Bayesian event detection for sport games with hidden Markov model

**Shigeru Motoi · Toshie Misu · Yohei Nakada ·
Tomohiro Yazaki · Go Kobayashi ·
Takashi Matsumoto · Nobuyuki Yagi**

Abstract Event detection can be defined as the problem of detecting when a target event has occurred, from a given data sequence. Such an event detection problem can be found in many fields in science and engineering, such as signal processing, pattern recognition, and image processing. In recent years, many data sequences used in these fields, especially in video data analysis, tend to be high dimensional. In this paper, we propose a novel event detection method for high-dimensional data sequences in soccer video analysis. The proposed method assumes a Bayesian hidden Markov model with hyperparameter learning in addition to the parameter leaning. This is in an attempt to reduce undesired influences from ineffective components within the high-dimensional data. Implemention is performed by Markov Chain Monte Carlo. The proposed method was tested against an event detection problem with sequences of 40-dimensional feature values extracted from real professional soccer games. The algorithm appears functional.

Keywords Event detection · Sports video analysis · Hidden Markov model · Bayesian learning · Metadata

S. Motoi · Y. Nakada · T. Yazaki · G. Kobayashi (✉) ·
T. Matsumoto
Graduate School of Science and Engineering, Waseda
University, 3-4-1, Okubo, Shinjuku-ku, Tokyo 169-8555, Japan
e-mail: kobayashi09@matsumoto.eb.waseda.ac.jp

S. Motoi
e-mail: motoi06@matsumoto.elec.waseda.ac.jp

Y. Nakada
e-mail: yohei@matsumoto.elec.waseda.ac.jp;
nakada@ise.aoyama.ac.jp

T. Yazaki
e-mail: yazaki07@matsumoto.elec.waseda.ac.jp

T. Matsumoto
e-mail: matsumoto@matsumoto.elec.waseda.ac.jp

T. Misu · N. Yagi
NHK Science & Technology Research Laboratories,
1-10-11 Kinuta, Setagaya-ku, Tokyo 157-8510, Japan
e-mail: misu.t-ey@nhk.or.jp

N. Yagi
e-mail: yagi.n-iy@nhk.or.jp

Present Address:
Y. Nakada
School of Science and Engineering Aoyama Gakuin University,
5-10-1, Fuchinobe, Chuo-ku, Sagamihara-shi,
Kanagawa 252-5258, Japan

1 Introduction

1.1 Background

Information describing the content of data is often called "metadata". In the context of video data, metadata refers to data that is used to facilitate content-based retrieval [1]. For example, in the case of video data for a drama, the metadata may include captions (subtitles), bibliographic information about the actors, and so on. In the case of video data for a sports event, the metadata may consist of information about the players, important event scenes, and so on. In the past decade, the importance of such metadata has been increasing in the broadcasting community, because of its potential applications to various broadcasting services, program production systems, and so forth, e.g., [2] [3]. Against this backdrop, there has been an increasing demand for effective metadata creation particularly for unscripted programs [4, 5].

A successful approach for creating metadata of unscripted sports games, is to use a framework with an event detection-based system [6, 7]. Figure 1 shows an example of such a system based on Ref. [6].

In such event detection-based systems, the event detection method is required to predict when the target event has occurred from given time-series data which is typically a high-dimensional, probabilistically uncertain data sequence extracted from videos.

1.2 Purpose

This paper attempts to perform soccer game event detection for constructing metadata creating system using the hierarchical Bayesian Hidden Markov Model (HMM) [8, 9] where performance evaluation was not conducted. "Hierarchical" in the present context refers to a learning scheme for hyperparameters behind HMM parameters. Note that hyperparameters often play an important role in a Bayesian framework in that HMM parameter distribution at a particular set of hyperparameter values could be largely different from that with another set of hyperparamter values. Therefore, such hierarchical structure endows more flexibilities with an HMM, however, adjusting the hyperparameters atutomatically from the data becomes non-trivial. Recall that in a Bayeisan HMM with discrete states, discrete outputs, the likelihood functions are generally defined through multinomial distributions so that the prior distributions are often Dirichlet because of the natural conjugacy. Care needs to be exercised, however, in designing hyperparameter distributions behind Dirichlet. This study assumes a particular form of the hyperparameter prior distributions in such a way that irrelevant features are automatically suppressed. In this way, the predictive capabilities could be improved. Training of the HMM and predictions of
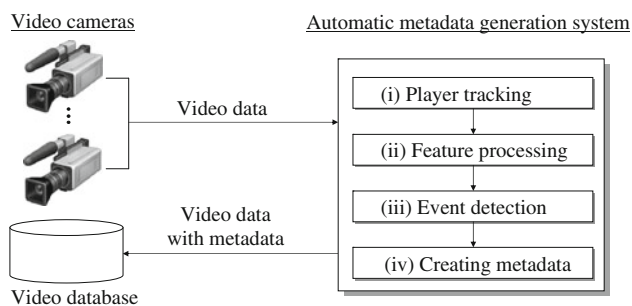

Fig. 1 An example of an automatic metadata creating system based on [6]. In this system, module (i) estimates the players' positions from the input video data sets filmed by several cameras. In module (ii), the feature vector sequence is extracted from the estimated players' positions. Module (iii) detects target events with the extracted sequence The metadata is created in module (iv) using the detected target events

event sequences are implemented by MCMC (Markov Chain Monte Carlo) method. The proposed method will be tested against 40-dimensional data sequences extracted from video data sets of professional soccer games. Performance will be evaluated with respect to three theoretical measures.

1.3 Related work

Since sports game data are generally sequential, and since HMM is a general model for capturing properties of sequential data, many of the papers on sports event detection are based on HMM [10–17]. Some of them use the maximum likelihood method where the Baum–Welch algorithm [18] is used for parameter estimation [19]. Many others use Bayesian framework.

There have been a variety of studies on HMMs with Bayesian learning (Bayesian HMMs), as well as their implementations. For instance, Ref. [20] describes a maximum a posteriori estimation for Bayesian HMMs. In Ref. [21], a variational Bayesian method is described for HMMs. A Bayesian HMM with a MCMC method is discussed in Ref. [22]. These standard Bayesian HMMs (often with several modifications for observation densities) have been successfully applied to many applications, including speech [20] and event detection [23].

The hierarchical Bayesian HMM to be described in the next section, is a generalized version of such standard Bayesian HMMs, where hyperparameter learning is performed behind the target HMM parameters.

2 Hierarchical Bayesian HMM for event detection

This section describes the hierarchical Bayesian HMM designed for the soccer game event detection problem. There are two points that we would like to address in this study.

1. The proposed model consists of discrete states and discrete outputs so that popular prior distributions within a Bayesian framework are Dirichlet since they are naturally conjugate with respect to the multinomial distributions. Recall that there are hyperparameteres behind Dirichlet distributions which control the properties of the distributions. Therefore, the prior distributions consist of a family of distributions parameterized by hyperparameters instead of a single distribution, which gives rise to more flexibility in a model. The proposed method attempts to *learn hyperparameteres* in addition to learn the target parameters of HMM. As such, prediction capabilities of Bayesian methods with

hyperparameter learning often outperform one without hyperparameter learning.

2. We will propose in this paper, a particular form of the hyperparameters for the HMM parameter prior distributions. More precisely, the hyperparameters are in a product form of two hyperparameters. One is the *commonality hyperparameter*, which describes the degree of commonality for the output emission probabilities among different hidden states. Another is the *common shape hyperparameter* that characterizes the average shape of the output emission probabilities among different hidden states. This prior structure enables us to suppress those output components that are ineffective for predictions. This is one of the desired properties when the output dimension is high. In the experiments to be reported below, output dimension is 40. Recall that there is an underlying topology behind HMMs. This paper assumes "ergodic" topology where every hidden state can go to any other state. Other topologies are also possible.

2.1 Data and hidden variables

Associated with an HMM, there are two sequential data. One is the observation data sequence $y := (o_1, \ldots, o_T)$ whereas another is the hidden variable (hidden state) sequence $z := (q_1, \ldots, q_T)$. Here o_t and q_t stand for time-series data and the hidden variable at time t, and T is the length of the sequence. The hidden variable q_t is a one-dimensional variable that can take a finite number of values among N states (i.e., $q_t \in \{1, \ldots, N\}$).

In event detection problems, the data o_t can consist of two types of variables, e_t and f_t. The variable e_t is a one-dimensional event variable at time t, and the variable $f_t := (f_{1,t}, \ldots, f_{L,t})$ is an L-dimensional feature variable at time t. Here, M_e is the number of target events ($e_t \in \{1, \ldots M_e\}$), and L is the dimension of the feature variable f_t. The variable $f_{l,t}$ represents the lth component of f_t, and M_{fl} is the number of symbols for $f_{l,t}$ (i.e., $f_{l,t} \in \{1, \ldots, M_{fl}\}$). In the experiment described later, the event variable e_t represents the occurrences of kick offs, corner kicks, and so on. The feature variable f_t consists of extracted values from players' positions at time t.

2.2 Observation model

Given the whole parameter set $\theta := (a, b, c)$ of an HMM, the probability of the data y_t can be written as

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

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

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

where a is a hidden variable transition probability, c is an initial hidden variable probability, and b is an emission probability of the data o_t. In event detection problems, the emission probability of the data $o_t = (e_t, f_t)$ in (2) is described by the following equations:

$$P(o_t|q_t,b) := P(e_t|q_t,b_e)P(f_t|q_t,b_f), \tag{4}$$

$$P(f_t|q_t,b_f) := \prod_{l=1}^{L} P(f_{l,t}|q_t,b_{f_l}), \tag{5}$$

where $b := (b_f, b_e)$ and $b_f := (b_{f_1}, \ldots, b_{f_L})$.

Consider the multinomial distribution for one data item be defined as $\mathcal{Multi}(x;p) := \prod_{i=1}^{K} p_i^{I(x=i)}$, where $\sum_{i=1}^{K} p_i = 1$, $p_i \geq 0$, and $I(\cdot)$ is an indicator function that returns 0 for *false* and 1 for *true*. Using this equation, the event emission probability $P(e_t|q_t, b_e)$ in (4) is written as

$$P(e_t|q_t,b_e) := \mathcal{Multi}(e_t;b_{e,q_t}), \tag{6}$$

where $b_e := (b_{e,1}, \ldots, b_{e,N})$ and $b_{e,i} := (b_{ei1}, \ldots, b_{eiM_e})$. The emission probability of the lth component $f_{l,t}$ in (5) is

$$P(f_{l,t}|q_t,b_{f_l}) := \mathcal{Multi}(f_{l,t};b_{f_l,q_t}), \tag{7}$$

where $b_{f_l} := (b_{f_l,1}, \ldots, b_{f_l,N})$ and $b_{f_l,i} := (b_{f_l,i1}, \ldots, b_{f_l,iM_{f_l}})$. The hidden variable transition probability $P(q_t|q_{t-1}, a)$ and the initial hidden variable probability $P(q_1|c)$ in (3) can be written as

$$P(q_t|q_{t-1},a) := \mathcal{Multi}(q_t;a_{q_{t-1}}), \tag{8}$$

$$P(q_1|c) := \mathcal{Multi}(q_1;c), \tag{9}$$

where $a := (a_1, \ldots, a_N)$, $a_i := (a_{i1}, \ldots, a_{iN})$, and $c := (c_1, \ldots, c_N)$.

2.3 Prior distribution for parameter set

Within a Bayesian framework, not only the observation model, but also the prior distribution of the parameter set is defined. In Bayesian HMMs, parameter independency in the prior distribution is assumed for simplicity of the implementation; that is:

$$P(\theta|\phi) = P(a|\alpha)P(b|\beta)P(c|\gamma), \tag{10}$$

$$P(a|\alpha) := \prod_{i=1}^{N} P(a_i|\alpha_i), \tag{11}$$

$$P(b|\beta) := P(b_e|\beta_e)P(b_f|\beta_f), \tag{12}$$

$$P(b_f|\beta_f) := \prod_{l=1}^{L} \prod_{i=1}^{N} P(b_{f_l,i}|\beta_{f_l,i}), \tag{13}$$

$$P(b_e|\beta_e) := \prod_{i=1}^{N} P(b_{e,i}|\beta_{e,i}), \tag{14}$$

where $\phi := (\alpha, \beta, \gamma), \alpha := (\alpha_1, \ldots, \alpha_N), \beta := (\beta_f, \beta_e), \beta_f := (\beta_{f_1}, \ldots, \beta_{f_L}), \beta_{f_l} := (\beta_{f_l,1}, \ldots, \beta_{f_l,N}), \beta_e := (\beta_{e,1}, \ldots, \beta_{e,N})$. Using the "naturally conjugate" Dirichlet prior distribution,[1] the prior distributions of $a_i, b_{f_l,i}, b_{e,i}$, and c in (10)–(14) are defined as follows:

$$P(a_i|\alpha_i) := \mathcal{D}ir(a_i; \alpha_i), \tag{15}$$

$$P(b_{e,i}|\beta_{e,i}) := \mathcal{D}ir(b_{e,i}; \beta_{e,i}), \tag{16}$$

$$P(b_{f_l,i}|\beta_{f_l,i}) := \mathcal{D}ir(b_{f_l,i}; \beta_{f_l,i}), \tag{17}$$

$$P(c|\gamma) := \mathcal{D}ir(c; \gamma), \tag{18}$$

where $\alpha_i := (\alpha_{i1}, \ldots, \alpha_{iN}), \beta_{e,i} := (\beta_{e,i1}, \ldots, \beta_{e,iN}), \beta_{f_l,i} := (\beta_{f_l,i1}, \ldots, \beta_{f_l,iN})$, and $\gamma := (\gamma_1, \ldots, \gamma_N)$.

2.4 Settings for hyperparameter set

In this paper, all components of the hyperparameter vectors except for $\beta_{f_l,i}$ are fixed at 1.0, like those of several conventional Bayesian HMMs (e.g., [7]).[2] On the other hand, to control negative influences from the components that have low dependency on the states (redundant components), this study considers a reparameterization of the hyperparameter vectors $\beta_{f_l,i}$, and the prior distribution of the reparameterized hyperparameters described below.

2.4.1 Reparameterization of $\beta_{f_l,i}$

In the hierarchical Bayesian HMM [8, 9], the reparameterization of the hyperparameter vector $\beta_{f_l,i}$ is described by the following equation:

$$\beta_{f_l,1} = \beta_{f_l,2} = \cdots = \beta_{f_l,N} := \lambda_{f_l}\eta_{f_l}. \tag{19}$$

The non-negative variable $\lambda_{f_l} \in R$ is the *commonality hyperparameter*, which describes the degree of commonality

for the emission probabilities $P(f_{l,i}|q_t, b_{f_l})$ among different hidden states. The hyperparameter $\eta_{f_l} := (\eta_{f_l,1}, \ldots, \eta_{f_l,M_{f_l}}) \in R^{M_{f_l}}$ is the *common shape hyperparameter* that defines the average shape of the emission probabilities $P(f_{l,t}|q_t, b_{f_l})$ for different hidden states, where $\eta_{f_l,i} > 0$, and $\sum_{i=1}^{M_{f_l}} \eta_{f_l,i} = 1$.

Here, let us take a closer look at how the commonality hyperparameter λ_{f_l} influences the emission probability $b_{f_l,i}$. Figure 2 describes the shapes of the prior distribution (13) with several settings of λ_{f_l}. First, consider a case where the commonality hyperparameter λ_{f_l} is large so that the probability mass is concentrated around a relatively narrow region, as shown in Fig. 2c, d. This amounts to the fact that there is a relatively small amount of diversity among the parameter vectors $b_{f_l,1}, \ldots, b_{f_l,N}$, i.e., $b_{f_l,1} \approx b_{f_l,2} \approx \cdots \approx b_{f_l,M_{f_l}} \approx \eta_{f_l}$. Therefore, low dependency of $f_{l,i}$ on the states is expected. If a component of the given data has little effect on the states, then that particular component may not carry useful information for prediction purposes, so that one wants to suppress such a component. On the other hand, when the commonality hyperparameter λ_{fl} is not so large (as shown in Fig. 2a, b), there would be more diversity of $b_{fl,i}$ among states, and hence, the dependency of $f_{l,i}$ on the states is expected.[3] By learning such reparameterized hyperparameters, the proposed hierarchical Bayesian HMM enables us to reduce negative influences from the redundant components.

2.4.2 Prior distribution for λ_{fl} and η_{fl}

This section describes the prior distributions of the hyperparameters λ_{fl} and η_{fl}, for learning these hyperparameters in a Bayesian framework to be described later. There is no well-known "naturally conjugate" prior distribution for the commonality hyperparameter λ_{fl}. Assuming that $\lambda_{fl} \in (0, \infty)$, there are several possible prior distribution for λ_{fl}. Among them, we assume the gamma distribution:

$$P(\lambda_{f_l}) := \mathcal{G}amma(\lambda_{f_l}; \kappa, \omega), \tag{20}$$

where $\mathcal{G}amma(\cdot; \kappa, \omega)$ stands for the gamma distribution with the non-negative shape parameter κ and the non-negative scale parameter ω.[4] In the experiments of Sect. 4, these hyperparameters are set to $\kappa = 0.5$ and $\omega = 100$, which enables λ_{fl} to be distributed widely in its range.

Naturally conjugate prior distribution for η_{fl} is not known either. However, because of the constraints of

---

[1] The Dirichlet distribution is described as $\mathcal{D}ir(x; d) := \frac{\Gamma(\sum_{i=1}^{K} d_i)}{\prod_{i=1}^{K} \Gamma(d_i)} \prod_{i=1}^{K} x_i^{d_i-1}$, where $x := (x_1, \ldots, x_K), \sum_{i=1}^{K} x_i = 1, x_i > 0, d := (d_1, \ldots, d_K)$, and $d_i > 0$. The function $\Gamma(\cdot)$ stands for the gamma function.

[2] This setting is one of the basic settings of the Dirichlet prior distribution, making it the same as a (non-informative) uniform prior distribution.

[3] Note that the emission probability $P(f_{l,t}|q_t, b_{fl})$ is defined through b_{fl}, as described in (7). Therefore, the diversity of $P(f_{l,t}|q_t, b_{fl})$ among the states corresponds to that of $\{b_{fl,i}\}_{i=1}^{N}$.

[4] The gamma distribution is defined as $\mathcal{G}amma(x; \kappa, \omega) := \frac{x^{\kappa-1} \exp(-\omega^{-1}x)}{\omega^\kappa \Gamma(\kappa)}$, where $\Gamma(\cdot)$ denotes the gamma function.
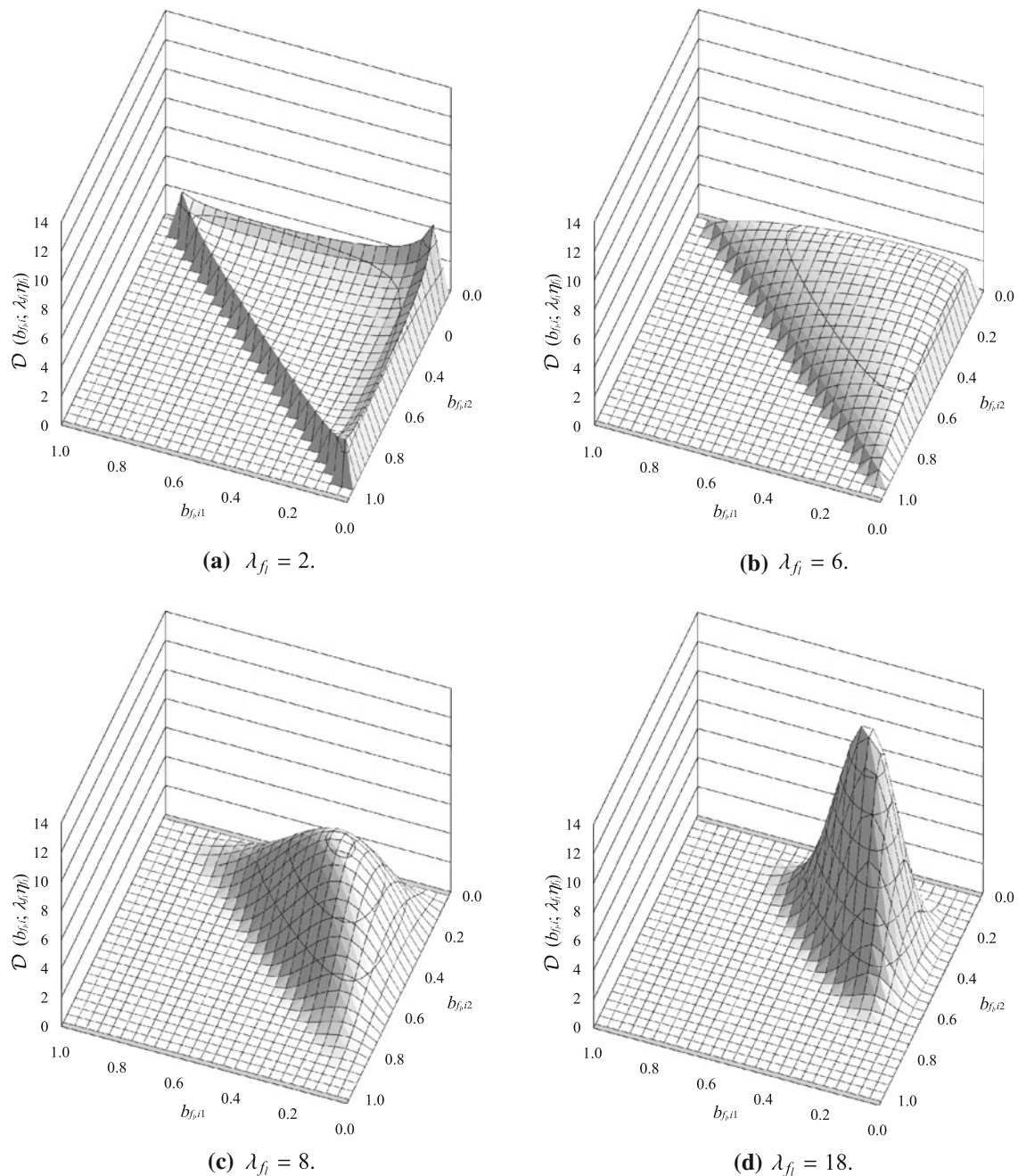
(a) $\lambda_{fl} = 2$.

(b) $\lambda_{fl} = 6$.

(c) $\lambda_{fl} = 8$.

(d) $\lambda_{fl} = 18$.

Fig. 2 The Dirichlet prior distribution for $b_{fl,i}$ with several settings of the commonality hyperparameter λ_{fl}. In this figure, the parameters $b_{fl,i}$ are three-dimensional variables, $b_{fl,i} = (b_{fl,i1}, b_{fl,i2}, b_{fl,i3})$, and the common shape hyperparameter η_{fl} is fixed at $\eta_{fl} = (0.3, 0.3, 0.4)$. Since the component $b_{fl,i3}$ can be determined automatically using $b_{fl,i3} = 1 - b_{fl,i1} - b_{fl,i2}$, the variable $b_{fl,i3}$ is omitted in this figure. It should be observed that the parameter $b_{fl,i}$ concentrates more around the average η_{fl} when the commonality hyperparameter λ_{fl} is larger

η_{fl}, the Dirichlet distribution is assumed in this paper as the prior distribution for η_{fl}, i.e.,

$$P(\eta_{fi}) := \mathcal{D}ir(\eta_{fi}; \eta_0), \tag{21}$$

where η_0 denotes the hyperparameter vector. In the experiments described in later sections, the vector η_0 is set to $\eta_0 = (1.0, \ldots, 1.0)$ by considering a non-informative setting for η_{fl}.

The model specifications described in this section can be summarized graphically as shown in Fig. 3.

3 Implementation of event prediction

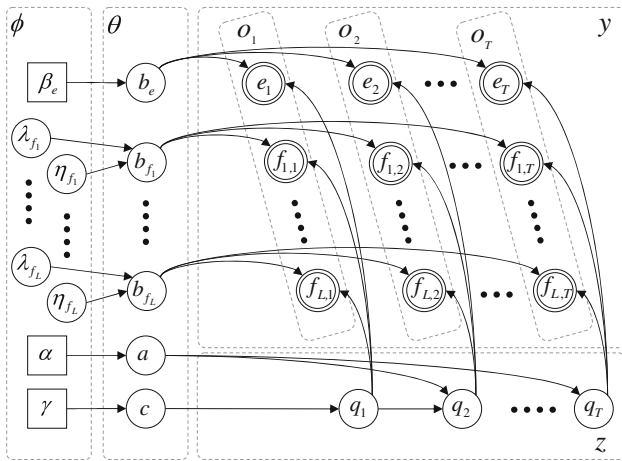The target problem of this paper can be formulated as follows.

Fig. 3 Graphical representation of the proposed model. In this figure, the *double circles* denote observable probabilistic variables, whereas the *single circles* are unobservable probabilistic variables. The *squares* denote the fixed variables, the *arrows* indicate probabilistic dependencies between variables, and the *dashed lines* show the groups of variables. For clarity, the hyperparameters and their dependencies are omitted here

Problem: Predict an event e_t^{new} in test data, when the feature variable sequence $f_{1:t}^{\text{new}} := (f_1^{\text{new}}, \ldots, f_t^{\text{new}})$ in the test data and a training data set Y are given.

In this problem, the training data set Y is considered as the set of time-series data sequences $\{y^d\}_{d=1}^D$, where D is the number of sequences, and d is the index of the sequence. Another data sequence $y^{\text{new}} := (o_1^{\text{new}}, \ldots, o_T^{\text{new}})$ is considered as test data, which is not included in the training data set Y. The test data y^{new} consists of two sequences: the event sequence $e_{1:T}^{\text{new}}$ and the feature variable sequence $f_{1:T}^{\text{new}}$, i.e., $y^{\text{new}} = (e_{1:T}^{\text{new}}, f_{1:T}^{\text{new}})$.

3.1 Bayesian predictive probability for target event

Within the Bayesian framework with the model described in Sect. 2, a reasonable approach for the target problem is to evaluate the predictive probability for the target event under the condition that the training data Y and the feature variable sequence f^{new} are available. Such predictive probability for the target event variable at time t is represented by

$$P(e_t^{\text{new}}|f_{1:t}^{\text{new}}, Y) = \int \int P(e_t^{\text{new}}|f_{1:t}^{\text{new}}, \theta) P(\theta, \phi|Y) \, d\theta \, d\phi, \tag{22}$$

via the evaluation of the joint posterior distribution:

$$P(\theta, \phi|Y) = \sum_Z P(\theta, \phi, Z|Y), \tag{23}$$

where

$$P(\theta, \phi, Z|Y) = \frac{P(Y|Z, \theta)P(Z|\theta)P(\theta|\phi)P(\phi)}{\sum_Z \int \int P(Y|Z, \theta)P(Z|\theta)P(\theta|\phi)P(\phi) \, d\theta \, d\phi}, \tag{24}$$

and Z stands for the set of hidden variable sequences $\{z^d\}_{d=1}^D$ corresponding to the data set Y.

3.2 Calculation of predictive probability

Because of their complexity, there is no closed-form analytical solution for the integrations in (22). Therefore, we use MCMC methods [22, 24, 25] to generate samples from the posterior distribution (23) of the hierarchical Bayesian HMM. Once the samples $\{(\theta^{(r)}, \phi^{(r)})\}_{r=1}^R$ are generated by the Monte Carlo method, the predictive probabilities (22) can be easily approximated as

$$P(e_t^{\text{new}}|f_{1:t}^{\text{new}}, Y) \approx \frac{1}{R} \sum_{r=1}^R P(e_t^{\text{new}}|f_{1:t}^{\text{new}}, \theta^{(r)}), \tag{25}$$

using only parameter samples $\{\theta^{(r)}\}_{r=1}^R$. Here, r stands for the index of the sample, and R is the number of samples. The conditional predictive probability $P(e_t^{\text{new}}|f_{1:t}^{\text{new}}, \theta^{(r)})$ in (25) can be calculated analytically by

$$P(e_t^{\text{new}}|f_{1:t}^{\text{new}}, \theta^{(r)}) = \sum_{q_t^{\text{new}}} P(e_t^{\text{new}}|q_t^{\text{new}}, b_e^{(r)}) P(q_t^{\text{new}}|f_{1:t}^{\text{new}}, \theta^{(r)}), \tag{26}$$

where q_t^{new} is the hidden state at time t, $b_e^{(r)}$ stands for the parameter b_e in the rth parameter sample $\theta^{(r)}$, and

$$P(q_t^{\text{new}}|f_{1:t}^{\text{new}}, \theta^{(r)}) = \frac{P(f_t^{\text{new}}|q_t^{\text{new}}, b_f^{(r)}) P(f_{1:t-1}^{\text{new}}, q_t^{\text{new}}|\theta^{(r)})}{\sum_{q_t^{\text{new}}} P(f_t^{\text{new}}|q_t^{\text{new}}, b_f^{(r)}) P(f_{1:t-1}^{\text{new}}, q_t^{\text{new}}|\theta^{(r)})}. \tag{27}$$

By considering only the feature variable sequence f^{new}, it is easy to compute the forward probability $P(f_{1:t-1}^{\text{new}}, q_t^{\text{new}}|\theta^{(r)})$ in (27) with the well-known forward procedure for HMMs, as follows:

$$P(f_{1:t-1}^{\text{new}}, q_t^{\text{new}}|\theta^{(r)}) = \sum_{q_{t-1}^{\text{new}}} P(f_{t-1}^{\text{new}}|q_{t-1}^{\text{new}}, b_f^{(r)}) P(q_t^{\text{new}}|q_{t-1}^{\text{new}}, a^{(r)})$$
$$P(f_{1:t-2}^{\text{new}}, q_{t-1}^{\text{new}}|\theta^{(r)}). \tag{28}$$

The implementation described in this section is summarized in Fig. 4.[5]

---

[5] In the actual implementation, the Metropolis–Hastings method is applied separately to each hyperparameter λ_{f_i} and hyperparameter vector η_{f_i}, which is a known basic strategy to improve the acceptance rates of the Metropolis–Hastings method. Since well-designed proposal distributions based on the information of the model also improve the efficiency of the Metropolis–Hastings method in many cases, this study uses proposal distributions designed from such information. The details of the designed proposal distributions are described in Appendix A.
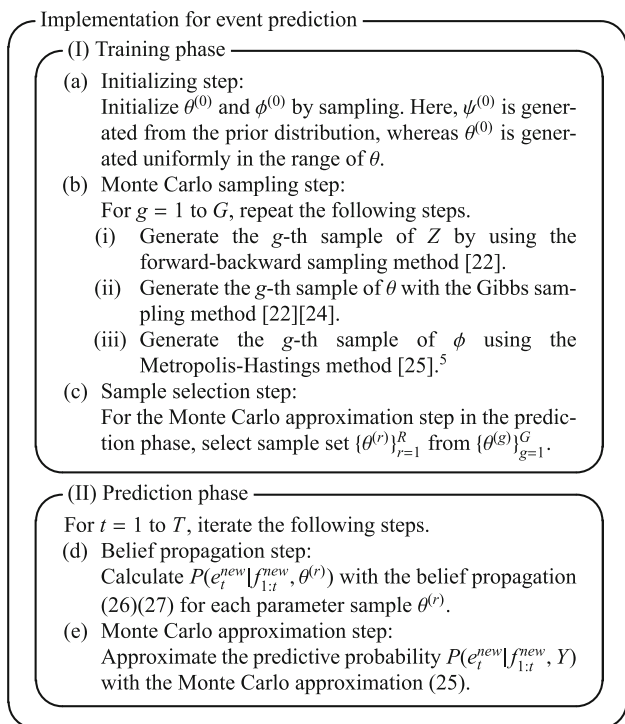
Implementation for event prediction

(I) Training phase

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

(b) Monte Carlo sampling step:
For $g = 1$ to G, repeat the following steps.
(i) Generate the g-th sample of Z by using the forward-backward sampling method [22].
(ii) Generate the g-th sample of θ with the Gibbs sampling method [22][24].
(iii) Generate the g-th sample of ϕ using the Metropolis-Hastings method [25].[5]

(c) Sample selection step:
For the Monte Carlo approximation step in the prediction phase, select sample set $\{\theta^{(r)}\}_{r=1}^{R}$ from $\{\theta^{(g)}\}_{g=1}^{G}$.

(II) Prediction phase

For $t = 1$ to T, iterate the following steps.
(d) Belief propagation step:
Calculate $P(e_t^{new}|f_{1:t}^{new}, \theta^{(r)})$ with the belief propagation (26)(27) for each parameter sample $\theta^{(r)}$.
(e) Monte Carlo approximation step:
Approximate the predictive probability $P(e_t^{new}|f_{1:t}^{new}, Y)$ with the Monte Carlo approximation (25).

Fig. 4 Procedure for event prediction

4 Event detection experiment for soccer games

To evaluate the metadata generation system in Fig. 1 in real situations, the proposed method was tested for the problem of event detection in soccer games. The details are described below.

4.1 Target data sequences

In this subsection, we explain the target data sequences of the event detection problem. Following Ref. [7], the target events are defined as Kick Off (KO), Corner Kick (CK), Free Kick (FK), Throw In (TI), and Goal Kick (GK). The event occurrence period is defined as a 9 s range centered on the time corresponding to the referee's instruction.

4.1.1 Target soccer games

For the data set in this experiment, we used videos of five half games in J-league (professional soccer league in Japan): the first four half games were used for training, and the last half game was used for testing. Table 1 shows basic information of the five target games and Fig. 5 shows example scenes of the video data set.

4.1.2 Players' positions

For all five half games, the players' position sequences were estimated from the corresponding video data set using the player tracking method in [26]. Figure 6 shows an example of the estimated players' position. By considering symmetries in the soccer games, reversed position sequences were also generated for the first four half games. More specifically, for each players' position sequence, we generated a non-reversed sequence, a long-side axis reversed sequence, a short-side axis reversed sequence, and a long-and-short-side axis reversed sequence for training. Thus, we used 16 players' position sequences for training, and 1 player's position sequence for testing.

4.1.3 Feature variables

Feature variable sequences, consisting of 40 components, were extracted from the players' position sequences. These 40 components were preliminarily selected from about five thousand candidates by applying a simple screening method with information-based criteria to the training data sets. The selected components included the average players' positions on the long-side axis (the X-coordinate) and the variance of the players' positions on the long-side axis. The features also included geometric information about the soccer field. The soccer field was divided into a $k_v \times k_h$ grid, and some of the features carry information about the grid square where that particular feature is located. Figure 7 schematically illustrates the case with $k_v = 3$ and $k_h = 5$. All components were quantized to 10 levels for HMM modeling. The details of the candidates and the screening method are described in Appendix B.

4.2 Settings

The proposed event detection method was evaluated using the feature variable sequences described above. The unit time for the HMM was 1 s, and the number of hidden states, N was, set at $N = 30$. For the computation described in Fig. 4, we generated 1,000 samples in the MCMC step (b) ($G = 1,000$), and we used the last 500 samples for the Monte Carlo approximation ($R = 500$) in this experiment.

Table 1 Basic information of target half games

#	Length [s]	Number of events					Training/testing
		KO	CK	FK	TI	GK	
1	2,390	2	1	14	27	9	For training
2	2,389	2	4	22	25	11	
3	2,454	1	3	4	24	20	
4	2,479	4	10	7	15	14	
5	2,460	3	1	10	19	10	For testing

Fig. 5 Example scene of the video data set. Two-directional cameras are used in this example. **a** Example scene of the video data set captured by the left video camera. **b** Example scene of the video data set captured by the right video camera
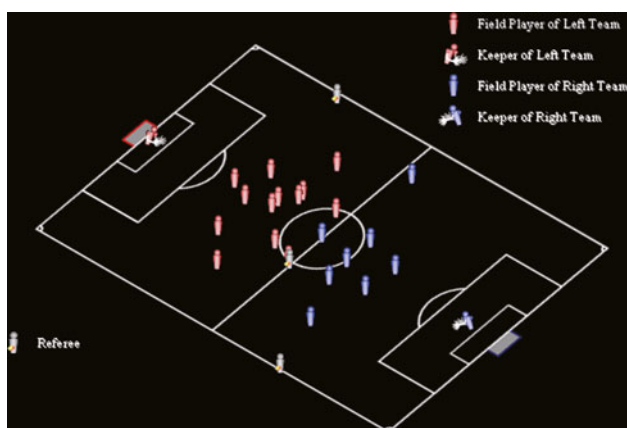

(a)

(b)

Fig. 6 Example of the estimated players' positions

The approximated predictive probability (25) was averaged over five independent trials to obtain a more accurate approximation.
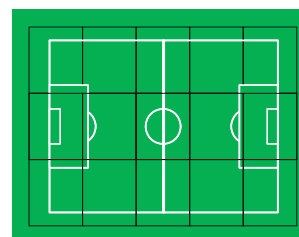

Fig. 7 Soccer field is divided into grid. This particular case is with 3×5 grid

4.3 Experimental results

The third column of Table 2 summarizes the Area Over ROC Curve (AOC) of each event, which is a detection error index defined with ROC curves drawn using the predictive probabilities for a specific event. Figure 8 shows

Table 2 Event AOC with test sequence

Event	Method		Difference
	without hyperparameter learning	with hyperparameter learning	
KO	6.539×10^{-2}	6.313×10^{-2}	2.261×10^{-3}
CK	8.659×10^{-3}	7.661×10^{-3}	9.973×10^{-4}
FK	5.121×10^{-1}	4.884×10^{-1}	2.374×10^{-2}
TI	2.028×10^{-1}	2.024×10^{-1}	4.484×10^{-4}
GK	1.639×10^{-1}	1.626×10^{-1}	1.336×10^{-3}

three examples of predictive probabilities estimated by the proposed method: (a) CK, (b) GK, and (c) FK.

One of the reasons for the low predictive capability of FK may be attributed to the fact that FK can take place without a game interruption. One of the possible improvement strategies could be taking into account the ball position which is not associated with the player positions.

Let us show how the hyperparameters were learned in our experimental results. As was alluded to earlier, the hyperparameter learning is one of the main points of this study. Figure 10 demonstrates box plots of some of the commonality hyperparameter posteriors, associated with features 17, 4, 6, 36, 37, 27, 1, 2 and 3. Since the hyperparameter associated with feature 17 is relatively large, feature 17 could be less relevant to the prediction problem in question. Features 1, 2, and 3 could be more relevant

than the others because the associated commonality hyperparameters are smaller. Features 4, 6, 36, 37 and 27 are of intermediate relevance for the target prediction problem. Figure 10 shows posterior trajectories of the commonality hyperparameters associated with features 17, 36 and 2, where the horizontal axis is the MCMC iteration number. The vertical axis is in log scale.

The features which appeared in Fig. 9 are described by the following list, where "Player occlusion weight" is defined as the parameter which places weight proportional to the number of occluded players. "Grid-based feature" has been defined in Sect. 4.1. In addition to a target feature quantity, each feature carries two descriptors: whether it is a statistical or a grid-based quantity, and whether or not a player occlusion weight is applied. The following list includes such information together with the meaning of each feature.

a. feature 17

1. minimum velocity among all the players
2. statistical feature
3. no player occlusion weight
4. indicates if the players move

b. feature 4

1. variance of the X-coordinates among all the players
2. statistical feature
3. no player occlusion weight
4. indicates the player density

Fig. 8 Examples of event predictive probabilities estimated by the proposed method. A ground truth is superimposed as a *gray rectangle*. The horizontal axis is over a 500-s interval around typical event occurrences. The ranges of the vertical axes are adjusted to show the maximum and minimum values of the predictive probabilities. **a** CK, **b** FK, **c** GK
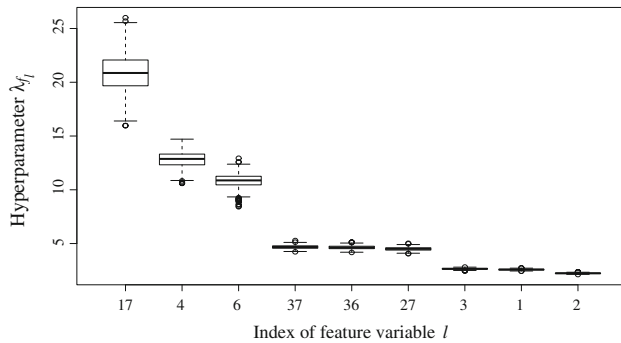
(a) Example of the predictive probability for (CK).

(b) Example of the predictive probability for (FK).

(c) Example of the predictive probability for (GK).
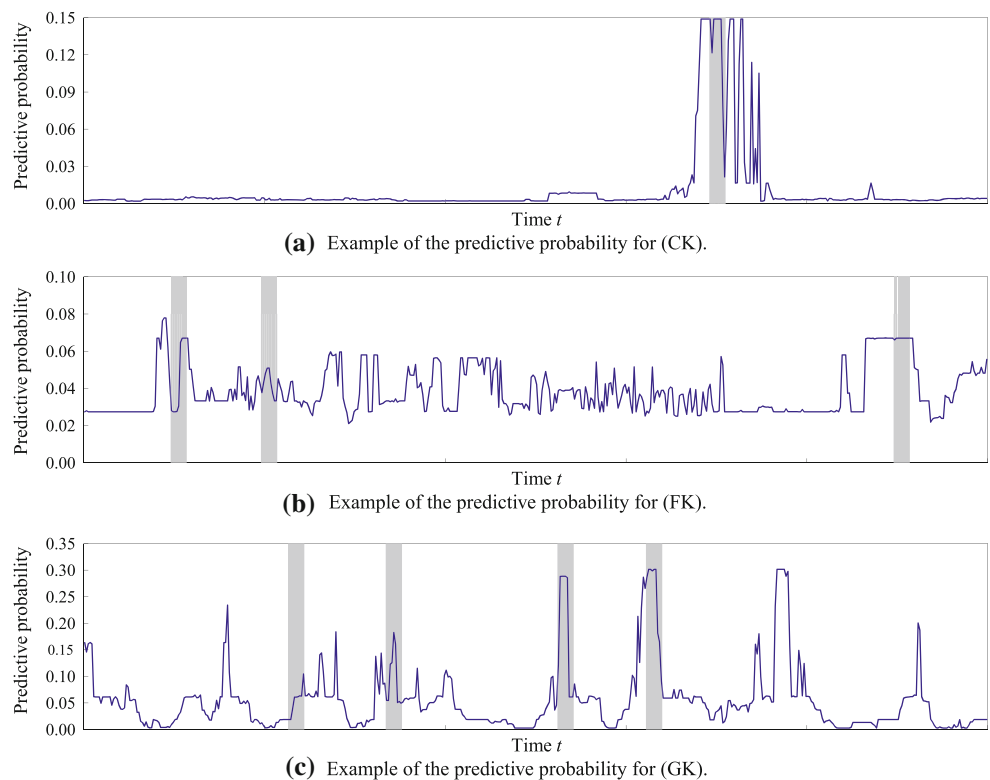
Fig. 9 Box plots of some of the commonality hyperparameter posteriors. The larger the posterior value of the hyperparameter, the less its relevance on the predictions. Feature 17 appears relatively irrelevant whereas features 3, 1, and 2 appear more relevant
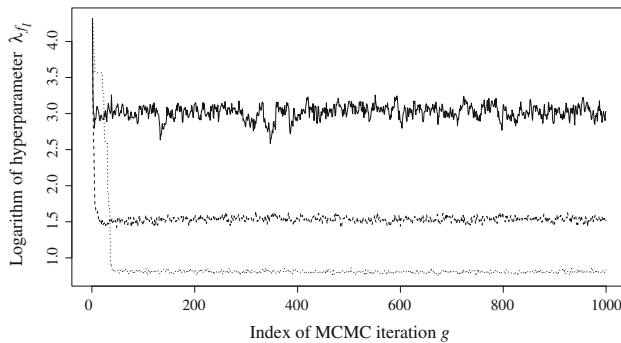

Fig. 10 Trajectories of the posterior samples of some of the commonality hyperparameters associated with features 17 (*top*), 36 (*middle*), and 2 (*bottom*), where the horizontal axis is the MCMC iteration number. The vertical axis is in log scale

c. feature 6

1. variance of the X-coordinates among all the players
2. statistical feature
3. with player occlusion weight
4. indicates the player density

d. feature 37

1. number of players in a grid
2. grid-based feature
3. no player occlusion weight
4. indicates the number of players in the center foreground area

e. feature 36

1. number of players in the center background area
2. grid-based feature
3. no player occlusion weight
4. indicates the number of players in the center background area

f. feature 27

1. sum of the players' velocities in the center field
2. grid-based feature

3. with player occlusion weight
4. indicates the total momentum of players in the center background area

g. feature 3

1. number of players in the center background area
2. grid-based feature
3. with player occlusion weight
4. indicates the number of players in the center background area

h. feature 1

1. number of players in the center field convolved with Gaussian kernels
2. grid-based feature
3. no player occlusion weight
4. indicates the density of players in the center field area

i. feature 2

1. mean of the X-coordinates among all the players
2. statistical feature
3. with player occlusion weight
4. indicates the players' average position

The number of players in the center field and the players' mean X-coordinate appears important, whereas the minimum velocity of the players appears less relevant for the event perdiction in question.

Average computational time for each time step is 5.019×10^{-3}s/sample with the following environment:

CPU: Intel(R) Xeon(TM) (3.00GHz), memory: 3.00 GB, OS: Microsoft Windows Server 2003 Enterprise Edition Service Pack 2, language: C++. Compiler: Microsoft(R) 32-bit C/C++ Optimizing Compiler Version.

With parallel processors each handling certain number of samples, real time computations are feasible.

The order of complexity for event detection is

$O(RN^2 + RNL)$, where $R = $#samples, $N = $#states,

$L = $#output values (dimension of feature variable).

4.4 Performance comparison

Of the many papers on sports event detection methods listed in the references, many of them use EM algorithms for parameter estimation, whereas references [16, 17] use MCMC. The advantages and disadvantages of MCMC and EM are relatively well understood. EM algorithms are simpler to implement, but they sometimes suffer from the local maxima problem, giving rise to sensitivity to the initial conditions. MCMC needs more work than EM to implement; however, it is relatively robust.

As was alluded already, there are two novelties, we believe, in the present study: (a) the hyperparameter

learning, and (b) the particular hyperparameter prior distributions. Recall that hyperparameters are one of the important ingredients of Bayesian learning in that they control the properties of the target prior distributions so that the shape of the target distribution at a certain set of hyperparameter values can be largely different from that with another set of hyperparameter values. Thus, if hyperparameters are learned automatically from the data, the prediction capability could be improved. This is what we attempt to demonstrate in our paper. Therefore, the point of performance comparison boils down to a comparison of cases with or without hyperparameter learning in HMM for soccer game event detection. Of the references [10–17, 19] on Bayesian HMM-based sports event detection methods, none appears to perform hyperparameter learning if we understood them correctly.

Here, we will describe performance comparisons of the HMM event detection method with and without hyperparameter learning. Both methods were trained and tested with the same feature variable sequences under the same settings. Our comparison was in terms of three theoretical measures. One is the paired Wilcoxon test on the differences of AOC values given in the second and third columns of Table 2. The result of this test is summarized by the following:

Paired Wilcoxon test

1. significance level: 0.05
2. one−sided alternative hypothesis: median of AOC differences >0
3. p−value: 0.03125.

The second and the third measures are information based. One is the cross entropy (negative averaged log likelihood):

$$H_{test}(q) := -1/T \sum_{t=1}^{T} \log_2 q(e_t^{new}),$$

whereas the other is the perplexity:

$$2^{H_{test}(q)}, \tag{29}$$

where $q(e_t^{new})$ stands for the approximated predictive probability. Table 3 shows these information-based

Table 3 Information-based error indexes with test sequence

Index	Method		Reduction (%)
	without hyperparameter learning	with hyperparameter learning	
Cross entropy	7.891×10^{-1}	7.866×10^{-1}	0.32
Perplexity	1.728×10^{-1}	1.725×10^{-1}	0.18

indexes for the predictive probabilities estimated by both methods.

These comparisons appear to indicate that the proposed method outperformed the method without hyperparameter learning.

5 Conclusion

In this paper, we have proposed a method for soccer game event detection with an HMM in a hierarchical Bayesian framework. The method was hierarchical in the sense that hyperparameter learning was performed in addition to the target HMM parameter learning. Furthermore, we have proposed particular hyperparameter prior distributions which are in a product form consisting of commonality hyperparameters and scale hyperparameters. One of the main reasons for using such prior distributions is their ability to automatically control ineffective features. The method was implemented by MCMC instead of the popular EM.

The proposed method was applied to 40-dimensional data sequences extracted from real professional soccer games. The performance was compared with a method without hyperparameter learning. The comparison was performed with respect to three information criteria: AOC, cross entropy, and perplexity. The proposed method appeared functional.

The following is a list of possible future research projects that are planned in our laboratory:

- Model extensions: In many cases of sequential data modeling, model extensions with generalized HMMs, also known as hidden semi-Markov Models, can improve the modeling performance, and they have been successfully applied to several problems (e.g., [27, 28]). Model extensions, including such a generalized HMM-based approach, are expected to be effective for event detection problems.
- Incomplete information: There are cases where some of the data is missing. Such a situation is called incomplete information" and is common in Bayesian HMM learning framework and is solvable. We would like to deal with this as the subject of a future research project.
- Problems with wider scope: In this paper, we focused on the event detection problem based on a small number of professional soccer game data. Currently we are in a process of obtaining a larger data set of soccer game videos of a professional league. Applications of the proposed method, with modifications, to problems other than event detection problems, such as sports strategy/situation analysis, may be interesting.
- Other sports: Although we applied the proposed method only to soccer games in this paper, the proposed

method is not limited to soccer games. Applications to other sports can be considered, e.g., rugby football, ice hockey, and basketball, among others.

Appendix A: Designed proposal distributions of hyperparamters

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 $Q(\cdot)$.[6] However, it is difficult to use the full conditional distributions of λ_{f_l} and η_{f_l} 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.[7]

Proposal distribution of λ_{f_l}

The full conditional distribution of λ_{f_l} is

$$P(\lambda_{f_l}|Y, Z, \theta, \{\eta_{f_{l'}}\}, \{\lambda_{f_{l'}}\}_{l' \neq l}) = P(\lambda_{f_l}|b_{f_l}, \eta_{f_l}). \quad (30)$$

Applying the log-normal distribution $\mathcal{LN}(\cdot)$, the full conditional distribution (30) can be approximated by

$$P(\lambda_{f_l}|b_k, \eta_{f_l}) \approx \mathcal{LN}(\lambda_{f_l}; \mu(\lambda_{f_l}^{(g-1)}), v(\lambda_{f_l}^{(g-1)})), \quad (31)$$

where the logarithmic mean $\mu(\cdot)$ is approximated by $\mu(\lambda_{f_l}) = \log \lambda_{f_l} + \nabla\nabla L_{f_l}(\lambda_{f_l})^{-1}\nabla L_{f_l}(\lambda_{f_l})$, and the logarithmic variance $v(\cdot)$ can be calculated as $v(\lambda_{f_l}) = \nabla\nabla L_{f_l}(\lambda)^{-1}$. The function $L_{f_l}(\lambda_{f_l})$ stands for $L_{f_l}(\lambda_{f_l}) := \log P(\log \lambda_{f_l}|b_k, \eta_{f_l}) =$

$\log P(\lambda_{f_l}|b_k, \eta_{f_l}) + \log \lambda_{f_l}$, the function $\nabla L_{f_l}(\cdot)$ is the first-order derivative of $L_{f_l}(\cdot)$, and the function $\nabla\nabla L_{f_l}(\cdot)$ stands for its second-order derivative.

However, the approximation (31) is not valid when the second-order derivative $\nabla\nabla L_{f_l}(\cdot)$ is small and/or $\log \lambda_{f_l}^{(g-1)}$ is far from the peak of $L(\lambda_{f_l})$. Therefore, here we consider (1) a simple modification for small values of $\nabla\nabla L_{f_l}(\cdot)$, and (2) a combination with a simple proposal distribution based on the logarithmic random walk. More specifically, the distribution described by

$$Q(\lambda_{f_l}; \cdot) := w_\lambda Q_{\text{app}}(\lambda_{f_l}; \cdot) + (1 - w_\lambda)Q_{\text{sim}}(\lambda_{f_l}; \cdot), \quad (32)$$

is used for the proposal distribution of λ_{f_l}. Here, the variable w_λ stands for the mixing rate of the two proposal distributions. The first proposal distribution $Q_{app}(\cdot)$ is defined using the approximation (31) with the modification for small $\nabla\nabla L_{f_l}(\cdot)$ as follows:

$$Q_{\text{app}}(\lambda_{f_l}; \cdot) := \mathcal{LN}(\lambda_{f_l}; \mu^*(\lambda_{f_l}^{(g-1)}), v^*(\lambda_{f_l}^{(g-1)})), \quad (33)$$

where the logarithmic mean and the logarithmic variance are modified by $\mu^*(\lambda_{f_l}) = \log \lambda_{f_l} + \nabla\nabla L_{f_l}^*(\lambda_{f_l})^{-1}\nabla L_{f_l}(\lambda_{f_l})$ and $v^*(\lambda_{f_l}) = \nabla\nabla L_{f_l}^*(\lambda_{f_l})^{-1}$ with the modification of $\nabla\nabla L_{f_l}(\cdot)$ as $\nabla\nabla L_{f_l}^*(\cdot) = \min(\nabla\nabla L_{f_l}(\cdot), s)$. The positive variable s controls the minimum value of $\nabla\nabla L_{f_l}^*(\cdot)$. On the other hand, the second proposal distribution $Q_{sim}(\cdot)$ in (32) is based on the logarithmic random walk, as follows:

$$Q_{\text{sim}}(\lambda_{f_l}; \cdot) := \mathcal{LN}(\lambda_{f_l}; \log \lambda_{f_l}^{(g-1)}, v_{\text{sim}}), \quad (34)$$

where v_{sim} stands for the logarithmic variance of this proposal distribution.

In the experiments described in this paper, we used the proposal distribution (32) with $w_\lambda = 0.5$, $s = 1.0$, and $v_{sim} = 0.01$. These gave reasonable and stable performance in our preliminary experiments.

Proposal distribution of η_k

The full conditional distribution of η_{f_l} can be written as

$$P(\eta_k|Y, Z, \theta, \{\eta_{k'}\}_{k' \neq k}, \{\lambda_{k'}\}) = P(\eta_k|b_{f_l}, \lambda_{f_l}). \quad (35)$$

It is difficult to approximate the distribution (35) itself with basic methods. Therefore, we consider only a rough approximation of the center of the distribution (35) 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 $\bar{\eta}_k(b_k) = \frac{1}{N}\sum_{i=1}^{N} b_{k,i}$. We assume that the center of the distribution (35) can be roughly approximated by this estimator $\bar{\eta}_k(b_k)$. In view of this assumption and the simplicity of the implementation, we consider the Dirichlet proposal distribution centered on $\bar{\eta}_k(b_{f_l}^{(g)})$:

Table 4 Summary of candidate components

Category	Sub-category	Description
Statistics-based variables	All players' information	Statistics of all players' positions
	Left team information	Statistics of left team players' positions
	Right team information	Statistics of right team players' positions
	Goalkeeper information	Positions of goalkeepers
Grid-based variables	Number information	The number of players in specific zone
	Velocity information	Velocity information of players in specific zone

$$Q_{\text{app}}(\eta_{f_i}; \cdot) := \mathcal{D}ir(\eta_{f_i}; v_{\text{app}} \bar{\eta}_k(b_{f_i}^{(g)})), \qquad (36)$$

where the user-settable positive variable v_{app} controls the concentrate level of the distribution.

Since this approximation is often not valid, the combination with a simple proposal distribution is also considered here. The actual proposal distribution for η_{f_i} in our implementation is described as

$$Q(\eta_{f_i}; \cdot) := w_\eta Q_{\text{app}}(\eta_{f_i}; \cdot) + (1 - w_\eta) Q_{\text{sim}}(\eta_{f_i}; \cdot), \qquad (37)$$

where w_η is the mixing rate of the two proposal distributions. The first proposal distribution $Q_{app}(\cdot)$ is described as (36). The second one, $Q_{sim}(\cdot)$, is defined using the previous sample $\eta_{f_i}^{(g-1)}$, as follows:

$$Q_{\text{sim}}(\eta_{f_i}; \cdot) := \mathcal{D}ir(\eta_{f_i}; v_{\text{sim}} \eta_{f_i}^{(g-1)}), \qquad (38)$$

where the positive variable v_{sim} controls the concentrate level of this distribution.

In the actual implementation for the experiment described in this paper, we set $w_\eta = 0.5$ and $v_{app} = v_{sim} = 100$, which showed reasonable performance in our preliminary experiments.

Appendix B: Candidate components and screening method

This part describes the screening method and candidate components used in Sect. 4.

Candidate components

As described in Sect. 4, we considered 5,063 candidates for feature variables. These candidates are categorized into two groups: statistics-based variables and grid-based variables. Statistics-based variables are statistical information

---

Screening procedure

(I) Information gain evaluation step:
  (a) For each candidate $f_{c,i}$ and event e, evaluate the information gain (39).
  (b) For each candidate $f_{c,i}$, calculate the sum of the information gain: $SG(i) := \sum_e IG(i; e)$.
  (c) Sort candidates $f_{c,i}$ in descending order according to the sum of information gain $SG(i)$.

(II) Candidate addition step:
  Start the following steps with the selected candidate set $\mathcal{S}$ empty and $k = 1$.
  (a) Select the candidate $f_{c,k\text{-th}}$, which has the k-th largest $SG(\cdot)$.
  (b) Calculate the mutual information between the candidate $f_{c,k\text{-th}}$ and each component in the set $\mathcal{S}$.
  (c) Add the candidate $f_{c,k\text{-th}}$ to the set $\mathcal{S}$, if all mutual information items $MI(\cdot)$ evaluated in step (b) are less than the threshold value MI_{thred}.
  (d) If the size of the set $\mathcal{S}$ equals the target number $N_{\mathcal{S}}$, stop the procedure. Otherwise, go back to step (a) after updating $k \leftarrow k + 1$.

Fig. 11 Screening procedure for candidates

of players' positions, e.g., average, median, maximum, minimum, and variance of players' positions. On the other hand, grid-based variables are based on information of a specific zone, which is derived from dividing the soccer field. For instance, a grid-based variable is the number of players in a specific zone. Table 4 shows a summary of all candidate components. Although most of these components are continuous variables, all of the components are quantized into ten levels.

Screening method

Let us consider two information-based criteria:

1. Information gain given specific event: To consider the dependency of the specific candidate on a specific event, we use information gain given the specific event e, described as

$$\text{IG}(i; e) := \sum_{f_{c,i}} \rho(f_{c,i}; e) \log_2 \frac{\rho(f_{c,i}; e)}{\rho(f_{c,i})}. \qquad (39)$$

Here, the variable $f_{c,i}$ denotes the ith candidate, and the function $\rho(f_{c,i})$ stands for the empirical probability distribution of $f_{c,i}$, i.e., the relative frequency of $f_{c,i}$ in the training data set. The function $\rho(f_{c,i}; e)$ represents the empirical conditional probability distribution of $f_{c,i}$ given specific event e, which is estimated using the training data set.

2. Mutual information between two candidates: For evaluation of the independency between two candidates, we consider mutual information between the two candidates, as follows:

$$\mathrm{MI}(i,j) := \sum_{f_{c,i}} \sum_{f_{c,j}} \rho(f_{c,i}, f_{c,j}) \log_2 \frac{\rho(f_{c,i}, f_{c,j})}{\rho(f_{c,i}) \rho(f_{c,j})}. \qquad (40)$$

Here, the function $\rho(f_{c,i}, f_{c,j})$ denotes the empirical joint probability distribution of $f_{c,i}$ and $f_{c,j}$, which is estimated from the training data set.

Using these two cirteria, the screening method selects feature variables from the candidates. The concrete procedure can be summarized as shown in Fig. 11. In the experiment described in Sect. 4, the threshold value I_{thred} is set at $\mathrm{MI}_{\text{thred}} = 0.75$, and the number of selected candidates $N_{\mathcal{S}}$ is $N_{\mathcal{S}} = 40$.

References

1. Jain R, Hampapur A (1994) Metadata in video databases. ACM SIGMOD Rec 23(4):27–33
2. Messina A, Boch L, Dimino G, Bailer W, Schallauer P, Allasia W, Groppo M, Vigilante M, Basili R (2006) Creating rich metadata in the TV broadcast archives environment: the Presto-Space project. In: Proceedings of 2nd international conference on automated production of cross media content for multi-channel distribution (AXMEDIS 2006), pp 193–200.
3. Lugmayr A, Niiranen S, Kalli S (2004) Digital interactive TV and metadata: future broadcast multimedia. Springer, Berlin
4. Sano M, Sumiyoshi H, Shhibata M, Yagi N (2005) Generating metadata from acoustic and speech data in live broadcasting. In: Proceedings of 2005 IEEE international conference on acoustics, speech, and signal processing (ICASSP 2005), vol 2, pp 1145–1148
5. Miyamori H, Nakamura S, Tanaka K (2005) Automatic indexing of broadcast content using its live chat on the Web. In: Proceedings of 2005 IEEE international conference on image processing (ICIP 2005), pp 1248–1251
6. Misu T, Takahashi M, Tadenuma M, Yagi N (2005) Real-time event detection based on formation analysis of soccer scenes. Inf Tech Lett 4(LI-003):141–144
7. Motoi S, Misu T, Nakada Y, Matsumoto T, Yagi N (2007) Bayesian hidden Markov model approach for events detection in sports movie. IPSJ SIG Notes 2007(1):133–139
8. Motoi S, Nakada Y, Misu T, Matsumoto T, Yagi N (2008) A novel hierarchical Bayesian HMM for multi-dimensional discrete data. In: Proceedings of 8th IASTED conference on artificial intelligence and applications (AIA 2008), pp 52–57
9. Motoi S, Nakada Y, Misu T, Matsumoto T, Yagi N (2008) A hierarchical Bayesian hidden Markov model for multi-dimensional discrete data. In: Frontiers in robotics, automation and control. In-Tech Publications, pp 357–374
10. Xu G, Ma Y-F, Zhang H-J, Yang S (2003) A HMM based semantic analysis framework for sports game event detection. In: International conference on image processing (ICIP 2003),vol 1, pp 25–28
11. Sadlier DA, O Connor NE (2005) Event detection in field sports video using audio visual features and a support vector machine. IEEE Trans Circuits Syst Video Technol 15(10):1225–1233
12. Kang Y-L, Lim J-H, Kankanhalli MS, Xu C-S, Tian Q(2004) Goal detection in soccer video using audio/visual keywords. In: International conference on image processing (ICIP2004), pp 1629–1632
13. Bashir FI, Khokhar AA, Schonfeld D (2007) Object trajectory-based activity classification and recognition using hidden markov models. IEEE Trans Image Process 16(7):1912–1919
14. Assfalg J, Bertini M, Del Bimbo A, Nunziati W, Pala P (2002) Soccer highlights detection and recognition using HMMs. In: ICME 2002, vol 1, pp 825–828
15. Wang J, Xu C, Chng E, Tian Q (2004) Sports highlight detection from keyword sequences using HMM. In: ICME 2004, vol 1, pp 599–602
16. Xie L, Xu P, Chang S-F, Divakaran A, Sun H (2004) Structure analysis of soccer video with domain knowledge and hidden Markov models. Pattern Recogn Lett 25:767–775
17. Xie L, Chang S-F, Divakaran A, Sun H (2003) Unsupervised discovery of multilevel statistical video structures using hierarchical hidden Markov models. In: ICME 2003, vol 3, pp 29–32
18. Baum LE, Petrie T, Soules G, Weiss N (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann Math Stat 41(1):164–171
19. Li B, Sezan MI (2001) Event detection and summarization in sport video. In: Proceedings of IEEE workshop on content-based access of image and video libraries (CBAIVL 2001), pp 132–138.
20. Huo Q, Chan C, Lee CH (1995) Bayesian adaptive learning of the parameters of hidden Markov model for speech recognition. IEEE Trans Speech Audio Process 3(5):334–345
21. MacKay DJC (1997) Ensemble learning for hidden Markov models. Cavendish laboratory, University of Cambridge. http://wol.ra.phy.cam.ac.uk/mackay/.
22. Scott SL (2002) Bayesian methods for hidden Markov models: recursive computing in the 21st century. J Am Stat Assoc 97(457):337–351
23. Rezek I, Roberts S (2005) Ensemble hidden Markov models with extended observation densities for biosignal analysis. In: Probabilistic modeling in bioinformatics and medical informatics, pp 419–450
24. Geman S, Geman D (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans Pattern Anal Mach Intell 6(6):721–741
25. Hastings W (1970) Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57(1):97–109
26. Misu T, Naemura M, Zheng W, Izumi Y, Fukui K (2002) Robust tracking of soccer players based on data fusion. In: Proceedings of 16th international conference on pattern recognition (ICPR 2002), vol 1, pp 556–561
27. Juang BH, Rabiner LR (1991) Hidden Markov models for speech recognition. Technometrics 33(3):251–272
28. Russell M, Moore R (1985) Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition. In: Proceedings of 1985 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1985), vol 10, pp 5–8