Sequence Pattern Extraction by Segmenting Time Series Data Using GP-HSMM with Hierarchical Dirichlet Process

Masatoshi Nagano[1,*], Tomoaki Nakamura[1], Takayuki Nagai[1], Daichi Mochihashi[2], Ichiro Kobayashi[3] and Masahide Kaneko[1]

***Abstract*— Humans recognize perceived continuous information by dividing it into significant segments such as words and unit motions. We believe that such unsupervised segmentation is also an important ability that robots need to learn topics such as language and motions. Hence, in this paper, we propose a method for dividing continuous time-series data into segments in an unsupervised manner. To this end, we proposed a method based on a hidden semi-Markov model with Gaussian process (GP-HSMM). If Gaussian processes, which are nonparametric models, are used, unit motion patterns can be extracted from complicated continuous motion. However, this approach requires the number of classes of segments in the time-series data in advance. To overcome this problem, in this paper, we extend GP-HSMM to a nonparametric Bayesian model by introducing a hierarchical Dirichlet process (HDP) and propose the hierarchical Dirichlet processes-Gaussian process-hidden semi-Markov model (HDP-GP-HSMM). In the nonparametric Bayesian model, an infinite number of classes is assumed and it becomes difficult to estimate the parameters naively. Instead, the parameters of the proposed HDP-GP-HSMM are estimated by applying slice sampling. In the experiments, we use various synthetic and motion-capture data to show that our proposed model can estimate a more correct number of classes and achieve more accurate segmentation than baseline methods.**

I. Introduction

Humans perceive continuous information by dividing it into significant segments. For example, we can recognize words by segmenting a speech wave and perceive unit motions by segmenting continuous motion. Humans can learn words and unit motions by appropriately segmenting the continuous information without explicit segmentation points, even though there are an infinite number of possibilities for the segmentation points. We consider that such an ability is also important for robots so that they can learn motions and language flexibly.

To this end, we previously proposed a Gaussian process-based hidden semi-Markov model (GP-HSMM). This model is a hidden semi-Markov model (HSMM) with a Gaussian process emission distribution, which makes it possible to segment the time-series data in an unsupervised manner [1]. Fig. 1 depicts the generative process of the GP-HSMM. Each state of the HSMM represents a class of a segment, and the segments are generated from the Gaussian process corresponding to their class. Similar motion patterns are generated from the same Gaussian process, and the observed continuous sequence is generated by combining them. By the estimating the parameters of the GP-HSMM, the time-series data can be segmented.

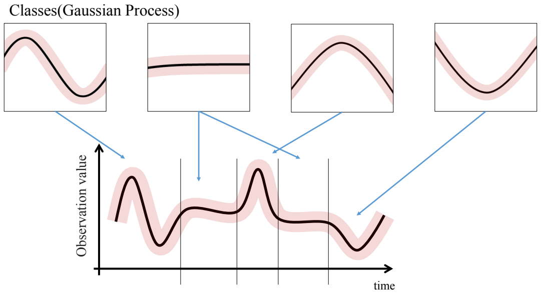


Fig. 1. Overview of the generative process of the proposed method.

However, a GP-HSMM requires the number of classes into which the continuous sequence is divided. This is a big drawback because the number of classes is usually not known in advance. In this paper, to overcome this problem, we extend GP-HSMM to the nonparametric Bayesian model HDP-GP-HSMM by introducing a hierarchical Dirichlet process (HDP) [2], which makes it possible to estimate the number of classes. The nonparametric Bayesian model assumes an infinite number of classes; however, a finite number of these classes are actually used for representing the training data. As a result, the number of classes can be estimated. By estimating the parameters of the HDP-GP-HSMM, the segments, their class, and the number of classes can be estimated at the same time. However, forward filtering–backward sampling, which is the parameter estimation method of GP-HSMM, cannot be naively applied to HDP-GP-HSMM because this model assumes an infinite number of classes. To apply it to HDP-GP-HSMM parameter estimation, we utilize the slice sampler proposed in [3] to limit the potential number of classes. We evaluate the efficiency of the proposed HDP-GP-HSMM using synthetic data and real motion-capture data. The experimental results show that HDP-GP-HSMM achieves a higher accuracy than some baseline methods.

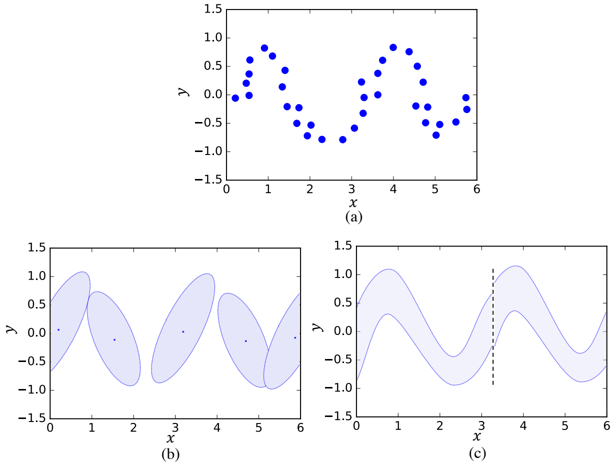
Many studies for unsupervised motion segmentation have been conducted. However, heuristic assumptions are used in many of them [4], [5], [6]. Wächter et al. proposed a method for the segmentation of object manipulation motions by robots and used contact between the end-effector and the object as a segmentation clue. The method proposed by Lioutikov et al. requires candidates for the segmentation points

[1]Department of Mechanical Engineering and Intelligent Systems, The University of Electro-Communications, Chofu-shi, Japan

[2]Department of Mathematical Analysis and Statistical Inference, Institute of Statistical Mathematics, Tachikawa, Japan

[3]Department of Information Sciences, Faculty of Sciences, Ochanomizu University, Bunkyo-ku, Japan

Correspondence*: n1832072@edu.cc.uec.ac.jp



(a)

(b)

(c)

Fig. 2. Illustration of representation of trajectory: (a) Observed data points, (b) representation by HMM and (c) representation by GP-HSMM.

in advance. In addition, the method proposed by Takano et al. utilizes the errors between the predicted and actual values as criteria for segmentation. Moreover, some methods use motion features such as the zero velocity of joint angles [7], [8], [9]. However, it is not clear that these heuristics are valid for all motions. In contrast, we do not use such heuristics and formulate the problem of the segmentation using a stochastic model. In some studies, the segmentation is formulated stochastically, and hidden Markov models (HMMs) are used [10], [11], [12], [13]. However, it is difficult for the HMMs to represent complicated motion patterns, and we instead use Gaussian processes, which are a type of non-parametric model that can represent more complicated time-series data than HMMs.

Regarding to the HMMs, HMM using HDP is also proposed [10], [14]. Beal et al. proposed HDP-HMM that has infinite number of states [10]. Moreover, Fox et al. extended it to sticky HDP-HMM [14], which can prevent over-segment by increasing self-transition probability. Fig. 2(b) depicts the representation of trajectory of data points (Fig. 2(a)) by HMM. HMM represents trajectory using five Gaussian distributions and the number of Gaussian distributions can be estimated by using HDP. However, one can see that details of the trajectory are lost. On the other hand, in GP-HSMM, trajectory can be represented continuously by using two GPs and, moreover, HDP makes it possible to estimate the number of GPs. Thus, we consider HDP-GP-HSMM can represents complex trajectories more accurately.

II. Nonparametric Bayesian model

In this study, the number of motion classes are estimated by utilizing a non-parametric Bayesian model. In the parametric Bayesian model case, the number of classes is finite, the parameters of a multinomial distribution are determined by a finite-dimensional Dirichlet distribution, and, finally, the classes are sampled from this multinomial distribution. In contrast, in the non-parametric Bayesian model case, an infinite number of classes are assumed, and the classes are sampled from the infinite-dimensional multinomial distribution. Only a finite number of classes are actually used, and, therefore, their number can be estimated. In the non-parametric Bayesian model, an infinite-dimensional multinomial distribution must be constructed, and one of the methods for it is the stick breaking process (SBP).

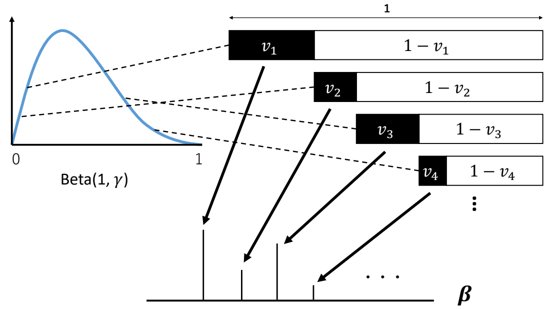


Fig. 3. Stick Breaking Process (SBP).

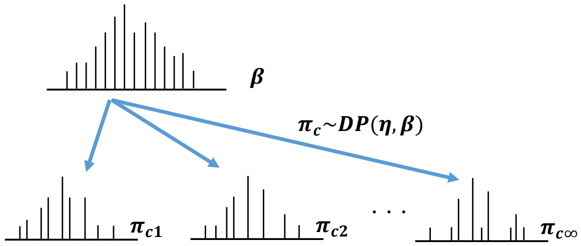

$\boldsymbol{\beta}$
$\boldsymbol{\pi}_c \sim DP(\boldsymbol{\eta}, \boldsymbol{\beta})$
$\boldsymbol{\pi}_{c1}$
$\boldsymbol{\pi}_{c2}$
$\boldsymbol{\pi}_{c\infty}$

Fig. 4. Hierarchical Dirichlet Process (HDP).

A. Stick Breaking Process (SBP)

The SBP [3] is a Dirichlet process that can construct an infinite-dimensional multinomial distribution. In the SBP, the parameters of infinite-dimensional multinomial distribution $\boldsymbol{\beta}$ are constructed by repeatedly breaking a stick, the length of which is one, with ratio v_k sampled from a beta distribution, as follows (Fig. 3):

$$v_k \sim \mathrm{Beta}(1,\gamma) \qquad (k = 1, \cdots, \infty), \tag{1}$$

$$\beta_k = v_k \prod_{i=1}^{k-1}(1 - v_i) \qquad (k = 1, \cdots, \infty). \tag{2}$$

This process is represented as $\boldsymbol{\beta} \sim \mathrm{GEM}(\gamma)$, where GEM stands for Griffiths, Engen and McCloskey [15].

B. Hierarchical Dirichlet Process (HDP)

Using a Dirichlet process, HMMs with countably many states can be constructed. However, if the SBP is naively used, the destinations reachable from each state have different states. In the case of models like HMMs, all state have to share the destination states and have different probabilities of transitioning to each state. To construct such a distribution, the distribution $\boldsymbol{\beta}$ that was generated by the SBP is shared with all states as a base measure, and transition probabilities $\boldsymbol{\pi}_c$, which are different in each state c, are generated by another Dirichlet process.

$$\boldsymbol{\pi}_c \sim \mathrm{DP}(\eta, \boldsymbol{\beta}). \tag{3}$$

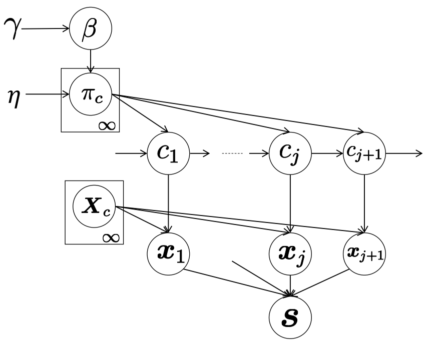


Fig. 5. Graphical model of HDP-GP-HSMM.

The method in which the probability distribution is constructed by a two-phase Dirichlet process is called an HDP (Fig. 4).

III. HDP-GP-HSMM

Fig. 5 shows a graphical model of the proposed method, which is a generative model of the time-series data. Here, $c_j (j = 1, 2, \cdots, \infty)$ denotes the classes of the segments, and we assumed that the number of classes is countably infinite. Probability $\boldsymbol{\pi}_c$ denotes the transition probability, which is generated from the $\boldsymbol{\beta}$ generated by the GEM distribution (SBP) parameterized by γ and the Dirichlet process parameterized by η.

$$\boldsymbol{\beta} \sim \mathrm{GEM}(\gamma), \quad (4)$$

$$\boldsymbol{\pi}_c \sim \mathrm{DP}(\eta, \boldsymbol{\beta}). \quad (5)$$

Class c_j of the j-th segment is determined by the class of the $(j-1)$-th segment and transition probability $\boldsymbol{\pi}_c$. The segment is generated by a Gaussian process with parameter $\boldsymbol{X}_c$ as

$$c_j \sim P(c | c_{j-1}, \boldsymbol{\pi}_c, \alpha), \quad (6)$$

$$\boldsymbol{x}_j \sim \mathcal{GP}(\boldsymbol{x} | \boldsymbol{X}_{c_j}), \quad (7)$$

where $\boldsymbol{X}_c$ denotes a set of segments that are classified into class c. Observed time series data is assumed to be generated by connecting segments sampled by the above generative process.

A. Gaussian process

In this paper, each class represents a continuous trajectory found by learning the emission x_i of time step i using a Gaussian process. In the Gaussian process, given the pairs $(\boldsymbol{i}, \boldsymbol{X}_c)$ of time step i and its emission, which are classified into class c, the predictive distribution of x^{new} of time step i^{new} becomes the Gaussian distribution

$$p(x^{new} | i^{new}, \boldsymbol{X}_c, \boldsymbol{i}) \propto \mathcal{N}(\boldsymbol{k}^T \boldsymbol{C}^{-1} \boldsymbol{i}, c - \boldsymbol{k}^T \boldsymbol{C}^{-1} \boldsymbol{k}), \quad (8)$$

where $k(\cdot, \cdot)$ denotes the kernel function and $\boldsymbol{C}$ is a matrix whose elements are

$$C(i_p, i_q) = k(i_p, i_q) + \omega^{-1} \delta_{pq}. \quad (9)$$

Here, ω denotes a hyper parameter that represents noise in the observation. Additionally, $\boldsymbol{k}$ is a vector whose elements are $k(i_p, i^{new})$, and c is $k(i^{new}, i^{new})$. A Gaussian process can represent complicated time-series data owing to the kernel function. In this paper, we used the following kernel function, which is generally used for Gaussian processes:

$$k(i_p, i_q) = \theta_0 \exp(-\frac{1}{2}\theta_1 ||i_p - i_q||^2 + \theta_2 + \theta_3 i_p i_q), \quad (10)$$

where θ_* denotes the parameters of the kernel.

Moreover, if the observations are composed of multidimensional vectors, we assume that each dimension is independently generated and, further, the predictive distribution is such that the emission $\boldsymbol{x} = (x_0, x_1, \cdots)$ of time step i is generated by a Gaussian process of class c, which is computed as follows:

$$\begin{aligned} \mathcal{GP}(\boldsymbol{x} | \boldsymbol{X}_c) &= p(x_0 | i, \boldsymbol{X}_{c,0}, \boldsymbol{I}) \\ &\quad \times p(x_1 | i, \boldsymbol{X}_{c,1}, \boldsymbol{I}) \\ &\quad \times p(x_2 | i, \boldsymbol{X}_{c,2}, \boldsymbol{I}) \cdots . \quad (11) \end{aligned}$$

Using this probability, the time-series data can be classified into the classes.

Algorithm 1 Blocked Gibbs Sampler

```
Repeat until convergence
for $n = 1$ to $N$ do
    for $j = 1$ to $J_n$ do
        $N_{c_{nj}} -= 1$
        $N_{c_{nj}, c_{n,j+1}} -= 1$
        Delete segments $\boldsymbol{x}_{nj}$ from $\boldsymbol{X}_{c_{nj}}$
    end for

    // Sampling segments and their classes
    $\boldsymbol{x}_{n*}, c_{n*} \sim P(\boldsymbol{x} | \boldsymbol{s}_n)$

    for $j = 1$ to $J_n$ do
        $N_{c_{nj}} ++$
        $N_{c_{nj}, c_{n,j+1}} ++$
        Append segments $\boldsymbol{x}_{nj}$ to $\boldsymbol{X}_{c_{nj}}$
    end for
end for
```

B. Parameter Inference

1) Blocked Gibbs Sampler: The parameters of the proposed model can be estimated by sampling segments and their classes. For efficient estimation, we utilize a blocked Gibbs sampler [16] in which all segments and their classes in the observed sequence are sampled. First, all observed sequences are randomly divided into segments and classified into classes. Next, segments $\boldsymbol{x}_{nj} (j = 1, 2, \cdots, J_n)$ obtained by segmenting $\boldsymbol{s}_n$ are excluded from the training data, and the parameters of Gaussian processes $\boldsymbol{X}_c$ and transition probabilities $P(c|c')$ are updated. The segments and their classes are sampled as follows:

$$(\boldsymbol{x}_{n,1}, \cdots, \boldsymbol{x}_{n,J_n}), (c_{n,1}, \cdots, c_{n,J_n}) \sim P(\boldsymbol{X}, \boldsymbol{c} | \boldsymbol{s}_n). \quad (12)$$

Using the sampled segments and their classes, the parameters of Gaussian processes $\boldsymbol{X}_c$ and transition probabilities

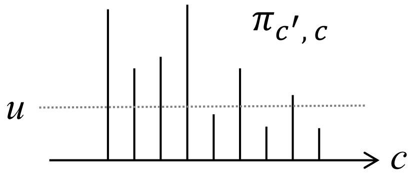
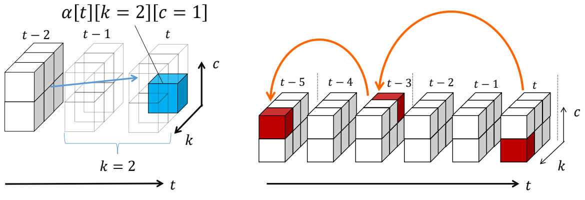


Fig. 6. Slice sampling truncates the number of classes by thresholding $\pi_{c',c}$.



(a) Example of Forward filtering

(a) Example of Backward sampling

Fig. 7. Forward filtering–backward sampling.

$P(c|c')$ are updated. Iterating this procedure, the parameters can be optimized. Algorithm 1 shows the pseudo code of the blocked Gibbs sampler. In this algorithm, $N_{c_{nj}}, N_{c_{nj},c_{n,j+1}}$ is a parameter to compute the transition probability in Eq. (15). However, we must also compute the probability in Eq. (12) for all possible patterns of segments and classes, which is impractical because of the infinite number of classes. To compute Eq. (12), we utilize forward filtering–backward sampling and a slice sampler, which are explained in the next section.

2) Slice Sampler: We assumed that the number of classes is countably infinite. However, it is difficult to compute Eq. (12) because c can be infinite possibilities. To overcome this problem, we utilize slice sampler that limits the number of classes stochastically proposed in [3]. In the slice sampler, auxiliary variable u_j for each time step j is introduced, and by thresholding transition probability $\boldsymbol{\pi}_c$ by using u_j, the number of classes become finite. The parameter u_j is sampled from following probability:

$$p(u_j|c_{j-1},c_j) = \frac{\xi(0<u_j<\pi_{c_{j-1},c_j})}{\pi_{c_{j-1},c_j}}, \tag{13}$$

where $\xi(A)=1$ if condition A is true; otherwise, it is 0. Fig. 6 illustrates the slice sampler. By truncating the classes with a transition probability π_{c_{j-1},c_j} that is less than u_j, the number of classes becomes finite.

3) Forward Filtering–Backward Sampling: The number of classes can be truncated by slice sampling. As a result, forward filtering–backward sampling [17] can be applied to compute Eq. (12). Forward filtering-backward sampling can be considered as MCMC version of the forward-backward algorithm which is a maximum likelihood estimation. Although the forward probabilities can be computed in the same manner as forward-backward algorithm, the latent variables are sampled backward based on forward probabilities in the backward sampling. In the forward filtering, the probability that k samples $\boldsymbol{s}_{t-k:k}$ before time step t form a segment of class c is as follows:

Algorithm 2 Forward Filtering–Backward Sampling

```
// Slice sampling
for  $j = 1$ to $J_n$ do
    $u_j \sim P(u_j|c_{j-1},c_j)$
end for
$\bar{C} = \max_j(\mathrm{count}(\pi_{c_{j-1},c_j} > u_j))$

// Forward filtering
for  $t = 1$ to $T$ do
    for  $k = 1$ to $K$ do
        for  $c = 1$ to $\bar{C}$ do
            Compute $\alpha[t][k][c]$
        end for
    end for
end for

// Backward sampling
$t = T, j = 1$
while  $t > 0$  do
    $k, c \sim p(\boldsymbol{x}_j|\boldsymbol{s}_{t-k:t})\alpha[t][k][c]$
    $\boldsymbol{x}_j = \boldsymbol{s}_{t-k:t}$
    $c_j = c$
    $t = t - k$
    $j = j + 1$
end while
return $(\boldsymbol{x}_{J_n}, \boldsymbol{x}_{J_n-1}, \cdots, \boldsymbol{x}_1), (c_{J_n}, c_{J_n-1}, \cdots, c_1)$
```

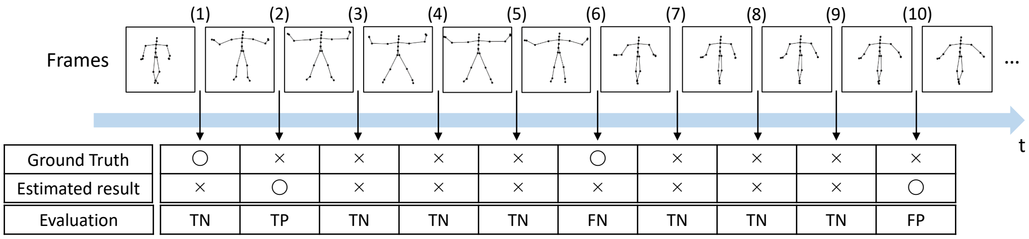
$$\begin{aligned}\alpha[t][k][c] &= \mathcal{GP}(\boldsymbol{s}_{t-k:k}|\boldsymbol{X}_c) \\ &\times \sum_{k'=1}^{K}\left\{\sum_{c'=1}^{\bar{C}} p(c|c')\alpha[t-k][k'][c']\right\},\end{aligned} \tag{14}$$

where $\bar{C}$ denotes the maximum number of classes estimated by slice sampling and K denotes the maximum length of segments. In addition, $p(c|c',\boldsymbol{\beta},\alpha)$ is the transition probability, which can be computed as follows:

$$p(c|c',\boldsymbol{\beta},\alpha) = \frac{N_{c'c}+\alpha\beta_{c'}}{N_{c'}+\alpha}, \tag{15}$$

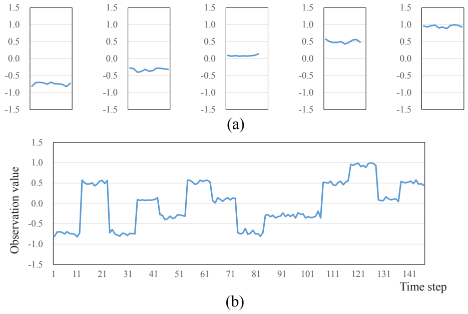
where $N_{c'}$ and $N_{c'c}$ represent the number of segments of class c and the number of transitions from c' to c, respectively. In addition, k' and c' are the length and class of possible segments before $\boldsymbol{s}_{t-k:k}$, and these probabilities are marginalized out. Moreover, $\alpha[t][k][*]=0$ if $t-k<0$, and $\alpha[0][0][*]=1.0$. Equation (14) can be recursively computed from $\alpha[1][1][*]$ using dynamic programming, as shown in Fig. 7(a). This figure depicts an example of computing $\alpha[t][2][1]$, which is the probability that the two samples before t become a segment whose class is c. In this case, the length is two and, therefore, all the segments with end points $t-2$ can potentially transit to this segment. In $\alpha[t][2][1]$, these possibilities are marginalized out.

Finally the length of segments and their classes can be sampled by backward sampling from $t=T$. Fig. 7(b) depicts an example of backward sampling. The length of segment k_1 and its class c_1 are sampled from the probabilities $\alpha[t][*][*]$. If $k_1=3$, k_2 and c_2 are sampled from $\alpha[t-3][*][*]$. By



Ground Truth	○	×	×	×	×	○	×	×	×	×
Estimated result	×	○	×	×	×	×	×	×	×	○
Evaluation	TN	TP	TN	TN	TN	FN	TN	TN	TN	FP

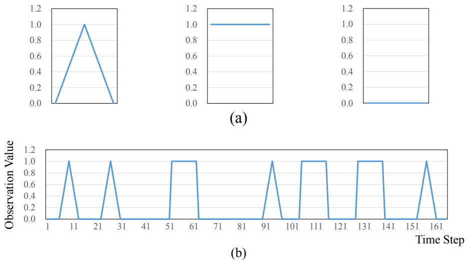
Fig. 8. Example of segmentation evaluation: TP is assigned to the boundary (2) because estimated boundary is within error range from true boundary, which is ±5% of the sequence length.



(a)

(b)

Fig. 9. Synthetic time series 1: (a) five unit sequences and (b) connected sequence.



(a)

(b)

Fig. 10. Synthetic time series 2: (a) three unit sequences and (b) connected sequence.

iterating this procedure until $t = 0$, the segments and their classes can be determined.

Algorithm 2 shows the pseudo code of forward filtering–backward sampling with slice sampling.

IV. Experiments

To validate the proposed HDP–GP-HSMM, we applied it to several types of time-series data. For comparison, we used HDP-HMM [10], HDP-HMM+NPYLM [11], BP-HMM[12], and Autoplait [13] as baseline methods.

A. Evaluation metrics

We used four measures to evaluate the accuracy of segmentation: normalized Hamming distance, precision, recall, and F-measure.

The normalized Hamming distance represents the error rate of the classification of the data points and is computed as follows:

$$ND(\boldsymbol{c}, \overline{\boldsymbol{c}}) = \frac{D(\boldsymbol{c}, \overline{\boldsymbol{c}})}{|\overline{\boldsymbol{c}}|}, \tag{16}$$

where $\boldsymbol{c}$ and $\overline{\boldsymbol{c}}$ respectively represent sequences of estimated classes and correct classes in the data points in the observed sequence. Moreover, $D(\boldsymbol{c}, \overline{\boldsymbol{c}})$ represents the Hamming distance between two sequences, and $|\overline{c}|$ is the length of the sequence. Therefore, the normalized Hamming distance ranges from zero to one, and smaller values indicate that the estimated classes are more similar to the ground truth. To compute precision, recall, and F-measure, we evaluated boundary points (boundaries between segments) as true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs), as shown in Fig. 8. TP is assigned to the points that are correctly estimated as boundary points. An estimated boundary point is treated as correct if the estimated boundary was within ±5% of sequence length, as shown in Fig. 8, Frame (2). TN is assigned to the points that are correctly estimated as not boundary points, as shown in Fig. 8, Frame (3). Conversely, FP and FN are assigned to points that are falsely estimated as boundary points, as shown in Fig. 8, Frame (10), and falsely estimated as not boundary points, as shown in Fig 8, Frame (6), respectively. From these boundary evaluations, the precision, recall, and F-measure are computed as follows:

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}}, \tag{17}$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}}, \tag{18}$$

$$F = \frac{2P \cdot R}{P + R}, \tag{19}$$

where N_{TP}, N_{FP}, and N_{FN} represents the number of boundary points estimated as TP, FP, and FN respectively.

B. Segmentation of synthetic time-series data

First, we generated the following synthetic time-series data (shown in Figs. 9 and 10), and applied the segmentation methods to them.

- **Synthetic time series 1:** Unit sequences are a series of constant values randomly selected from a set of five values. The length of each unit sequence is randomly generated from a uniform distribution.

  $$k \sim \mathrm{uniform}(9, 13). \tag{20}$$

  Moreover, normal random values are added to each sequence as follows:

  $$x \sim x + \mathrm{normal}(x, 0.05). \tag{21}$$

TABLE I

SEGMENTATION RESULTS OF SYNTHETIC TIME SERIES 1.

	Hamming distance	Precision	Recall	F-measure	# of estimated classes
HDP-GP-HSMM	0.011	1.0	1.0	1.0	5
HDP-HMM	0.011	1.0	1.0	1.0	5
HDP-HMM+NPYLM	0.48	0.53	1.0	0.69	35
BP-HMM	0.36	1.0	0.94	0.97	4
Autoplait	0.79	0.00	0.00	0.00	3

TABLE II

SEGMENTATION RESULTS OF SYNTHETIC TIME SERIES 2.

	Hamming distance	Precision	Recall	F-measure	# of estimated classes
HDP-GP-HSMM	0.041	1.0	1.0	1.0	3
HDP-HMM	0.13	0.43	1.0	0.60	8
HDP-HMM+NPYLM	0.45	0.57	1.0	0.72	19
BP-HMM	0.038	1.0	1.0	1.0	3
Autoplait	0.35	0.17	0.15	0.16	2

The sequence shown in Fig. 9 is generated by connecting these unit sequences.

- **Synthetic time series 2:** The sequence shown in Fig. 10(b) is generated by randomly connecting the three sequences shown in Fig. 10(a).

Tables I and II show the segmentation results of synthetic time series 1 and 2. The results in these tables show that the Hamming distance of the proposed method is less than 0.1 and, moreover, all the precision, recall, and F-measure values are 1.0. These results indicate that our proposed HDP-GP-HSMM is able to estimate the segments almost completely correctly. Furthermore, the number of classes were correctly estimated as five in synthetic time series 1 and three in synthetic time series 2.

HDP-HMM was also able to correctly segment synthetic time series 1. This is because a simple model with a Gaussian emission distribution is suitable for this time series. However, HDP-HMM was not able to correctly segment synthetic time series 2 which did not follow a Gaussian distribution. In contrast, HDP-GP-HSMM and BP-HMM, which are more complicated models, were able to deal with such data.

C. Segmentation of motion-capture data

To evaluate the validity of the proposed method using more complicated real data, we applied it to the following three motion capture data series.

- **Chicken dance:** We used a sequence of motion capture data of a human performing a chicken dance from the CMU Graphics Lab Motion Capture Database.[1] The dance includes four motions, as shown in Fig. 11. The accelerations of both hands and feet, which were used in the experiment in [13], were also used in these experiments.
- **Exercise motion:** We also used three sequences of exercise motion capture data from subject 14 in the CMU Graphics Lab Motion Capture Database. This motion includes 11 motions, as shown in Fig. 12. The two-dimensional position (horizontal and vertical) of both hands and feet were used.

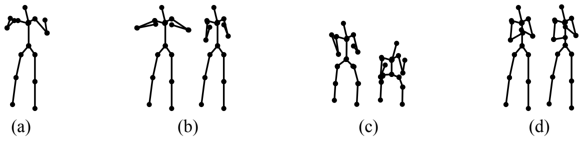


Fig. 11. Four unit motions included in the chicken dance: (a) beaks, (b) wings, (c) tail feathers, and (d) claps.

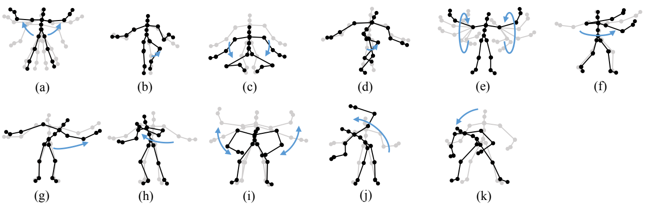


Fig. 12. Eleven unit motions included in the exercise motion: (a) jumping jack, (b) jogging, (c) squatting, (d) knee raise, (e) arm circle, (f) twist, (g) side reach, (h) boxing, (i) arm wave, (j) side bend, and (k) toe touch.

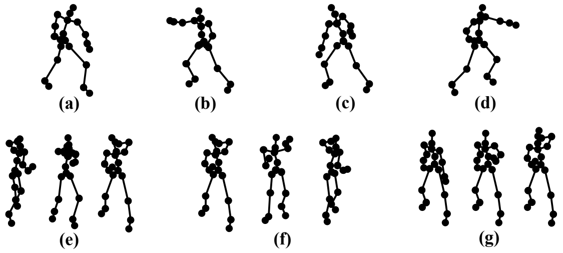


Fig. 13. Examples of motion included in the karate motion: (a) left lower guard, (b) right punch, (c) right lower guard, (d) left punch, (e) left upper guard, (f) right upper guard, and (g) preliminary motion.

- **Karate motion:** To include more complicated motions, we used karate motion capture from mocapdata.com[2]. Four sequences into which it is divided were used. Fig. 13 shows examples of the unit motions of the karate data. As for the other data series, the two-dimensional position of both hands were used.

All the data was preprocessed by down sampling and normalization. In the case of HDP-GP-HSMM, all the data was downsampled to reduce computational cost, and min-max normalization, in which the values are normalized to a range from -1 to 1, was applied. In the case of other methods, the segmentation was affected by normalization and down sampling for each data and, therefore, we choose a better normalization from min-max and z-score, and a better sample rate from fifteen fps, eight fps, four fps and two fps. The z-score normalization makes means and standard deviation of the normalized values zero and one respectively. The used values for down sampling and normalized methods are shown in Table III. HDP-GP-HSMM requires hyper parameters, and we used $\lambda = 12.0, \theta_0 = 1.0, \theta_1 = 1.0, \theta_2 = 0.0, \theta_3 = 16.0$, which were determined empirically for segmentation of 4 fps and min-max normalized data. Blocked Gibbs sampler was repeated 10 times, which was the number of times to converge the parameters.

[1] `http://mocap.cs.cmu.edu/`: subject 18, trial 15

[2] `http://www.mocapdata.com/`

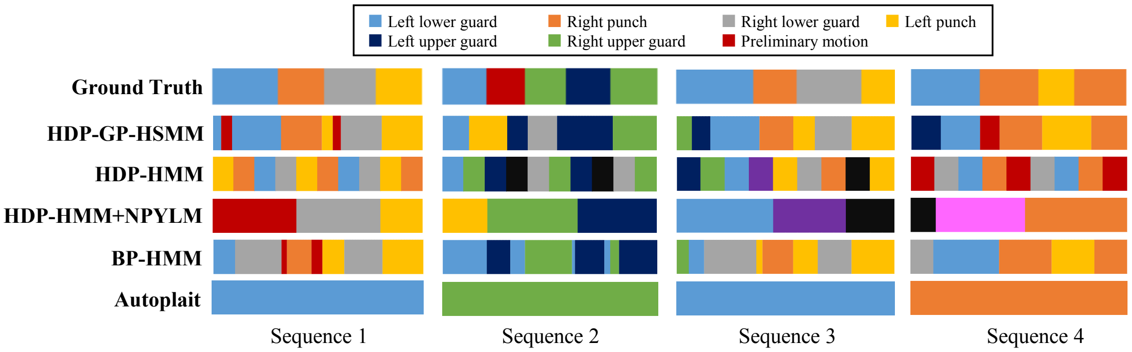


Fig. 14. Segmentation results for the karate motion.

TABLE III
VALUES USED FOR PREPROCESSING.

	Chicken dance	Exercise motion	Karate motion
HDP-GP-HSMM	4 fps / min-max	4 fps / min-max	15 fps / min-max
HDP-HMM	8 fps / z-score	2 fps / z-score	2 fps / min-max
HDP-HMM+NPYLM	15 fps / z-score	2 fps / min-max	4 fps / min-max
BP-HMM	15 fps / z-score	15 fps / z-score	15 fps / z-score
Autoplait	Original	15 fps / z-score	15 fps / z-score

TABLE IV
SEGMENTATION RESULTS FOR THE CHICKEN DANCE.

	Hamming distance	Precision	Recall	F-measure	# of estimated classes
HDP-GP-HSMM	0.13	1.0	0.86	0.92	4
HDP-HMM	0.46	0.39	0.71	0.50	4
HDP-HMM+NPYLM	0.65	0.63	0.71	0.67	8
BP-HMM	0.16	0.70	1.0	0.82	4
Autoplait	0.026	1.0	1.0	1.0	4

TABLE V
SEGMENTATION RESULTS FOR THE EXERCISE MOTION.

	Hamming distance	Precision	Recall	F-measure	# of estimated classes
HDP-GP-HSMM	0.31	0.38	0.95	0.55	10
HDP-HMM	0.82	0.070	1.0	0.13	14
HDP-HMM+NPYLM	0.63	0.61	1.0	0.76	26
BP-HMM	0.23	0.25	1.0	0.40	18
Autoplait	0.61	0.67	0.18	0.28	5

TABLE VI
SEGMENTATION RESULTS FOR THE KARATE MOTION.

	Hamming distance	Precision	Recall	F-measure	# of estimated classes
HDP-GP-HSMM	0.32	0.48	0.85	0.62	7
HDP-HMM	0.58	0.38	1.0	0.55	9
HDP-HMM+NPYLM	0.61	0.67	0.31	0.42	10
BP-HMM	0.39	0.34	0.75	0.47	7
Autoplait	0.60	0.00	0.00	0.00	3

Tables IV, V, and VI show the results of segmentation on each of the three motion-capture time series. The results in these tables show that HDP-GP-HSMM is able to accurately segment all three sets of motion capture data. BP-HMM and Autoplait are able to segment the chicken dance motion-capture data; however, exercise and karate motions are not segmented well by these methods. This is because simple and discriminative motions were repeated in the chicken dance, which makes this motion the simplest of the three motions in this experiment. In contrast, the Gaussian process use in HDP-GP-HSMM is a non-parametric method, which makes it possible to represent the complicated motion patterns in the exercise and karate motion-capture data.

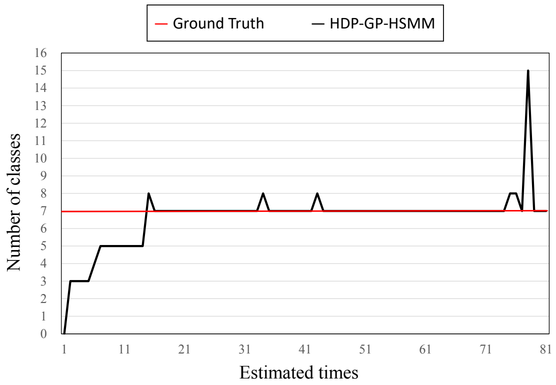


Fig. 15. Transition of the number of estimated classes.

Moreover, the number of chicken-dance and karate-motion classes could be correctly estimated by HDP-GP-HSMM. In the case of exercise motion, 10 classes were estimated by HDP-GP-HSMM, which was less than the correct number 11 because similar unit motions, such as motion F and motion H, as shown in Fig. 12, were classified as the same motion class. However, the number of classes estimated by HDP-GP-HSMM is still closer than those estimated by the other methods. Therefore, we conclude that HDP-GP-HSMM yielded better results in this case.

Fig. 14 illustrates the segmentation results for the karate motion. Here, the horizontal axis represents time steps, color reflects motion classes, and the top bar represents the ground truth of segmentation. It is clear that the segments and their classes estimated by HDP-GP-HSMM are similar to the ground truth. Furthermore, Fig. 15 shows the transition of the number of estimated classes by HDP-GP-HSMM. In this graph, the horizontal axis represents the number of iterations of the blocked Gibbs sampler, which fluctuates greatly in the earlier phase. However, it eventually converges to the correct value. Using HDP, the number of classes can be estimated while the segments and their classes are being estimated.

V. CONCLUSION

In this paper, we proposed the HDP-GP-HSMM, which makes it possible to segment time-series data and determine the number of classes by introducing an HDP into an HSMM whose emission distribution is a Gaussian process. In the proposed method, a slice sampler truncates the number of classes and makes it possible to apply forward filtering–backward sampling. The experimental results showed that the segments, their classes, and the number of classes can be estimated correctly using the proposed method. Moreover, the results also showed that the HDP-GP-HSMM is effective on various types of time-series data including synthetic and real sequences.

However, the computational cost HDP-GP-HSMM is very high because it takes $O(N^3)$ to learn N data points using a Gaussian process. Because of this problem, HDP-GP-HSMM cannot be applied to large datasets. We are planning to reduce the computational cost by introducing an approximation method for the Gaussian process proposed in [18], [19].

Moreover, we used the assumption where dimensions of the observation were independent to simplify the computation, and we consider this assumption is reasonable because the experimental results showed the proposed method works well. However, the dimensions of the observation are not actually independent and we also consider the dependency between the dimensions should be needed to model more complicated whole body motions. We consider that multi-output Gaussian processes can be used to represents dependencies between dimensions[20], [21].

Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR15E3 and JSPS KAKENHI Grant Number JP17K12758.

References

[1] T. Nakamura, T. Nagai, D. Mochihashi, I. Kobayashi, H. Asoh, and M. Kaneko, “Segmenting Continuous Motions with Hidden Semi-Markov Models and Gaussian Processes,” Frontiers in Neurorobotics, vol.11, no.67, 2017.

[2] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, “Hierarchical Dirichlet processes,” Journal of the American Statistical Association, vol.101, no.476, pp.1566–1581, 2006.

[3] J. V. Gael, Y. Saatci, Y. W. Teh, and Z. Ghahremani, “Beam Sampling for the Infinite Hidden Markov Model,” International Conference on Machine Learning, pp.1088–1095, 2008.

[4] M. Wächter and T. Asfour, “Hierarchical segmentation of manipulation actions based on object relations and motion characteristics,” International Conference on Advanced Robotics(ICAR), pp.549–556, 2015.

[5] R. Lioutikov, G. Neumann, G. Maeda, and J. Peters, “Probabilistic segmentation applied to an assembly task,” IEEE-RAS International Conference on Humanoid Robots, pp.533–540, 2015.

[6] W. Takano, and Y. Nakamura, “Real-time unsupervised segmentation of human whole-body motion and its application to humanoid robot acquisition of motion symbols,” Robotics and Autonomous Systems, vol.75, pp.260–272, 2016.

[7] A. Fod, M.J. Matarić, and O.C. Jenkins, “Automated derivation of primitives for movement classification,” Autonomous Robots, vol.12, no.1, pp.39–54, 2002.

[8] T. Shiratori, A. Nakazawa, and K. Ikeuchi, “Detecting dance motion structure through music analysis,” IEEE International Conference on Automatic Face and Gesture Recognition, pp.857–862, 2004.

[9] J. F. S. Lin, and D. Kulić, “Segmenting human motion for automated rehabilitation exercise analysis,” Conference of IEEE Engineering in Medicine and Biology Society, pp.2881–2884, 2012.

[10] M. J. Beal,, Z. Ghahramani, and C. E. Rasmussen, “The infinite hidden Markov model,” In Advances in Neural Information Processing Systems, pp.577–584, 2001.

[11] Taniguchi. T, and Nagasaka. S, “Double articulation analyzer for unsegmented human motion using Pitman–Yor language model and infinite hidden Markov model, ” In IEEE/SICE International Symposium on System Integration, pp.250–255, 2011.

[12] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, “Joint modeling of multiple related time series via the beta process,” arXiv preprint arXiv:1111.4226, 2011.

[13] Y. Matsubara, Y. Sakurai, and C. Faloutsos, “Autoplait: Automatic mining of co-evolving time sequences,” ACM SIGMOD International Conference on Management of Data, pp.193–204, 2014.

[14] E. B. Fox, E. B. Sudderth, M. I. Jordan and A. S. Willsky, “The sticky HDP-HMM: Bayesian nonparametric hidden Markov models with persistent states”, Annals of Applied Statistics, vol. 5, no. 2A, pp. 1020–1056, 2011

[15] J. Pitman, “Poisson-Dirichlet and GEM invariant distributions for splitand-merge transformations of an interval partition,” Combinatorics, Probability and Computing, vol.11, pp.501–514, 2002.

[16] Jensen. Claus S, Kjærulff. Uffe and Kong. Augustine, “Blocking Gibbs sampling in very large probabilistic expert systems,” International Journal of Human-Computer Studies, vol. 42, no.6, pp.647–666, 1995.

[17] K. Uchiumi, T. Hiroshi, and D. Mochihashi, “Inducing Word and Part-of-Speech with Pitman-Yor Hidden Semi-Markov Models,” Joint Conference of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp.1774–1782, 2015.

[18] D. Nguyen-Tuong, J. R. Peters, and M. Seeger, “Local Gaussian process regression for real time online model learning and control,” In Advances in Neural Information Processing Systems, pp.1193–1200, 2009.

[19] Y. Okadome, K. Urai, Y. Nakamura, T. Yomo, and H. Ishiguro, “Adaptive lsh based on the particle swarm method with the attractor selection model for fast approximation of Gaussian process regression,” Artificial Life and Robotics, vol.19, pp.220–226, 2014.

[20] Álvarez. Mauricio, Luengo. David, Titsias. Michalis and Lawrence. Neil, “Efficient multioutput Gaussian processes through variational inducing kernels,” Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp.25–32, 2010.

[21] Dürichen. Robert, Pimentel. Marco AF, Clifton. Lei, Schweikard. Achim and Clifton. David A, “Multitask Gaussian processes for multivariate physiological time-series analysis,” IEEE Transactions on Biomedical Engineering, vol.62, no.1, pp.314–322, 2015.