0.20 & 0.05 & 0.05 & 0.20 \\ 0.20 & 0.50 & 0.20 & 0.05 & 0.05 \\ 0.05 & 0.20 & 0.50 & 0.20 & 0.05 \\ 0.05 & 0.05 & 0.20 & 0.50 & 0.20 \\ 0.20 & 0.05 & 0.05 & 0.20 & 0.50 \end{bmatrix}, \tag{23}$$

$$b^*_{IND} = \begin{bmatrix} 0.20 & 0.20 & 0.20 & 0.20 & 0.20 \\ 0.20 & 0.20 & 0.20 & 0.20 & 0.20 \\ 0.20 & 0.20 & 0.20 & 0.20 & 0.20 \\ 0.20 & 0.20 & 0.20 & 0.20 & 0.20 \\ 0.20 & 0.20 & 0.20 & 0.20 & 0.20 \end{bmatrix}. \tag{24}$$

Here, the column number represents the index of the hidden state, and the row number represents the index of the observable symbols. It should be noted that the matrix b^*_{IND} contains identical probability vectors in each column (state); in other words, the components with b^*_{IND} state-independent, whereas the components with b^*_{DEP} are state-dependent. Using these matrices, we considered the following 5 cases with different numbers of components:

(i) $b^*_1 = b^*_2 = b^*_{DEP}$,

(ii) $b^*_1 = b^*_2 = b^*_{DEP}$ and $b^*_3 = b^*_{IND}$,

(iii) $b^*_1 = b^*_2 = b^*_{DEP}$ and $b^*_3 = b^*_4 = b^*_{IND}$,

(iv) $b^*_1 = b^*_2 = b^*_{DEP}$ and $b^*_3 = b^*_4 = b^*_5 = b^*_{IND}$,

(v) $b^*_1 = b^*_2 = b^*_{DEP}$ and $b^*_3 = \cdots = b^*_6 = b^*_{IND}$.

The first 2 components in each case are state-dependent.

C. Model settings

In each of the cases described above, we trained and tested the extended model using various datasets containing 10 independent sequences (T = 100) generated from the target HMM. We also trained and tested a conventional Bayesian HMM with fixed hyperparameters with the same datasets for comparison.[7] We trained the conventional model using an MCMC implementation based on (Scott, 2002). In our extended model and in the conventional model, we set the number of hidden states to N = 5, i.e. the same number of hidden states as that of the target HMM.

D. Results

Figure 4 shows the averaged differences between the fitness score (20) of the extended model and that of the conventional model. When all components were state-dependent (case (i)), the extended model performed slightly worse than the conventional model. When the training dataset contained state-independent components (cases (ii) to (v)), however, the extended model performed better than the conventional one, as indicated by the higher averaged score differences as the number of state-independent components increased. This result demonstrates that the extended model is robust against state-independent components.

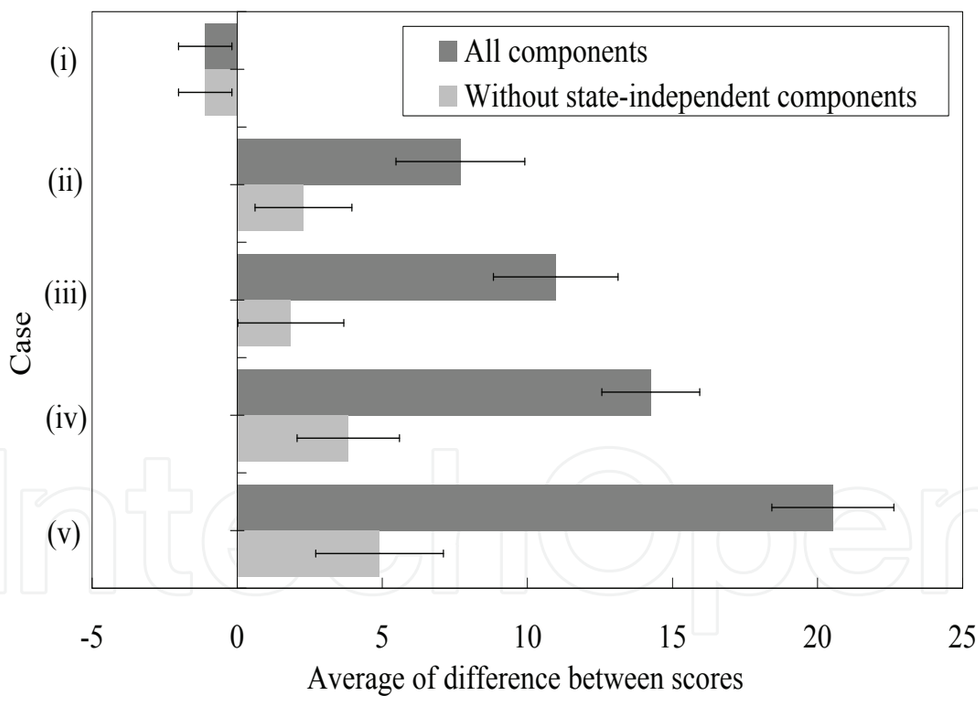


Fig. 4. The differences between scores (extended model score minus conventional model score) for all components of the test datasets. Differences based on the scores for the first two components (state-dependent components) of the test datasets are also shown. Differences of 10 independent trials were averaged. The error bars indicate the standard error.

[7] Each component of the hyperparameters is fixed at 1.0.

4.2 Soccer dataset experiment

We used real-world datasets with an additional irrelevant component to verify the performance of our extended model. This experiment was designed to demonstrate the ability of our model to discriminate the irrelevant component by Bayesian modeling with the commonality hyperparameter λ_k . This hyperparameter is closely related to the redundancy (state-independency) of a particular data component $o_{k,t}$.

This is a preliminary experiment a project involving event detection of Bayesian modeling for soccer games (Motoi et al., 2007).

A. Target data sequence

In our previous work (Motoi et al., 2007), the original dataset consisted of data sequences for 5 half-games of soccer. Each sequence was composed of 27-dimensional time-series data obtained from the position sequences of players. These positions were automatically extracted from video images by tracking the players using a method based on that in (Misu et al., 2005 ; Misu et al., 2002).

We used the sequence for only 1 half-game (length T = 2390) for the sake of simplicity. This sequence contained only 6 selected components and 1 additional component.

B. Selected and additional components

We used the following 6 selected variables for modeling: (a) the center of all players in the x direction; (b) the center of all players in the y direction; (c) the center of the left team players in the x direction; (d) the center of the left team players in the y direction; (e) the center of the right team players in the x direction; and (f) the center of the right team players in the y direction. We also added another variable to the target data sequence as the irrelevant component: (g) the x center of all the players in another half-game. The x and y directions correspond to the long axis and short axis of the playing field, respectively.

C. Model settings

In modeling the target data, we discretized all components in the extended model into 10 symbols (in other words, M_k = 10 for all components). We also set the number of hidden states to N = 10. Two examples of the discretized data components are shown in Figure 5.

D. Results

Boxplots of the commonality hyperparameter samples generated from the posterior distribution are shown in Figure 6 (18). The irrelevant component (g) has the largest hyperparameter λ_k , suggesting the possibility of discriminating irrelevant components by using the hyperparameters $\{\lambda_k\}$.

5. Application to real event detection

The results described in the previous section demonstrated the capability of our extended model in an event detection problem in soccer games (Motoi et al., 2007). In this section, we apply the extended model to event detection in sports videos. Our goal here is to detect

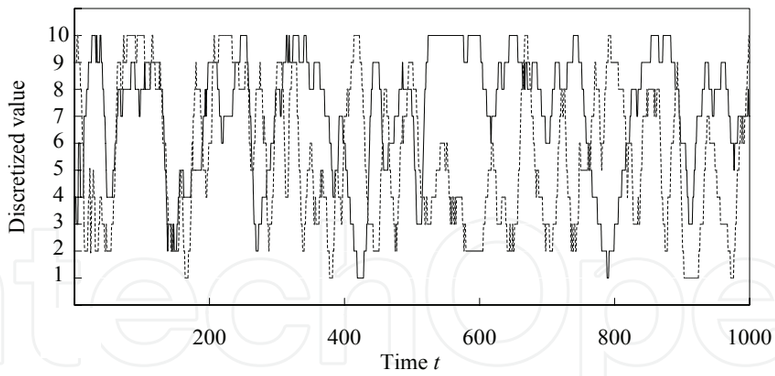


Fig. 5. Trajectories of the (discretized) variables (a) and (b), plotted in the range t = 1 to 1000 for clarity. The solid line is (a), and the dotted line is (b).

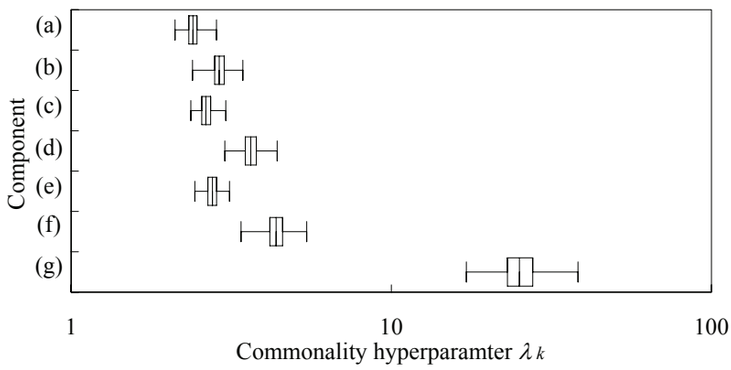


Fig. 6. Boxplots of the commonality hyperparameters $\{\lambda_k\}$ (500 samples) in Experiment 4.2. The smallest sample, lower quartile, median, upper quartile, and largest sample are shown for each λ_k .

target events from data sequences. Such events include kick offs, corner kicks, free kicks, throw ins, and goal kicks. Details of the data sequences are described in the following.

5.1 Modeling with a given data sequence

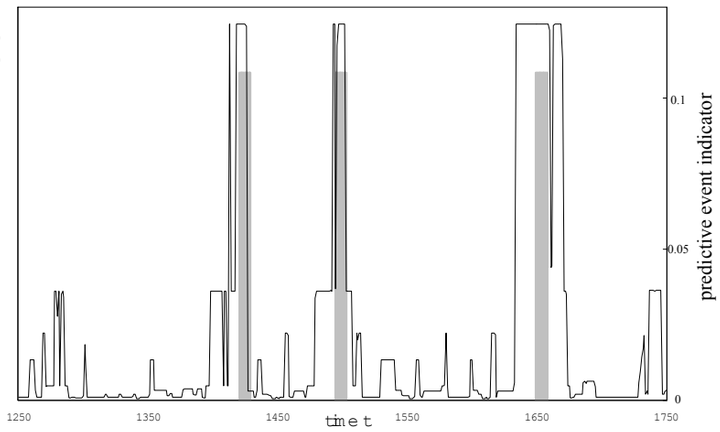
In this modeling, the raw dataset consisted of the positions of all players, which were automatically extracted from videos of 7 half games. Forty components associated with each target event were contained in the given data sequence.[8] We trained both the conventional and extended HMMs using the sequences for the 40 associated component in all 7 half games.

5.1.1 Demonstration

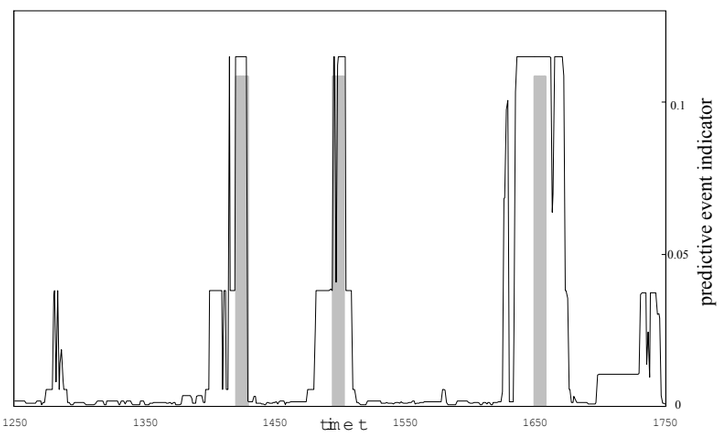
In this section, we show the predicted results for a corner kick event in another half game. This half game was independent from the 7 half games used to train the HMMs. Examples

[8]First, 1065 candidate components were generated from players' positions. We then selected the 40 associated components using standard information-based criteria showing the degree of the association with each event.

of the predicted results with our extended model and the conventional in (Motoi et al., 2007) are shown in Fig. 7. Actual events are indicated in gray. These results show that the conventional model gives more false alerts compared with the extended model, indicating the capability of the extended model to reduce the negative influence of redundant components in the 40 given components.



(a) Extended model.



(b) Conventional model.

Fig. 7. Predicted results for corner kick event. The range t = 1250 to 1750 contains 3 of the 4 target events in this half game. The regions of actual events are shown in gray.

6. Conclusions

In this chapter, we have described an extended Bayesian HMM for multidimensional discrete data sequences including redundant components. For the extended model, we also described an implementation of Bayesian learning based on a Markov chain Monte Carlo scheme. We evaluated the performance of the extended model with this implementation using two example datasets. We also demonstrated its application to an event detection problem with 40-dimensional data sequences extracted from videos of actual soccer games.

Our results showed that the extended Bayesian HMM has reasonable performance in the presence of redundant components in the data.

7. Acknowledgements

The authors greatly appreciate insightful comments from Prof. A. Doucet. The authors also thank A. Matsui, S. Clippingdale, I. Yamada, and M. Takahashi of NHK for their help and advice. T. Kaburagi, H. Sasaki, and J. Maruyama at Waseda University also greatly contributed to this study.

Appendix: Proposal distributions

The proposal distribution can be any probability distribution so long as certain conditions are satisfied. The design of the proposal distribution, however, strongly affects the efficiency. When applying the Metropolis-Hastings method to each variable separately, a promising approach is to employ the full conditional (posterior) distribution as the proposal distribution $\mathrm{Q}(\cdot)$. [9] However, it is difficult to use the full conditional distributions of λ_k and η_k as their proposal distributions in the model, because these distributions do not belong to any standard families of probability density functions having known direct sampling methods. Therefore, we use proposal distributions designed based on information from the full conditional distributions.

A. Proposal distribution of λ_k

The full conditional distribution of λ_k is

$$P\left(\lambda_k \mid Y, Z, \theta, \{\eta_{k'}\}, \{\lambda_{k'}\}_{k' \neq k}\right) = P\left(\lambda_k \mid b_k, \eta_k\right). \tag{25}$$

Applying the log-normal distribution $\mathcal{LN}(\cdot)$, the full conditional distribution (25) can be approximated by:

$$P\left(\lambda_k \mid b_k, \eta_k\right) \approx \mathcal{LN}\left(\lambda_k; \mu\left(\lambda_k^{(g-1)}\right), \nu\left(\lambda_k^{(g-1)}\right)\right). \tag{26}$$

Here,

$$\mu\left(\lambda_k\right) := \log\lambda_k + l_k''\left(\lambda_k\right)^{-1} l_k'\left(\lambda_k\right), \quad \nu\left(\lambda_k\right) := l_k''\left(\lambda\right)^{-1}$$
$$l_k\left(\lambda_k\right) := \log P\left(\log\lambda_k \mid b_k, \eta_k\right) = \log P\left(\lambda_k \mid b_k, \eta_k\right) + \log\lambda_k,$$

$l_k'(\cdot)$ is the first-order derivative of $l_k(\cdot)$, and $l_k''(\cdot)$ is its second-order derivative. However, the approximation (26) is not valid when the previous sample $\log\lambda_k^{(g-1)}$ is far from the peak

[9] In this scenario, the Metropolis-Hastings algorithm is completely "rejection-less"; in other words, it is identical to Gibbs sampling.

of $P(\log\lambda_k \mid b_k, \eta_k) = \lambda_k P(\lambda_k \mid b_k, \eta_k)$. In a number of preliminary numerical experiments, this proposal distribution (26) showed low acceptance rate. Slightly expanding the logarithm variance of the proposal distribution is a simple way to improve the low acceptance rate in such cases. Thus

$$Q(\lambda_k;\cdot) := \mathcal{LN}\left(\lambda_k; \mu\left(\lambda_k^{(g-1)}\right), (1+\varepsilon)v\left(\lambda_k^{(g-1)}\right)\right), \tag{27}$$

where $\varepsilon(\geq 0)$ is a user-settable variable. In the experiments described in this chapter, we used the proposal distribution (27) with $\varepsilon = 0.2$. This gave reasonable and stable performance in our preliminary experiments.

B. Proposal distribution of η_k

The full conditional distribution of η_k is

$$P(\eta_k \mid Y, Z, \theta, \{\eta_{k'}\}_{k' \neq k}, \{\lambda_{k'}\}) = P(\eta_k \mid b_k, \lambda_k). \tag{28}$$

It is difficult to approximate the distribution (28) itself with basic methods. Therefore, we consider only a rough approximation of the center of the distribution (28) in this study.
When the parameter b_k is given, one of the simplest estimators for the common (average) shape of $\{b_{k,i}\}$ is $\overline{\eta}_k(b_k) = \frac{1}{N}\sum_{i=1}^{N} b_{k,i}$. We assume that the center of the distribution (28) can be roughly approximated by this estimator $\overline{\eta}_k(b_k)$. In view of this assumption and the simplicity of the implementation, we use the Dirichlet proposal distribution centered on $\overline{\eta}_k(b_k)$:

$$Q(\eta_k;\cdot) := \mathcal{D}(\eta_k; \nu\overline{\eta}_k(b_k)). \tag{29}$$

Here, $\nu(>0)$ is a user-settable variable. In this study, we set $\nu = 100$, which resulted in reasonable performance in a number of preliminary numerical experiments.

8. References

Funada, A.; Sasaki, H.; Nakada, Y.; Matsumoto, T. (2005). Monte Carlo HMM for on-line handwriting recognition. *Proc. 15th Jpn. Neural Netw. Soc. Ann. Conv.* (*JNNS2005*), pp. 137–138, Sep. 2005 [in Japanese].

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721-741, Nov. 1984.

Günter, S. & Bunke, H. (2003). Fast feature selection in an HMM-based multiple classifier system for handwriting recognition. *Proc. 25th DAGM Symp. Pattern Recognit.* (*DAGM2003*), pp. 289–296, Sep. 2003.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, vol. 57, no. 1, pp. 97-109, Apr. 1970.

Huo, Q.; Chan, C.; Lee, C-H. (1995). Bayesian adaptive learning of the parameters of hidden Markov model for speech recognition. *IEEE Trans. Speech Audio Process.*, vol. 3, no. 5, pp. 334–345, Sep. 1995.

Kaburagi, T.; Muramatsu, D.; Matsumoto, T. (2007). Transmembrane protein structure predictions with hydropathy index/charge two-dimensional trajectories of stochastic dynamical systems. *J. Bioinform. Comput. Biol.*, 2007 (in press).

Krogh, A.; Larsson, B.; Heijne, G.; Sonnhammer, E. (2001). Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J. Mol. Biol.*, vol. 305, no. 3, pp. 567–580, Jan. 2001.

MacKay, D. J. C. (1992). A practical Bayesian framework for backpropagation networks. *Neural Comput.*, vol. 4, no. 3, pp. 448–472, May 1992.

MacKay, D. J. C. (1992). *Bayesian modeling and neural networks*, PhD thesis, Dept. of Computation and Neural Systems, CalTech, 1992.

MacKay, D. J. C. (1997). Ensemble learning for hidden Markov models. Technical report, Cavendish Laboratory, University of Cambridge, 1997.(Available from http://wol.ra.phy.cam.ac.uk/mackay/)

Matsumoto, T.; Nakajima, Y.; Saito, M.; Sugi, J.; Hamagishi, H. (2001). Reconstructions and predictions of nonlinear dynamical systems: a hierarchical Bayesian approach. *IEEE Trans. Signal Process.*, vol. 49, no. 9, pp. 2138–2155, Sep. 2001.

Misu, T.; Naemura, M.; Zheng, W.; Izumi, Y. and Fukui, K. (2002). Robust Tracking of Soccer Players Based on Data Fusion. *Proc. 16th Int. Conf. Pattern Recognit.* (*ICPR2002*), vol. 1, pp. 556-561, Aug. 2002

Misu, T.; Takahashi, M.; Gohshi, S.; Tadenuma, M.; Fujita, Y. and Yagi, N. (2005). Visualization of offside lines based on realtime video processing. *IEICE Trans.*, vol. J88-D-II, no. 8, pp. 1681-1692, Aug. 2005 [in Japanese].

Motoi, S.; Misu, T.; Nakada, Y.; Matsumoto, T.; Yagi, N. (2007). Bayesian hidden Markov model approach for events detection in sports movie. *Spec. Interest Group Notes of Inf. Process. Soc. Jpn.* (*IPSJ SIG Notes*), vol. 2007, no. 1, pp. 133-139, Jan. 2007 [in Japanese].

Nakada, Y.; Matsumoto, T.; Kurihara, T.; Yosui, K. (2005). Bayesian reconstructions and predictions of nonlinear dynamical systems via the hybrid Monte Carlo scheme. *Signal Process.*, vol. 85, no. 1, pp. 129–145, Jan. 2005.

Neal, R. M. (1996). *Bayesian learning for neural networks*, Lecture Notes in Statistics 118, Springer-Verlag, New York, USA, Aug. 1996.

Nouza, J. (1996). Feature selection methods for hidden Markov model-based speech recognition. *Proc. 13th Int. Conf. Pattern Recognit.* (*ICPR96*), vol. 2, pp. 186– 190, Aug. 1996.

Qi, Y.; Minka, T.; Picard, R.; Ghahramani, Z. (2004). Predictive automatic relevance determination by expectation propagation. *Proc. 21st Int. Conf. Mach. Learn.* (*ICML2004*), pp. 85–92, Jul. 2004.

Scott, S. L. (2002). Bayesian methods for hidden Markov models: recursive computing in the 21st century. *J. Am. Stat. Assoc.*, vol. 97, no. 457 pp. 337–351, Mar. 2002.

Tipping, M. E. (2000). The relevance vector machine. *Adv. Neural Inf. Process. Syst.*, vol. 12, pp. 652–658, Jun. 2000.

Tusnady, G.; Simon, I. (1998). Principles governing amino acid composition of integral membrane proteins: application to topology prediction. *J. Mol. Biol.*, vol. 283, no. 2, pp. 489–506, Oct. 1998.

Yasuda, H.; Takahashi, K.; Matsumoto, T. (2000). A discrete HMM for online handwriting recognition. *Int. J. Pattern Recognit. Artif. Intell.*, vol. 14, no. 5, pp. 675–688, Aug. 2000.

Frontiers in Robotics, Automation and Control

Edited by Alexander Zemliak

ISBN 978-953-7619-17-6
Hard cover, 450 pages
Publisher InTech
Published online 01, October, 2008
Published in print edition October, 2008

This book includes 23 chapters introducing basic research, advanced developments and applications. The book covers topics such us modeling and practical realization of robotic control for different applications, researching of the problems of stability and robustness, automation in algorithm and program developments with application in speech signal processing and linguistic research, system's applied control, computations, and control theory application in mechanics and electronics.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen