# Unsupervised Segmentation of Human Motion Data Using Sticky HDP-HMM and MDL-based Chunking Method for Imitation Learning

∗Tadahiro Taniguchi, Keita Hamahata (Ritsumeikan University)
and Naoto Iwahashi (NICT)

*Ritsumeikan Univeirsity, 1-1-1 Noji Higashi, Kusatsu, Shiga 525-8577, Japan, taniguchi@ci.ritsumei.ac.jp, hamahata@em.ci.ritsumei.ac.jp*

*National Institute of Information and Communications Technology,3-5 Hikari-dai Seika-cho Sohraku-gun Kyoto-fu 619-0289 Japan, naoto.iwahashi@nict.go.jp*

**Abstract**

We propose an unsupervised motion segmentation method for enabling a robot to imitate and perform various unit motions by observing unsegmented human motion. Natural unsegmented human motion data contains various types of unit motions, e.g., "waving good-bye", "walking" and "throw a ball". A robot has to segment the data and extract unit motions from the data to imitate the motions. In previous work, an ergodic hidden Markov model (HMM) was used to model unsegmented human motion. However, there are two main problems with the classical use of this model. The first problem is that setting an appropriate number of hidden states is difficult because how complex the motions contained in the learning data are and how many there are is unknown. The second problem is that we did not have an effective chunking method that could chunk elemental motions into meaningful unit motions without being captured by local minima. To overcome these problems, we develop an unsupervised motion segmentation method for imitation learning using a sticky hierarchical Dirichlet process HMM (sticky HDP-HMM), a nonparametric Bayesian model, and an unsupervised chunking method based on a Gibbs sampler and the minimal description length (MDL) principle of imitation learning of unsegmented human motion. We develop this chunking method to work with the sticky HDP-HMM and extract unit human motions. We conducted several experiments to evaluate this method. The proposed method could extract unit motions from unsegmented human motion data. The sticky HDP-HMM can be used to model unsegmented human motion more accurately than with a conventional HMM and simultaneously estimate the number of hidden states. We also evaluated the dependency of the HDP-HMM on the hyperparameters of the model.

*keywords*: unsegmented human motion, imitation learning, nonparametric Bayse model, sticky HDP-HMM, ergodic HMM

# 1 Introduction

Humans have an extraordinary ability to imitate learning compared with other animals including primates. Imitation learning is important not only for obtaining physical skills but also social skills such as expressing one's intentions using gestures. Enabling a robot to learn by imitation will result in a robot having many behaviors, e.g., "waving good-bye", "walking" and "throw a ball". If a robot can learn by imitating, it can automatically acquire new skills, similar to human children. Therefore, imitation learning in robotics has been gaining attention for this reason. In this paper, we develop an unsupervised motion segmentation method and the above-mentioned robotic imitation learning architecture to enable a robot to incrementally learn various unit motions by observing unsegmented human bodily motion. "When to imitate" is an important problem to be solved to develop such a learning architecture. When a learner tries to imitate another person's behaviors, the learner has to decide what segment of behavior to imitate from the demonstrator. For example, suppose a person approaches a robot, performs several motions, e.g., raising his/her hands, nodding several times, turning around and waving good-bye, and leaving. The displayed motion is unsegmented. The learner, i.e., a robot in our research, does not know which segment is a unit motion. Therefore, it must determine what to learn from the exhibited continuous motion. We focus on the problem of imitation learning from unsegmented motion. Therefore, we propose an unsupervised motion segmentation method for use with our proposed imitation learning architecture.

## 1.1 Imitation Learning in Robotics

Imitation learning in robotics has been studied as learning by watching, learning from demonstration, and programming by demonstration over the past two decades [1, 2]. Several studies on imitation learning in robotics aimed at enabling a robot to incrementally acquire various movements, skills, and controllers by only observing human motions have been conducted. Computational imitation learning was first studied in manipulator robotics to reduce manual programming of machines [3]. Shaal et al. gave a good review on the computational approach to imitation learning [3]. Nehaniv et al. insisted that imitation learning must address the following four key problems: "who to imitate", "what to imitate", "how to imitate", and "when to imitate" [4, 5]. Billard et al. mentioned that previous work has mostly focused on "how to imitate". The problem of "how to imitate" focuses on an imitation mechanism that precisely reproduces a demonstrator's actions assuming that task features and target movement are pre-specified. The issue with "what to imitate" is finding which of the task features are relevant for reproduction. They pointed out that "what to imitate" is an important problem in imitation learning in robotics. For example, when a robot observes a trajectory demonstrated by a human participant, it cannot determine "what to imitate" from the demonstration. For example, the robot can adopt either of three imitation strategies, end-point level, trajectory level, or path level. Each of these strategies reproduces different movement. Therefore, a robot has to observe demonstrations several times and infer the relevant features of the imitation task. Billard et al. developed a general policy for imitation

learning [6]. Sugiura et al. developed a reference-point-dependent probabilistic model to solve a similar problem [7]. These studies solved a part of the problem of "what to imitate". In contrast, "when to imitate" has not been sufficiently studied. Once an imitating agent determines the target demonstrator and starts observing its movement, the agent has to segment the all the demonstrator's behaviors to determine the behavior to be imitated a the beginning and end [4]. Schaal also pointed out that the imitation of complex movement sequences that involves leaning a sequence of primitives and when to switch between them is an important problem [3]. In the future, robots will function in our daily living environment by observing humans moving naturally around them. Their movements will be continuous and unsegmented. Therefore, it is important to segment continuous movement into reasonable unit motions to which humans can give some meanings and to extract those worth imitating. To achieve this, an unsupervised motion segmentation method that enables a robot to segment an unsegmented motion imitated from a human is necessary.

## 1.2 Motion segmentation

Even a one-year-old child can extract a unit motion from another's unsegmented motion and imitate it. This shows that human children can obviously solve the problem of "when to imitate". A robot has to segment target human motion data and extract meaningful motion segments to solve the same problem. Many segmentation methods for achieving this have been developed and evaluated.

The first and most classical type of these methods focuses on local features in continuous motion time series data. Rubin defined elementary motion boundaries, e.g. "start", "stop", and "impulse" [8]. Fod defined a time when the velocity of the joint degree crosses zero as a boundary of a movement [9]. These methods tend to segment human motion data into too many fragments by using only local features. In addition, these methods are easily affected by noise. As a result, unit motions segmented using these methods rarely have semantic meanings for human observers.

The second type focuses on local dynamics and the predictability of motion data. Many researchers have modeled human motion by using a hybrid linear dynamical system or Gaussian mixture model and segmented human motion based on them [10, 11, 12, 13, 14]. They assumed that linear dynamics, which is represented by a module of a hybrid dynamical system, generate local time series data and estimate their segments based on this assumption. By considering an interval during which a linear predictor is used as a segment corresponding to the linear predictor, a hybrid dynamical system can be used as a segmentation method. An expectation-maximization (EM) algorithm and other learning algorithms make it possible to automatically learn the linear predictor's parameters. However, in most cases, a movement represented by single linear dynamics or a Gaussian distribution does not have a semantic meaning for human observers. These methods also segments human motion data into too many fragments.

The third type uses more complex nonlinear predictors that also use short-term context information in some models to obtain a meaningful unit motion. Ogata and other researchers use a recurrent

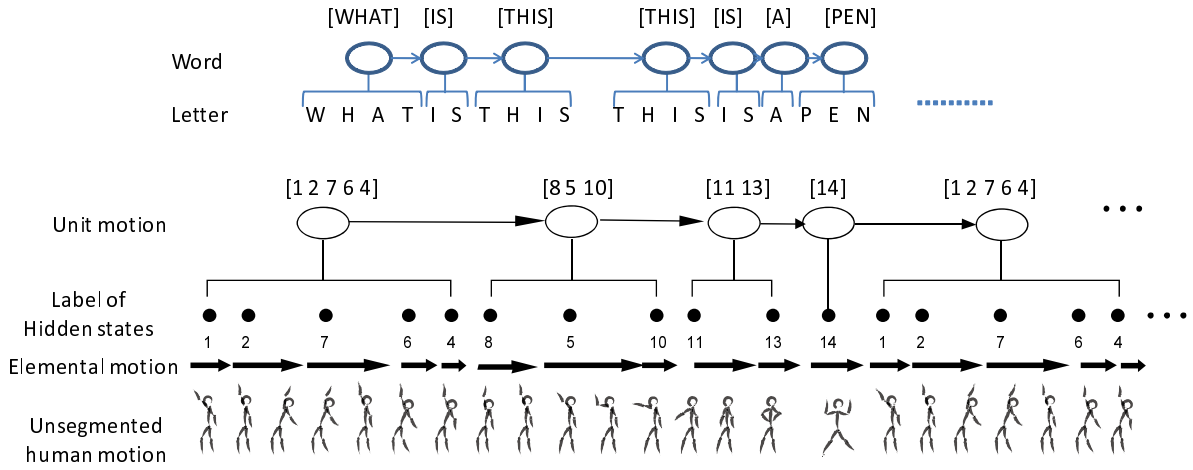


Figure 1: Assumption of double articulation in motion segmentation

neural network (RNN) and its extended learning architectures, e.g. a recurrent neural network with parametric bias (RNNPB) [15, 16, 17], multi-time scale recurrent neural network (MTRNN) [18], and a mixture of RNN experts [19]. Okada et al. used a polynomial function as a local predictor and developed a dynamics-based self organization map (DBSOM) [20]. However, the complexity of the obtained segments depends on the characteristics and degrees of freedom (DOFs) of the unit nonlinear predictors. In addition, the obtained parameters of these learning architectures are usually difficult for us to understand from a mathematical viewpoint. For example, synaptic weights in an RNN are difficult to analyze after several motions are obtained with the learning architecture. Nonlinear models based on Gaussian processes and kernel techniques have successfully been applied to modeling of human bodily motion data [21]. However, imitation learning from unsegmented motion must simultaneously solve the problem of "when to imitate" and model human motion data.

The fourth type finds repeated segments from a continuous time series. Kadone uses autocorrelation to find repeated motion segments [22], and Chippa proposed a Bayesian statistical model and variational Bayesian estimation algorithm [23]. However, these approaches usually require much computational time because they do not make the most use of local dynamical information. Therefore, they have to compare every possible segment in the target unsegmented motion potentially.

To overcome these problems, we combined the second and fourth types and derived an efficient unsupervised motion segmentation method based on the notion of *double articulation*. We introduce this notion in the next subsection.

## 1.3 Double articulation

In the context of motion segmentation, Barbic distinguished between *high-level behavior* and *low-level behavior* [12]. Low-level behavior is represented as the output of linear dynamics. However, Barbic and we are interested in semantically meaningful units of behaviors. We call such segments high-level

behavior, such as walking, running, sitting, throwing a ball, and swinging a stick. Obviously, high-level behavior is more complex than low-level behavior. The third and fourth types of methods discussed in the previous subsection were proven to extract higher-level behaviors than the first and second types. However, they have several problems. Therefore, we propose a method for extracting high-level behavior by connecting several low-level behaviors extracted using an advanced version of the ergodic hidden Markov model (HMM). The method is based on double articulation, which is well known in semiotics.

Figure 1 shows the basic concept of double articulation in our unsupervised motion segmentation method. We, humans, have a double articulation structure in our spoken sentences and other many semiotic data. First, a spoken auditory signal is segmented into letters or phonemes. In the next process, the phonemes are chunked into words. Usually, we do not give any meanings to phonemes, but give certain meanings to words. We assume that humans assume and use a similar structure of segments for recognizing and imitating human motions. This is our basic assumption. We assume that unsegmented motion is segmented to low-level behaviors based on its linearity or its locality of distribution in its state space. We call low-level behavior *elemental motion* in this paper. Next, elemental motions are chunked into a *unit motion*, which corresponds to words in spoken language. We propose unsupervised motion segmentation method which extracts unit motions based on this idea. In the next subsection, we give an overview of our proposed imitation learning method.

## 1.4 Imitation learning architecture overview

We give an overview of our proposed motion segmentation method and imitation learning architecture in this subsection. Figure 2 shows a schematic overview of the overall architecture.

First, a large amount of high-dimensional motion data are observed by a robot and recoded. Singular value decomposition (SVD) reduces their dimensionality as preprocessing. This reduces successive computational costs and extracts low-dimensional features, which mainly relate to unit motions embedded in unsegmented motions.

A sticky hierarchical Dirichlet process HMM (sticky HDP-HMM) [24] is used to segment and model the target preprocessed motion data (described in Section 2). By using a sticky HDP-HMM, a robot can obtain elemental motions and sequences of labels of hidden states without fixing the number of types of elemental motions. We call a sequence of labels of hidden states that corresponds to observed unsegmented motion data a *document* (see Figure 2). After obtaining a document, it is segmented into a sequence of words, chunked letters, by using our unsupervised chunking method based on minimal description length (MDL) principle (described in Section 3). The document is segmented into words that correspond to unit motions.

Taniguchi et al. proposed an imitation learning architecture [25] for unsegmented human motion. They use an ergodic switching autoregressive model (SARM) [26], which is a variation of an HMM using an EM algorithm, to estimate the architecture's model parameters for abstracting elemental motions. An ergodic HMM is an HMM whose hidden states are all mutually connected. Usually, an EM algorithm

is adopted to learn the HMM parameters. It is known that the ergodic HMM's learning process has fertile local minima because it has high complexity and flexibility. Therefore, a learning method that guarantees global optimality is expected to be developed. The Markov Chain Monte-Carlo (MCMC) algorithm was introduced to overcome this problem [27]. In addition, it is difficult to determine the number of hidden states beforehand in imitation learning from unsegmented motion compared with segmented motion because it is difficult to assess the complexity and variety of unsegmented motion before the learning process. To determine the number of hidden states, model selection methods using several model selection criteria, e.g. minimal description length (MDL), Bayesian information criterion (BIC), and Akaike information criterion (AIC), have been used. However, these model selection methods require off-line computation and repeated learning computation for all model parameter settings. This requires such a large amount of computational cost and time that these models cannot be used for developing an autonomous robot's on-line adaptation. To overcome this problem, a learning model that can adequately and automatically determine the number of hidden states in an on-line manner is required. Recently, a nonparametric Bayesian model that can estimate the number of hidden states in an on-line manner has been gaining attention. Therefore, ourimitation learning architecture uses a sticky HDP-HMM [24], which is a nonparametric Bayesian model, to enable imitation learning from unsegmented human motion.

After obtaining elemental motions, the learning method must chunk elemental motions into unit motions. Taniguchi et al. used a heuristic chunking method that does not guarantee globally optimal chunking to chunk elemental motions into unit motions. We adopted minimal description length criteria and a Gibbs sampler to search globally optimal chunking results and obtain reasonable unit motions.

The chunked elemental motions correspond to a sequence of hidden states of an HMM. A set of a certain sequences of hidden states can be interpreted as a left-to-right HMM, which has been used to model human body movement in many studies on imitation learning in robotics [28, 7, 29, 30]. Many motion generation methods have been developed for left-to-right HMM modeling of human motion. Our motion segmentation method based on double articulation asuumption can use these developed generation techniques. Therefore we limit the discussion to motion generation techniques. As a result, the both of two learning processes, the sticky HDP-HMM segmenting unsegmented human motion data in continuous state spaces into elemental motions and unsupervised chunking method chunking elemental motions to unit motions in discrete symbolic state spaces, use the MCMC algorithm, which can theoretically prevent local minima in their learning process.

Another promising nonparametric Bayesian model is the infinite hierarchical HMM (IHHMM) proposed by Hellor et al. [31]. The IHHMM provides flexible modeling of sequential data by allowing a potentially unbounded number of levels in hierarchy. However, the IHHMM is too flexible to extract concrete unit motions from unsegmented human motions. A hidden state in every hierarchy has an ergodic HMM having the emission distribution over discrete symbols. An ergodic HMM can represent many types of concrete sequence discrete symbols. Therefore, an HMM obtained in a high level of the hierarchy does not always correspond to a single unit motion. When we use an HMM in imitation

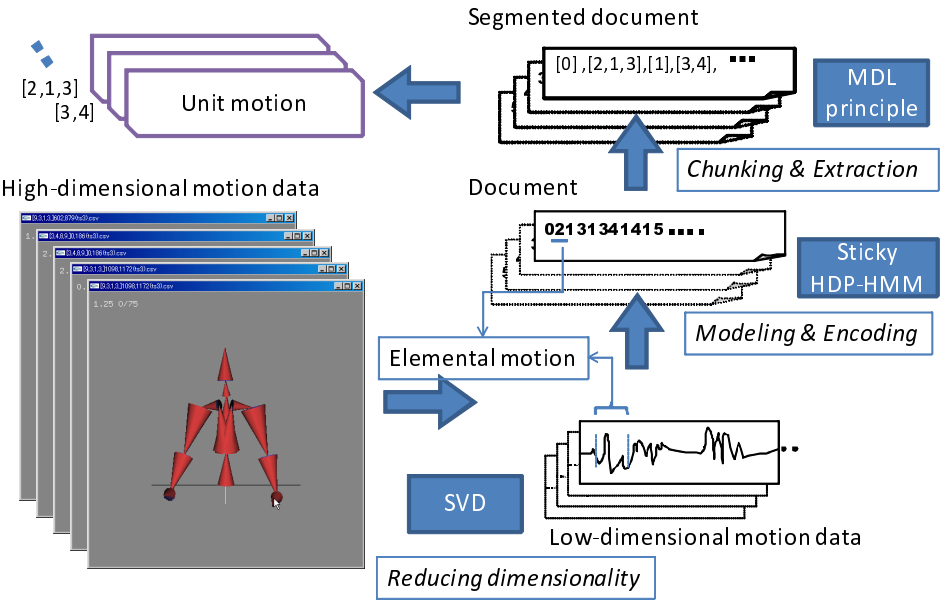


Figure 2: Overview of proposed imitation learning architecture

learning in robotics, we usually need a left-to-right transition model, especially in the motion generation process. Our unsupervised motion segmentation method uses an ergodic HMM at the start. However, a sequence of hidden states can be interpreted into a left-to-right HMM as a result. This is why we propose our segmentation method based on the notion of double articulation.

This paper is organized as follows. In the second section, we introduce nonparametric Bayesian models and the sticky HDP-HMM. In the third section, we describe our unsupervised chunking method based on the MDL principle. In the fourth section, we discuss an experiment in which we applied our method to human upper body motion data and evaluated it. In the fifth section, we discuss the results of the experiment and evaluation. Finally, we conclude this paper in the sixth section.

## 2 Nonparametric Bayesian Model

A nonparametric Bayesian model is a flexible machine learning framework that can automatically determine model complexity by referring to the distribution and hidden structure of target data. In this section, we introduce the Dirichlet process and other nonparametric models and introduce the sticky HDP-HMM used in our imitation learning architecture. We then describe a numerical blocked Gibbs sampler algorithm of the sticky HDP-HMM proposed by Fox [24].

### 2.1 Dirichlet process

Various parametric mixture models having a fixed number of hidden states are used for modeling and clustering spoken language, human motion, and other feature vector steams. However, the model

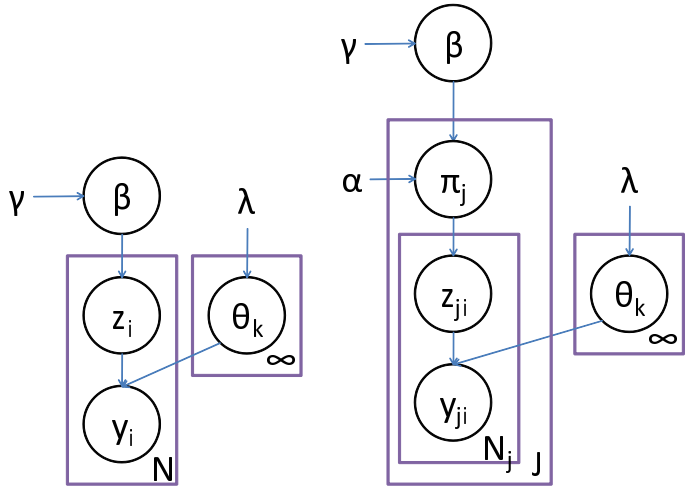


Figure 3: Graphical model of Dirichlet Process Mixture

selection problem still remains. Both the EM algorithm for maximizing likelihood, maximum a posteriori (MAP) estimation, and MCMC algorithm for the Bayesian model share the same problem in model selection and parameter search. Model selection methods, such as the MDL principle, are often used to determine the number of hidden states. However, these methods require off-line computation and a large amount of computational cost because we have to compute learning experiment for all settings of the number of hidden states.

In contrast, the Dirichlet process model, which can automatically determine an adequate number of hidden states, is gaining attention. The Dirichlet process, which is a generative model of hidden states, is a stochastic process generating a potentially infinite number of probability distributions [32]. Mathematically, the Dirichlet process is considered an infinite dimensional Dirichlet distribution. A graphical model of the Dirichlet process mixture (DPM), which is a mixture model constructed using the Dirichlet process, is shown in Fig. 3(left).

In an ordinal Bayesian parametric mixture model, a finite dimensional Dirichlet distribution generates a multinomial distribution, which outputs a hidden state and outputs a probabilistic distribution corresponding to a hidden state emitting an observable variable[1]. However, in a DPM, the number of hidden states is determined depending on observed data. In other words, the Dirichlet process presumes an infinite dimensional Dirichlet distribution as a prior distribution, and a finite number of its dimensions was found to describe the observed data as a result.

Sethuraman proposed the stick-breaking process (SBP) to give a concrete construction process of a Dirichlet process [33]. The SBP constructs an infinite dimensional multinomial distribution by breaking

[1]If the mixture model is a Gaussian mixture model (GMM), the emission distribution is a Gaussian distribution.

a stick having a probability of one with a sequence of rate output from a beta distribution.

$$v_l|\gamma \sim \text{Beta}(1,\gamma) \qquad l = 1,2,\ldots \tag{1}$$

$$\beta_k = v_k \prod_{l=1}^{k-1}(1-v_l) \qquad k = 1,2,\ldots \tag{2}$$

The procedure forms an infinite dimensional multinomial distribution $\beta = (\beta_1, \beta_2, \cdots)$, where $\gamma$ is a concentration parameter. For convenience, we write $\beta \sim \text{GEM}(\gamma)$ (GEM stands for Griffiths, Engen and McCloskey, e.g., [34]). In the generative model of the DPM, a data sample $y_i$ is output from a selected emission distribution with a parameter $\theta_k$ generated from a prior distribution of the emission distribution. The index of emission distribution $k$ is selected based on the multinomial distribution generated from the SBP.

On the other hand, the Dirichlet process, which is expressed by using the P'olya urn scheme [35], is called the Chinese Restaurant Process (CRP). The metaphor is as follows. Each data sample corresponds to a customer visiting a restaurant, each cluster corresponds to one table, and a parameter of a mixture component corresponds to a dish served at a table. The $i$-th customer sits at a table with a probability proportional to the number of customers who have already been seated, or sits at a new table with the probability proportional to $\frac{\gamma}{i-1+\gamma}$. The P'olya urn scheme does not directly refer to the construction of a multinomial distribution. It only refers to draws from the distribution. One of the most important properties of the CRP is exchangeability, which enables us to easily apply a Gibbs sampler to the DPM.

## 2.2 Hierarchical Dirichlet Process

A hierarchical Dirichlet process takes distribution $G_0$, generated using the Dirichlet process with base measure $H$, as its base measure. The Dirichlet process using the shared base measure $G_0$ generates several potentially infinite dimensional multinomial distributions $G_j$, where $j$ means the index of the Dirichlet process in the lower layer. This means that each $G_j$ shares atoms [36].

A graphical model of the hierarchical DPM (HDPM) is shown in Fig. 3(right). In contrast with the CRP for DPM, a generative process of the HDPM can be described as a Chinese Restaurant Franchise (CRF). In the CRF, there are $J$ restaurants. Every restaurant belongs to an HDP franchise group and potentially has an infinite number of tables, i.e., clusters, and visiting customers, i.e., observed data samples, sit at the tables. The $j$-th restaurant's $i$-th customer sits at a table $t_{ji} \sim \pi_j$, and a dish, i.e., an emission distribution parameter, $k_{jt} \sim \beta$ is served. Each observed data sample $y_{ji}$ is generated from an emission probability whose parameter is $\theta'_{ji} = \theta^*_{jt_{ji}} = \theta_{k_{jt_{ji}}}$. As a whole, the HDP generates $J$ number of DPs partially sharing the same atoms (see [36] for the details). Many extended HDP models were developed and applied to several applications, e.g., document clustering, document retrieval, and speaker diarization [37]. One of them is the infinite hidden Markov model (iHHM), which is used to model time series data.

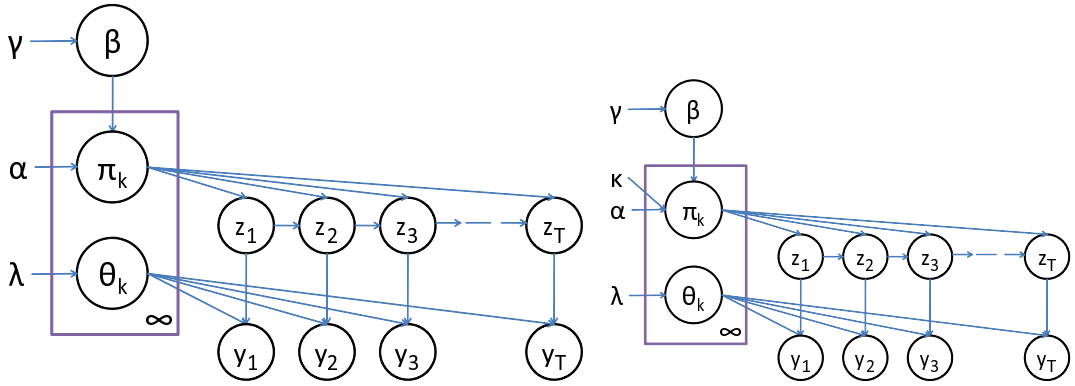


Figure 4: Graphical model of (left) HDP-HMM and (right) sticky HDP-HMM

## 2.3 Infinite Hidden Markov Model

The iHMM, proposed by Beal [38], is a first nonparametric Bayesian statistical model that can be substituted for an HMM whose selection probability of hidden states is temporally related in a Markovian manner. A potentially infinite number of hidden states is assumed with the iHMM. Through its inference process, the iHMM can flexibly estimate the number of hidden states. In a conventional HMM, the number of hidden states is fixed. In contrast, the iHMM has a potentially infinite number of hidden states. The HDP-HMM is a flexible statistical model whose number of hidden states is determined adaptively depending on given training data. However, it does not have an adequate generative model and an efficient inference algorithm.

The [36] extends the HDPM into the hierarchical Dirichlet process-hidden Markov model (HDP-HMM), which is an adequate generative model for iHMM. A graphical model of the HDP-HMM is shown in Fig. 4(left). We refer to this model as HDP-HMM in this paper.

In the HDP-HMM, the SBPs GEM($\gamma$) having the concentration parameter $\gamma$ produces $\beta$, which produces $\pi_k$ for all hidden states. $\pi_k$ is a multinomial distribution corresponding to each hidden state. In a generative process, the next state is selected using a multinomial distribution corresponding to the hidden states.

The multinomial distributions correspond to a transition probability matrix in a conventional HMM. This means that the HDP-HMM has transition matrices having potentially infinite dimensions. Therefore, a left-to-right structure is generally not presumed in the HDP-HMM. A hidden state can transit to every state stochastically, which means that the HDP-HMM is an ergodic HMM by definition. This causes another problem. State transition to other hidden states occurs frequently in the HDP-HMM. This comes from the fact that $\pi_k$ does not have any self transition bias. However, from practical use of HMMs in continuous dynamical systems, e.g., modeling and segmenting, spoken language, human motion, and data from sensory networks, a hidden state is expected to be sustained for a certain number of time steps. For example, a hidden state corresponds to a phoneme in speech recognition, an elemental motion in human motion, and a target system's elemental activation pattern in a sensory network are

expected to be sustained for a certain period. In conventional HMMs, many researchers and engineers have been overcoming the problem by biasing self transition probability and by setting initial parameters to introduce practical state persistency to the model.

Beal introduced self-transition bias to the iHMM. However, they did not make the self-transition bias involved in a whole generative model. To theoretically give an adequate self transition bias to the HDP-HMM, Fox et al. [24] proposed a sticky HDP-HMM by giving the HDP-HMM a hyperparameter *kappa*, which increases self transition probability.

## 2.4 sticky HDP-HMM

Fox et al. [24] proposed a sticky HDP-HMM with a self-transition bias [24]. This model is an extension of the HDP-HMM. By biasing the self transition probability, this sticky HDP-HMM can reduce the frequency of transition among hidden states. Therefore, this model is more effectively used to model and segment a continuous observed real data stream, e.g., speaker diarization and speech recognition. If the segmentation process, outputting elemental motions, produces too many fragments, i.e., too many state transitions, the posterior word extraction process does not work well and cannot extract unit motions. For our purpose, the stickiness the sticky HDP-HMM provides is important.

Fox also describes a numerical computation algorithm using a blocked Gibbs sampler. Straightforward application of the forward filtering-backward sampling algorithm for an HMM [27] to the iHMM is not feasible because it is impossible to accumulate forward messages for an infinite number of hidden states. Therefore, halting an SPB and truncating the number of hidden states are unavoidable. Fox et al. proposed a blocked Gibbs sampler by adopting weak-limit approximation. This accelerates the inference sampling process in the HDP-HMM. Practically, the approximation is not so problematic for the purpose of motion learning. Therefore, we adopted the blocked Gibbs sampler proposed by Fox et al. [24]. The precise formulation, derivation, and discussion of the sticky HDP-HMM and its blocked Gibbs sampler is omitted in this paper [2]. As an alternative, Gael et al. developed an efficient inference technique for the iHMM called the beam sampler [39]. The beam sampler can sample true posterior distribution without any approximations by incorporating a slice sampling technique. In this paper, we use the weak-limit approximation by Fox et al. for practical use.

The sticky HDP-HMM algorithm is described as follows [24].

Given a previous set of state-specific transition probabilities $\boldsymbol{\pi}^{(n-1)}$, the global transition distribution $\beta^{(n-1)}$, and emission parameters $\boldsymbol{\theta}^{(n-1)}$, set

1. $\boldsymbol{\pi} = \boldsymbol{\pi}^{(n-1)}$ and $\boldsymbol{\theta} = \boldsymbol{\theta}^{(n-1)}$. Working sequentially backwards in time, calculate messages $m_{t,t-1}(k)$.

[2] For more information, see Fox et al.'s research [24].

**(a)** For each $k \in \{1, \ldots, L\}$, initialize messages to

$$m_{T+1,T}(k) = 1$$

**(b)** For each $t \in \{T-1, \ldots, 1\}$ and $k \in \{1, \ldots, L\}$, compute

$$m_{t,t-1}(k) = \sum_{j=1}^{L} \pi_k(j)\, \mathrm{N}(y_t; \mu_j, \Sigma_j) m_{t+1,t}(k)$$

2. Sample state assignments $z_{1:T}$ working sequentially forward in time, starting with $n_{jk} = 0$ and $Y_k = \emptyset$ for each $(j,k) \in \{1, \ldots, L\}^2$:

**(a)** For each $k \in \{1, \ldots, L\}$, compute the probability

$$f_k(y_t) = \pi_{z_{t-1}}(k)\, \mathrm{N}(y_t; \mu_k, \Sigma_k) m_{t+1,t}(k)$$

**(b)** Sample a state assignment $z_t$:

$$z_t \sim \sum_{k=1}^{L} f_k(y_t) \delta(z_t, k)$$

**(c)** Increment $n_{z_{t-1} z_t}$ and add $y_t$ to the cached statistics for the new assignment $z_t = k$:

$$Y_k \leftarrow Y_k \oplus y_t$$

3. Sample the auxiliary variables $\boldsymbol{m}, \boldsymbol{w}$, and $\bar{\boldsymbol{m}}$

**(a)** For each $(j,k) \in \{1, \ldots, L\}^2$, set $m_{jk} = 0$ and $n = 0$. For each customer in restaurant $j$ eating dish $k$, that is for $i = 1, \ldots, n_{jk}$, sample

$$x \sim \mathrm{Ber}(\frac{\alpha\beta_k + \kappa\delta(j,k)}{n + \alpha\beta_k + \kappa\delta(j,k)}).$$

Increment $n$, and when $x = 1$, increment $m_{jk}$.

**(b)** For each $j \in \{1, \ldots, K\}$, sample the number of override variables in restaurant $j$:

$$w_{j\cdot} \sim \mathrm{Binominal}(m_{jj}, \frac{\rho}{\rho + \beta_j(1-\rho)})$$

Set the number of informative tables in restaurant $j$, considering dish $k$, to:

$$\bar{m}_{jk} = \begin{cases} m_{jk} & j \neq k \\ m_{jj} - w_j & j = k \end{cases} \tag{3}$$

4. Update the global transition distribution $\beta$ by sampling

$$\beta \sim \mathrm{Dir}(\gamma/L + \bar{m}_{\cdot 1}, \ldots, \gamma/L + \bar{m}_{\cdot L})$$

5. For each $k \in \{1, \ldots, L\}$, sample a new transition distribution and emission parameter based on

the sample state assignments

$$\begin{aligned}\pi_k &\sim \mathrm{Dir}(\alpha\beta_1 + n_{k1}, \ldots, \\ &\quad \alpha\beta_k + \kappa + n_{kk}, \ldots, \alpha\beta_L + n_{kL}) \\ \theta_k &\sim p(\theta|\lambda, Y_k)\end{aligned}$$

6. Fix $\boldsymbol{\pi}^{(n)} = \boldsymbol{\pi}, \beta^{(n)} = \beta, \boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}$ .

7. Optionally, resample the hyperparameters $\gamma, \alpha$, and $\kappa$, as described by Fox et al. [24].

When we assume the emission distributions corresponding to hidden variables are Gaussian distributions, the emission parameter is $\theta_k = \{\mu_k, \Sigma_k\}$. In this case, if the Gaussian prior on the mean $\mu_k$ is $\mathrm{N}(\mu_0, \Sigma_0)$ and the inverse-Wishart prior on the covariance $\Sigma_k$ is $\mathrm{IW}(\nu, \Delta)$, they are sampled as follows.

$$\Sigma_k|\mu_k \sim \mathrm{IW}(\bar{\nu}_k\bar{\Delta}_k, \bar{\nu}_k) \tag{4}$$

$$\mu_k|\Sigma_k \sim \mathrm{N}(\bar{\mu}_k, \bar{\Sigma}_k), \tag{5}$$

where

$$\bar{\nu}_k = \nu + |Y_k| \tag{6}$$

$$\bar{\nu}_k\bar{\Delta}_k = \nu\Delta + \sum_{y_t \in Y_k} (y_t - \mu_k)(y_t - \mu_k)' \tag{7}$$

$$\bar{\Sigma}_k = (\Sigma_0^{-1} + |Y_k|\Sigma_k^{-1})^{-1} \tag{8}$$

$$\bar{\mu}_k = \bar{\Sigma}_k(\Sigma_0^{-1}\mu_0 + \Sigma_k \sum_{y_t \in Y_k} y_t). \tag{9}$$

$Y_k = \{y_t|z_t = k\}$ is a set of observed data samples whose hidden state is $z_t = k$, and $|Y_k|$ is its cardinality. The conditional posterior distributions of Gaussian parameters can be calculated analytically. Therefore, we can obtain a new sample based on this sampling procedure.

The concentration parameters $\alpha, \gamma$ can be estimated by putting Gamma distributions as hyper prior distributions. However, we do not treat hyperparameter estimation in this paper for simplicity. Instead, we evaluate the effect of the hyperparameters in several experiments.

We can obtain model parameters and sequences of hidden states, which are described as a "document" written in a set of letters representing hidden states of the sticky HDP-HMM, by applying the above algorithm.

# 3 Chunking method based on minimal description length

We assume that a unit motion consists of a chunk of elemental motions. This corresponds to the relationship between a spoken word and phoneme. Taniguchi et al. [25, 40] proposed an imitation learning architecture that enables a robot to extract characteristic unit motions from unsegmented

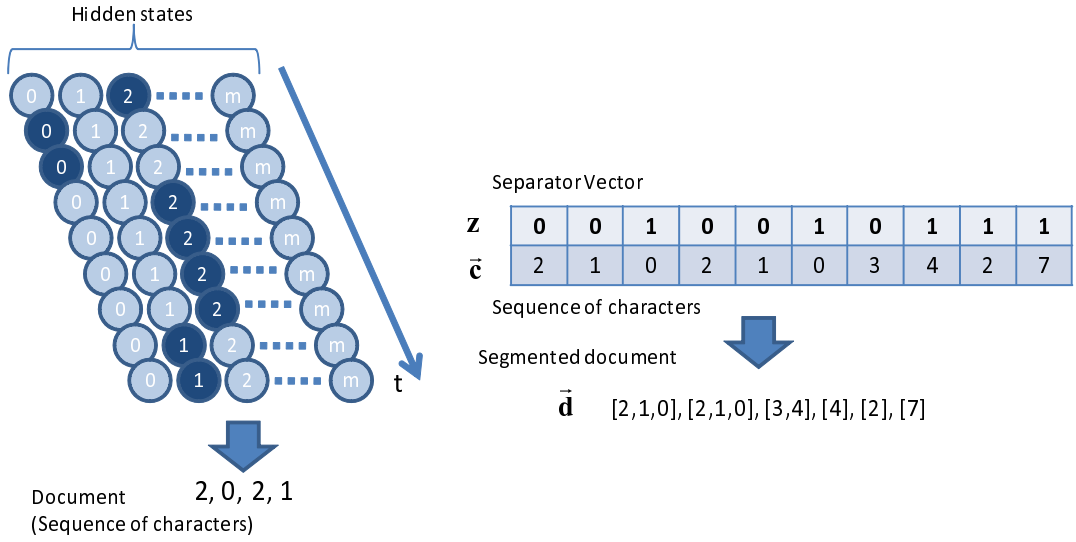


Figure 5: (Left) sequence of hidden states is transformed into document. (Right) separator vector determines segmentation of given document.

hand movement by using the keyword extraction method proposed by Umemura [41]. However, this keyword extraction method is a heuristic methodology in which the segmentation results highly depend on several hand-coded parameters and initial conditions. In contrast, Tanaka et al. developed a motion segmentation method [42] based on the MDL principle. Based on the same idea, we propose a simple chunking method based on the MDL principle. The method uses a Gibbs sampler as an optimization algorithm by constructing a probabilistic model of chunking based on MDL.

### 3.1 Description length

We assume that a document is encoded in two steps, encoding it using words included in a dictionary then encoding the dictionary using letters. This two-part code length depends on both segmented documents and a dictionary.

Figure 5 shows how a segmented document is obtained. First, obtained sequences of indices of hidden states are converted into a document by omitting consecutive indices.

A document $d$ is represented by a sequence of characters $\vec{c} = (c_1, c_2, \cdots, c_L)$, where $L$ is the number of characters of the document. A segmented document $\vec{d_i}$ is described as a sequence of words $\{w_j\}$ contained in a dictionary *dict.*. A word $w_j$ is described as a sequence of letters $w_j = (a_1^j, a_2^j, \cdots, a_{m_j}^j)$, where $m_j$ is the length of the word $w_j$. In this case, the description length of a corpus $\vec{d} = \bigoplus_i \vec{d_i}$, which is a direct sum of documents $\vec{d_i}$, becomes

$$L(\vec{d}) = -\sum_{w_j \in dict.} \#(w_j|\vec{d}) \log(p(w_j|\vec{d})) \tag{10}$$

$$p(w_j|\vec{d}) = \#(w_j|\vec{d}) / \sum_k \#(w_k|\vec{d}), \tag{11}$$

where $\#(w_j|\vec{d})$ represents the frequency of word $w_j$ in the segmented corpus $\vec{d}$. For example, if $\vec{d} = ([a\ b][a\ b][a][a\ b][c\ a][b])$, $\#([a\ b]|\vec{d}) = 3$. We define the description length $L(dict.)$ of a dictionary $dict. = \{w_i|w_i \in \vec{d}\}$, which contains every word appearing in the target segmented corpus, as follows.

$$L(dict.) = -\sum_j \#(a_j|dict.)\log(p(a_j|dict.)) \tag{12}$$

$$p(a_j|dict.) = \#(a_j|dict.)/\sum_k \#(a_k|dict.) \tag{13}$$

The sum of the two description lengths $L^{double}(\vec{d}) = L(\vec{d}) + L(dict.)$ becomes a two-part code description length.

We assume a dictionary that minimizes the two-part code description length is a true dictionary. Therefore, we have to search for optimal segmentation that outputs a dictionary minimizing a two-part code description.

### 3.2 Inference of separators

To minimize the two-part code description length, the learning system has to find an optimal dictionary. Taniguchi et al. adapted a local search method for organizing a dictionary in similar settings [43]. However, a deterministic local search of a dictionary is easily captured by local minima. In this section, we introduce a simple probabilistic model for document segmentation and an inference algorithm based on MCMC algorithms. We introduce a hidden variable, called separator vector $z = (z_1, z_2, \cdots, z_L)$ $z_l \in \{0, 1\}$, which separates each word in a document. For example, if $d = (a\ b\ a\ b\ a\ a\ b\ c\ a\ b)$ and $z = (0, 1, 0, 1, 1, 0, 1, 0, 1, 1)$. a segmented document $\vec{d}$ becomes $\vec{d} = ([a\ b][a\ b][a][a\ b][c\ a][b])$, If $z_l = 1$, a separator is inserted between $c_l$ and $c_{l+1}$. Figure 5 shows these relationships. If all the separator vectors $z_i$ for documents in a corpus are determined, segmented documents and a dictionary corresponding to the corpus is uniquely obtained. We define a simple probabilistic model for generating a two-part coded corpus, i.e., a segmented document and a dictionary. A two-part coded corpus $\vec{d}$ with probability $p(\vec{d})$, which exponentially decreases while the two-part code description length increases, can be defined as follows.

$$p(\vec{d}|\vec{c}) \propto exp(-\beta L^{double}(\vec{d})) \tag{14}$$

$$p(z|\vec{c}) \equiv p(\vec{d}|\vec{c}) \tag{15}$$

where $\beta$ is the inverse temperature parameter. This probabilistic distribution outputs a dictionary, which reduces the two-part code description length with high probability. The problem we have to solve is finding $z$ that maximizes $p(z|\vec{c})$. It is impossible to calculate $p(z)$ for all possibilities of $z$ because of the exponentially huge possibility of $z$. Therefore, we use the Gibbs sampler for sequentially sampling $z_l$.

$$p(z_l = b|z_{/l}, \vec{c}) \propto p(z_l = b, z_{/l}|\vec{c}) \qquad (b = 0, 1), \tag{16}$$

where $z_{/l} = z_1, z_2, \cdots, z_{l-1}, z_{l+1}, \cdots, z_L$. By sampling $z_l$ and sequentially updating the value, we can approximately obtain an i.i.d. sample of $p(z|\vec{c})$. By sampling a sufficient number of $z$ and choosing $\hat{z}$, which has the highest posterior probability, we can find an adequate sample, whose document $\vec{d}$ is efficiently segmented, and organize a dictionary.

### 3.3 Unit motion extraction

After obtaining an adequate dictionary, keywords corresponding to unit motions should be extracted from the set of words. This additional extraction process is required because the obtained dictionary is only a bag of words. We introduce natural heuristic criteria that satisfy unit human motion corresponding to a keyword.

1. A keyword appear more than once,
2. $t_{min}/t_{max} > r_\tau$, and
3. $t_{min} > \tau_{min}$,

where $t_{min}$ is the minimal length of a time series corresponding to the word, and $t_{max}$ is the maximal length of a time series corresponding to the word. The first criterion means that a characteristic unit human motion appears several times. The second one means a unit human motion does not have too large of a temporal variance. The third one means that a unit human motion is sustained for a certain period. Words in the obtained dictionary satisfying these criteria are extracted as keywords. After obtaining keywords, a robot can display a motion corresponding to the keyword by showing the original observed motion data or by using several motion generation methods [7, 28].

## 4 Experiments

We conducted an experiment to evaluate our imitation learning architecture that uses our proposed motion segmentation method, a sticky HDP-HMM for modeling unsegmented human motion, and our MDL-based chunking method. In these experiments, recorded high dimensional time series data representing unsegmented human upper body motions were input into the learning architecture as learning data samples.

### 4.1 Motion capture and reduction of dimensionality

Human upper body motion, which includes a DOF of 36, was recorded using the motion capture system Gypsy 5 Torso (Meta motion). Each joint angle of a human's upper body, including neck, shoulders, collar bone, elbows, wrists, head, and hips, was measured. A participant was required to manipulate three target objects. When the participant manipulated an object, he/she was required to exhibit a corresponding unique unit motion. Figure 6 shows images of the motion capture system.

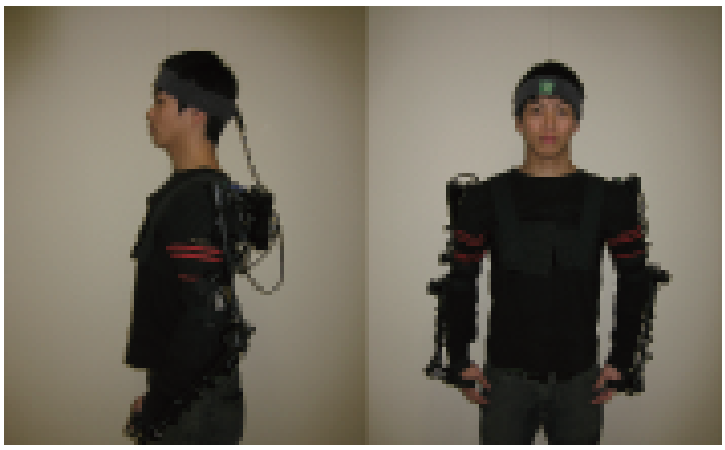

Figure 6: Motion capture system

In these experiments, we used captured human motion data as a learning data set. We asked a participant to move for 20 seconds while being recorded. The frame rate was 60 Hz. During the recording session, we asked the participant to manipulate three types of target objects, a toy, ball, and stick. The participant was asked to arbitrarily switch target objects, and the continuous motion was recorded as an unsegmented motion. The participant was allowed to insert small arbitrary motions between two specific unit motions. Therefore, the time series contained the three types of bodily motions without any explicit segmentation. Figure 7 shows the unit motions embedded in the measured unsegmented human motion.

Here, $X = [x(1), x(2), \cdots, x(T)]$ is the recoded motion data, where $x(t),\ t = 1, 2, \cdots, T$ is a $D$ dimensional column vector measured at time $t$, and $T$ is the length of the time series. The high dimensional human motion data measured using the motion capture system is computationally costly to model due to its complexity. However, human motion potentially has a low dimensional sub space because of motor coordination or motion synergy. This means a characteristic unit human motion is embedded in a low dimensional sub space. Singular value decomposition (SVD) is often used to reduce the dimensionality of human motion and extract a unit human motion embedded in a low dimensional sub space [44]. To reduce the computational cost and to cut meaningless dimensions, we reduced the dimensionality by using SVD. Singular value decomposition provides the most efficient sub vector space, which approximates a given high dimensional time series from the viewpoint of the square error. Every joint angle is assumed to depend on each other linearly for SVD. It might be pointed out that the assumption of linearity is too strong. If the system has nonlinearity, we can use the Gaussian process latent variable model [45], kernel principle component analysis [46], or other dimension reduction methods. However, it is known that human bodily motion has strong linearity among its joint coordinates. This is called the whole body kinematic coordination [47]. Therefore, we simply apply SVD to reduce the dimensionality of human bodily motion.

Singular value docomposition decomposes $X$ to $X = U\Sigma V^T$, where $\Sigma$ is a $D \times T$ matrix ( $\Sigma =$

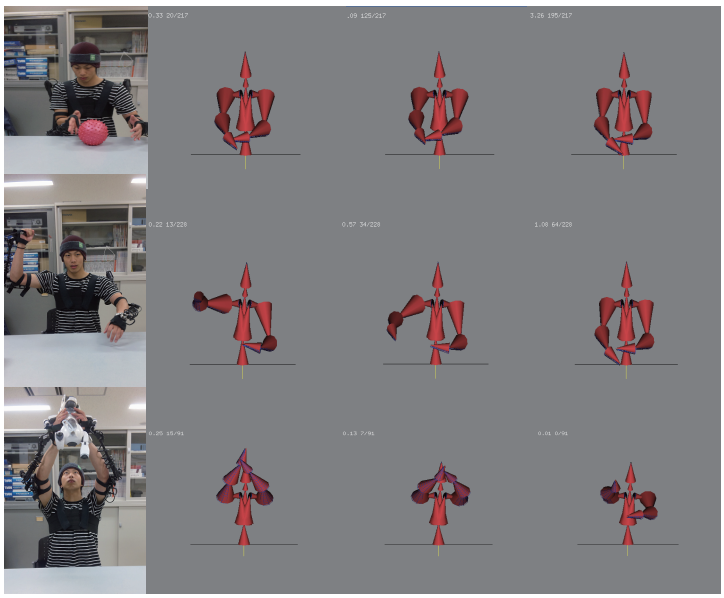

Figure 7: Target objects and their unit motions

$[\omega_{ij}], \omega_{ij} = \sigma_i \delta_{ij} (1 \leq i \leq D, 1 \leq j \leq T))$, $U$ is a $D \times D$ orthogonal matrix, $V$ is a $T \times T$ orthogonal matrix, and $T$ is transposition.

When $\Sigma$ is substituted with $\Sigma_{(K)}$ =diag$(\sigma_1, \sigma_2, \cdots, \sigma_K)$ $(K < D)$ and $U, V^T$ is restricted to he first $K$ columns and $K$ rows, we can obtain $U_{(K)}, V_{(K)}$ as follows.

$$X_{(K)} = U_{(K)} \Sigma_{(K)} V_{(K)}^T \tag{17}$$

This matrix $X$ becomes the best approximation of $X$ with $rank$ $K$ from the viewpoint of the Frobenius norm. This means that this provides the best approximation with $K$ dimensional vector space minimizing the sum of square prediction errors from the viewpoint of time series modeling. In this experiment, we reduced dimensionality to a dimension with an 80% cumulative proportion. As a result, the dimensions were reduced to 6.

## 4.2 Experimental Conditions

We set the sticky HDP-HMM parameters $\alpha = 0.1, \gamma = 0.1, and \kappa = 0.9$ as default values. Before the sticky HDP-HMM learning phase, the Gaussian prior distribution's mean value and variance were calculated and set to $\mu_0 = 0$, and $\Sigma_0 = d_s I$, respectively, which are the hyperparameters of the prior distribution of the mean vectors of each emission distribution. We set the DOF as 9 = (dimension of output space) +3. $\Delta = d_f I$ for the inverse-Wishert distribution, which is a prior distribution of the variance-covariance matrices of the Gaussian distribution. We set the hyperparameter of emission distribution $d_s = 1.0 \times 10^{-3}$ and $d_f = 1.0 \times 10^{-4}$ by referring to the variance of data in state space. We evaluated the effect of these hyperparameters afterward. In the chunking method, we used $\beta = 1.0$ as an inverse temperature parameter. We assumed $r_{tau} = 0.5$ and $\tau_{min} = 1.0$ as parameters for keyword

extraction.

### 4.3 Results

We show the results of motion modeling using the sticky HDP-HMM. Five samples of the learning process are shown in Figure 8. A Gibbs sampler is a stochastic learning algorithm. Therefore, every trial gave a different result probabilistically. Figure 8(b) shows five samples of log likelihood transitions. The log likelihood gradually increased through Gibbs sampling. Figure 8(a) shows the transition of the number of hidden states. Initially, the number of hidden states was set to the maximal value. The number of hidden states gradually changed and almost converged to around 15. To see the distribution of the number of hidden states, we executed the same experiment 30 times. The estimated number of hidden states is shown in Figure 9.

As we can see in Fig. 8(b), the evaluation criterion increased with oscillation not monotonously. This is due to the property of Gibbs sampling. Gibbs sampling is not a learning method that monotonously increases a target evaluation criterion, but one which generates sample output from a target distribution. The initial sample was located in a distant from the peak of probabilistic distribution of the parameter space, and samples which have high posterior probability were gradually obtained.

After ten blocked Gibbs samplings of hidden states and model parameters, a sequence of inference hidden states was obtained for each time series. The sequence was obtained as a *document* written in letters representing hidden states, like a document written in an unknown language. The proposed chunking method was applied to the data. The Gibbs sampling procedure stochastically searches for efficient segmentation and a dictionary for minimizing the two-part description length $L^{double}$ of the target corpus (Fig. 10).

Finally, four keywords were obtained. For example, $[3\ 11] \times 5, [10\ 7\ 14\ 10\ 8\ 2\ 8] \times 6, [10\ 7\ 14\ 10\ 8] \times 2$, and $[16\ 6] \times 3$ were obtained. The segments corresponding to the extracted unit motions are shown in Fig. 11. We confirmed that each extracted phrase corresponds to each unit motion exhibited for each object. However, two phrases corresponding to the motion of "toy" were extracted. [10 7 14 10 8] which is a substring of [10 7 14 10 8 2 8] was also extracted as a unit motion. The both of them corresponded to "lifting up toy" motion. This showed that our unsupervised motion segmentation method sometimes can encode a type of unit motions to different sequences of labels and recognize them as different types of unit motions. These results show that our imitation learning architecture can extract unit human motions from unsegmented human motion data.

For further investigation, we compared the extracted segments by our proposed method with segments which human participants manually extracted. We asked five participants to segment manually the unsegmented human motion data which used in this experiment.

First of all, we examine precision and recall of our segmentation method. Here, we consider that a motion is extracted meaningfully when an automatically extracted segment shares a certain length of

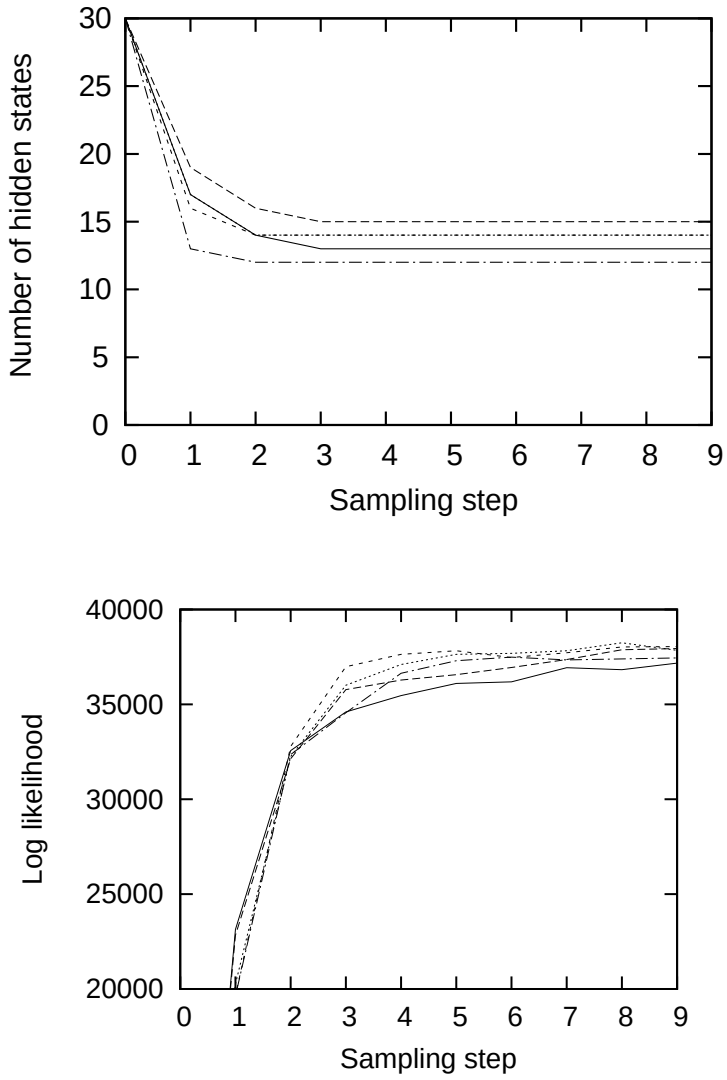


Figure 8: Top: (a) transition of number of hidden states, bottom: (b) Log posterior probability for estimation of hidden states ($d = 1.0 \times 10^{-4}$)

interval with a manually extracted one. Based on this criterion, precision and recall of our segmentation method is shown in Figure 12(left). This shows that our proposed method could extract the most of all segments which are considered as meaningful segments by human participants.

Secondly, we evaluated how the automatically extracted segments were similar to manually ones, namely, segments which are extracted by human participants. Figure 12(right) shows how the automatically and manually extracted segments share the same intervals. A concordance rate $r_c$ is defined as followings. If there are two segments whose intervals are $[t^i_{start}, t^i_{end}]$ and $[t^j_{start}, t^j_{end}]$, their concordance rate $r_c$ is defined as $r_c = 2 \times \frac{\min(t^i_{end}, t^j_{end}) - \max(t^i_{start}, t^j_{start})}{t^i_{end} - t^i_{start} + t^j_{end} - t^j_{start}}$. *Human average* means a rate of participants shares the same segment with the segment averaged over all participants. *Human worst case* means a rate of participants shares the same segment with another person who outputs the most dissimilar

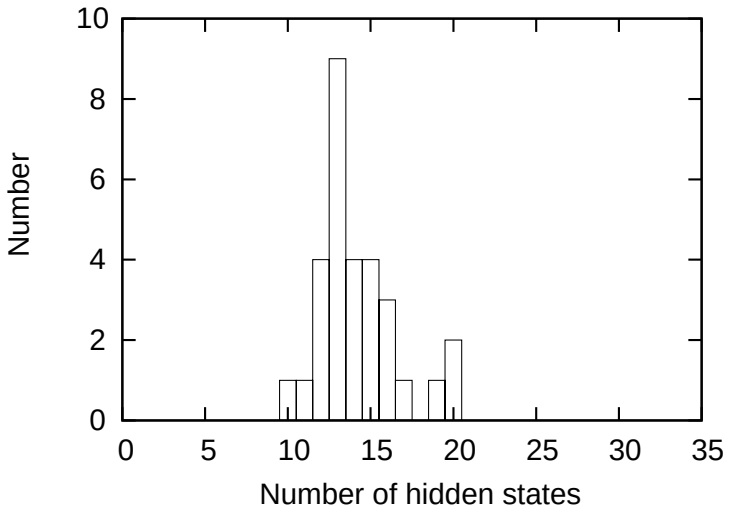


Figure 9: Estimated number of hidden states

segments to the participant's segmentation. The concordance rates averaged over all participants were shown in Figure 12 for all target objects. This figure shows that relatively good results were obtained for *toy* and *stick*. However, seemingly insufficient result was obtained for *playing with ball* motion. We found that the unit motion which was extracted as [16 6] was terminating motion of playing with ball by observing the automatically extracted segment. The result is understandable considering the assumption of our method. This method is proposed based on the notion of double articulation. We assumed that a unit motion has left-to-right structure. In other words, a unit motion can be represented by a certain sequence of hidden states. However, the motion, playing with ball, contained variable number of iterative elemental motions. A person who exhibited the motion pushed the ball left and right for several times. This kind of motion cannot be represented by a single left-to-right HMM. In other words, it cannot be represented by a sequence of letters corresponding to hidden states. Therefore, terminating motion which has left-to-right structure was extracted as a unit motion corresponding to the motion of playing with ball.

As a result, our proposed method could extract the human unit motion which potentially have left-to-right structure. Therefore, our segmentation method can be used in imitation learning in which left-to-right HMM models a unit motion. In order to develop a further unsupervised segmentation method, to develop a mehtod which can extract iterative unit motion. This is our future work.

# 5 Discussion

## 5.1 Dependency on Hyperparameters of Inverse-Wishert Distribution

These results do not suggest that the sticky HDP-HMM requires hand-coded parameter settings. The hierarchical Bayesian model requires settings of hyperparameters. We can estimate hyperparameters

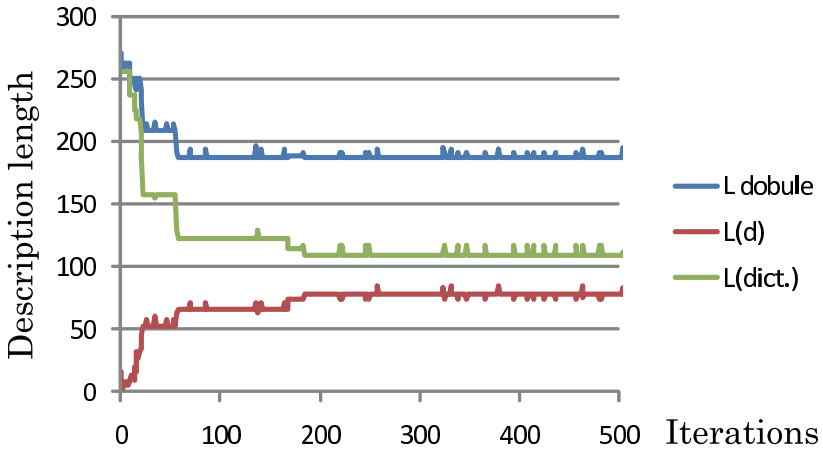


Figure 10: Description length transition of dictionary and target document

by putting prior distributions to them. However, the prior distribution also has hyperparameters. Therefore, we cannot completely omit them. In this paper, we do not estimate hyperparameters, but in this subsection, we examine their effect. We first changed the hyperparameters $\Sigma_0 = d_s I$ for Gaussian distribution and $\Delta = d_f I$ of the inverse-Wishert distribution, which is a prior distribution of the variance-covariance matrix of the Gaussian distribution. We drastically changed the hyperparameters to determine their effect on the number of hidden states. The number of hidden states averaged over three trials for each setting of hyperparameter is shown in Figure 13. Figure 13 shows the results in a double-logarithmic graph. The results suggest smaller hyperparameters result in a larger number of hidden states. This means that an emission distribution with larger variance can cover a broader area of the observed distributed data. Therefore, a smaller number of Gaussian distributions is required to cover the data. Therefore, the variance of each emission distribution is loosely affected by its hyperparameter, and the number of hidden states is determined with the sticky HDP-HMM. In this approach, the number of hidden states is determined in a bottom-up manner.

However, considering that the amplitude of the target low-dimensional time series was about $1.0 \times 10^{-2}$, natural candidates of hyperparameters are about $d_s = d_f = 1.0 \times 10^{-4}$.

Identifying the complexity of contained unit human motion is difficult within the context of imitation learning of segmented human motion. However, identifying the preciseness of modeling is comparatively easier. The order of variance of input data is at least easy to evaluate. Figure 13 shows that the effect of hyperparameters is not as big under the condition that the order of $d_f$ and $d_s$ can be evaluated. Therefore, the sticky HDP-HMM is a suitable modeling method for such imitation learning because it does not require designating the number of hidden states of the HMM beforehand.

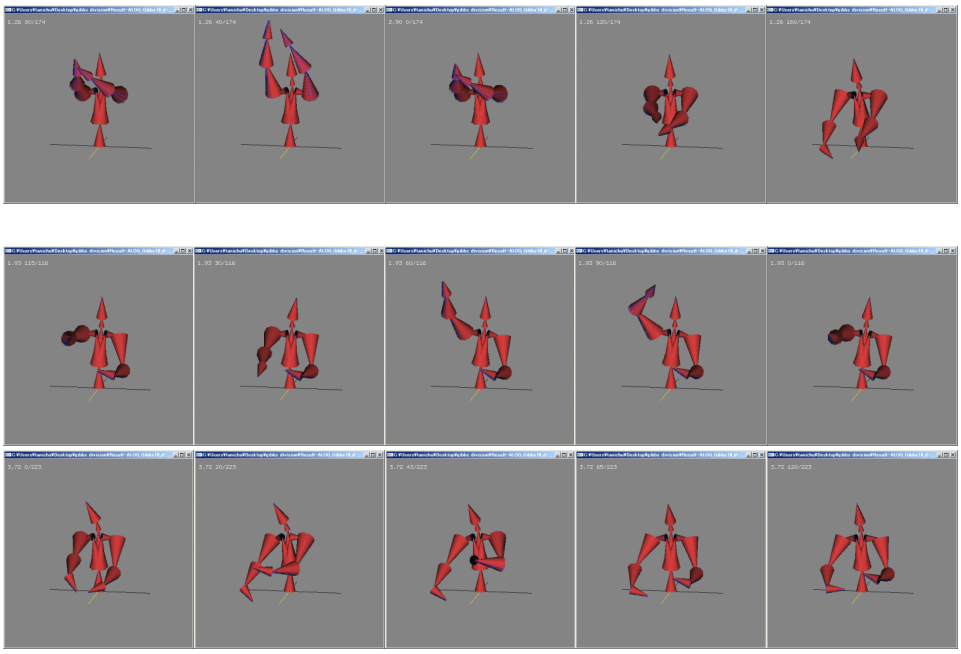

Figure 11: Extracted unit human motions, Top: lifting up toy, corresponding to [10 7 14 10 8 2 8], middle: swinging stick, corresponding to [3 11], bottom: playing with ball, corresponding to [16 6]

## 5.2 Dependency on concentration parameters, $\alpha$ and $\gamma$

The HDP's hyper-parameters $\alpha$ and $\gamma$, which are called concentration parameters, will also affect the number of hidden states in a top-down manner. Figure 14 shows the experimental results of the relationship between hyperparameters and the estimated number of hidden states. The number of hidden states averaged over three trials for each hyperparameter setting is shown in Figure 14.This figure shows that $\alpha$ and $\gamma$ did not have any significant effect on the number of hidden states. This looks strange at first glance from the viewpoint of the original meaning of the concentration parameter in the DP. However, this result is not strange because of $L$ weak-limit approximation adopted in the sticky HDP-HMM's blocked Gibbs sampler algorithm. This weak limit approximation makes the Dirichlet process qualitatively similar to the finite Dirichlet distribution. Therefore, the effect of $\alpha$ and $\gamma$ do not directly affect the generation of a new hidden state. This clearly comes from the equations including $\alpha$ and $\gamma$ in the algorithm described in Section 3. In other words, the sticky HDP-HMM with a blocked Gibbs sampler is not sensitive to the settings of concentration parameters. In other words, the concentration parameter $\alpha$ and $\gamma$ cannot control the number of hidden states.

## 5.3 Comparison between HMM and sticky HDP-HMM

Next, we compared the learning results of the sticky HDP-HMM with those of conventional HMMs using the Baum-Welch algorithm for parameter estimation (ML-HMM)[48] and Bayesian HMMs using

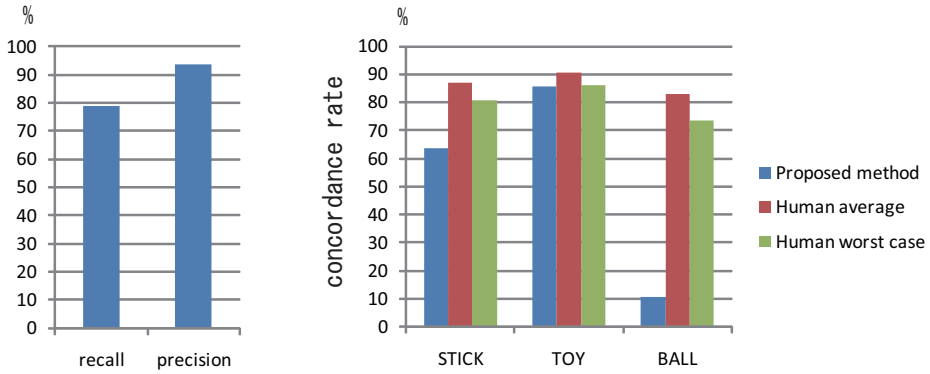


Figure 12: (Left) precision and recall for unit motion extraction. and (right) comparison among correspondence rates of segments extracted by proposed method and those of manually extracted segments.

the blocked Gibbs sampler (Bayes-HMM)[27]. These models are difficult to compare because the sticky HDP-HMM and HMMs using the Baum-Welch algorithm have different evaluation criteria. In this subsection, we compare the three schemes based on log likelihood and show the results in Figure 15.

The Baum-Welch algorithm in a conventional HMM sometimes becomes unstable if its variance-covariance matrix approaches a zero matrix. Therefore, the regularization parameter $\epsilon I$ is added to estimate the variance-covariance matrix, where $I$ is an identity matrix and $\epsilon = 1.0 \times 10^{-4}$ is a very small positive value. $\epsilon$ and $d_f$ had almost the same effect on the learning results of the ML-HMM and Bayes-HMM, respectively.

The results are shown in Fig. 15. The log likelihood averaged over three trials for each number of hidden states for the ML-HMM and Bayes-HMM are shown in Fig. 15.Ten iterations were executed for each condition. In contrast, we could not predict the number of hidden states of the sticky HDP-HMM before the learning process. Therefore, we executed the experiment 30 times using sticky HDP-HMM and its log likelihood of hidden states averaged over the 30 trials and their average are plotted in Figure 15.

In this comparison, we manually changed the number of hidden states for the ML-HMM and Bayes-HMM.

The sticky HDP-HMM's log likelihood was highest after ten iterations among the three models. However, this result is a little strange because the Baum-Welch algorithm in conventional HMMs maximizes the likelihood, which is in contrast to the Gibbs sampler in the sticky HDP-HMM which maximizes posterior probability. There are two reasons for this phenomenon. One is the effect of the regularization term. In this simulation experiment, the regularization term $\epsilon I$ was introduced to prevent problems in calculating the inverse matrix of the variance-covariance matrix of each Gaussian distribution. However, this regularization biases the estimation of the variance-covariance matrix and reduces log likelihood. The other reason is the effect of local minima. The learning result of conventional HMMs is captured by local minima because the Baum-Welch algorithm is a local search method. In contrast, the Gibbs

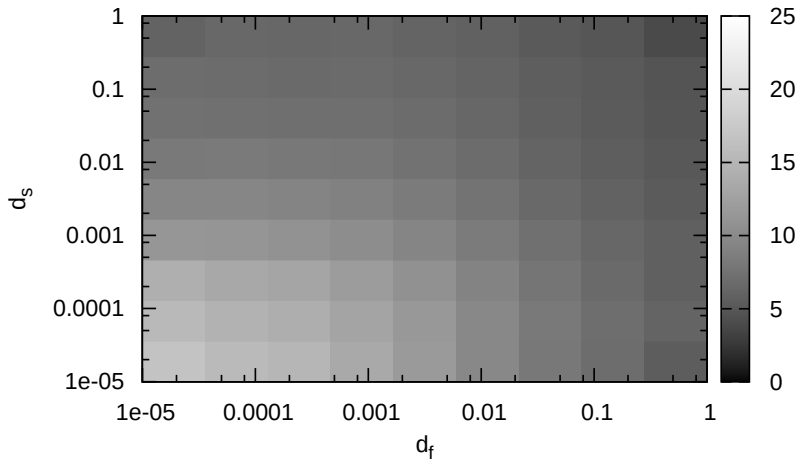


Figure 13: Number of hidden states depending on hyperparameter $d_s$ of Gaussian distribution and hyperparameter $d_f$ of inverse-Wishart distribution

sampler used in the sticky HDP-HMM cannot to be captured by local minima and approaches the global optimal value.

### 5.4 Effect of stickiness $\kappa$

The stickiness hyperparameter $\kappa$ distinguishes the sticky HDP-HMM from the original HDP-HMM. We evaluated the effect of $\kappa$. Figure 16 shows the obtained transition matrix $\pi$ when $\kappa = 0$ and $\kappa = 1.0$, which we used in our experiment. This shows that *kappa* increases self transition and decreases the frequency of transition to other hidden states. This means the number of characters in an encoded document increases without $\kappa$. Therefore, stickiness is obviously important for our proposed architecture.

### 5.5 Interplay between sticky HDP-HMM modeling and MDL-based chunking method

A sticky HDP-HMM models unsegmented motion with a flexible number of Gaussian distributions determined through the MCMC. The number of hidden states affects the following chunking process. If the number of hidden states increases, the modeling error usually decreases. Although the modeling in the bottom level, i.e., elemental motion, becomes more precise, this makes the top level symbolic chunking process more difficult. Figure 17 shows that the initial and final description lengths become longer if the number of hidden states increases. The same data obtained from the experiments (Fig. 9) are used in this figure. The reduction rate in this figure shows that the document's description length can be reduced using the proposed unsupervised chunking method, but the effectiveness depends on

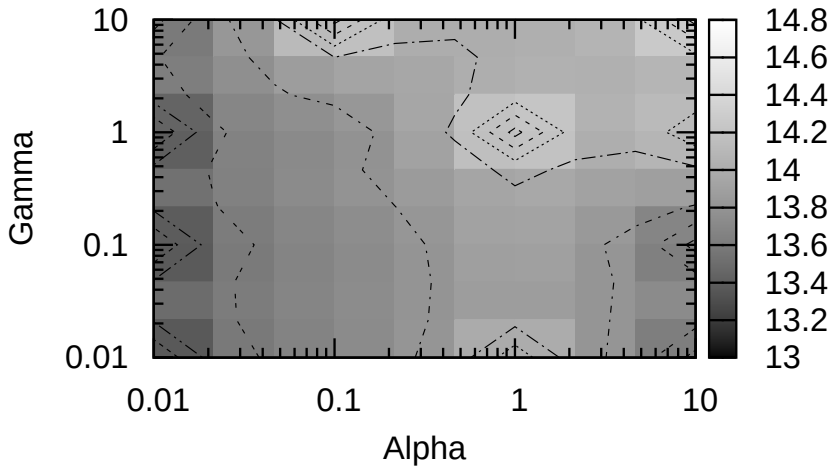


Figure 14: Number of hidden states depending on hyperparameters $\alpha and \gamma$

the number of hidden states. If the number of hidden states increases, a unit motion becomes a longer sequence of letters representing hidden states. It becomes difficult for the same types of unit motions to have exactly the same sequence of hidden states and it becomes difficult to reduce the total description length. This shows that there is a trade-off between bottom-level modeling of motion data and top-level description length of a document. The Dirichlet process can determine the number of hidden states from the viewpoint of posterior probability. However, it is also important to find a good balance between the two layers' criteria. To develop a method for balancing the top- and bottom-layer criteria is for our future work.

# 6 Conclusion

We proposed a motion segmentation method for an imitation learning architecture for unsegmented human motion using a sticky HDP-HMM and unsupervised chunking method that uses a Gibbs sampler based on the MDL principle. The number of hidden states was automatically determined adaptively. The model parameters of the HDP-HMM were not captured by local minima. We also showed that the number of hidden states is also affected by hyperparameters characterizing elemental emission distributions in a bottom-up manner. Our proposed chunking method segmented input documents. We constructed the imitation learning architecture for unsegmented human motion by proposing unsupervised motion segmentation method based on the notion of double articulation.

Our proposed unsupervised motion segmentation method can save a computational cost compared with similar previously proposed methods. Roughly speaking, an iteration of learning process of sticky HDP-HMM requires almost the same computational cost as those of ML-HMM and Bayes-HMM. How-

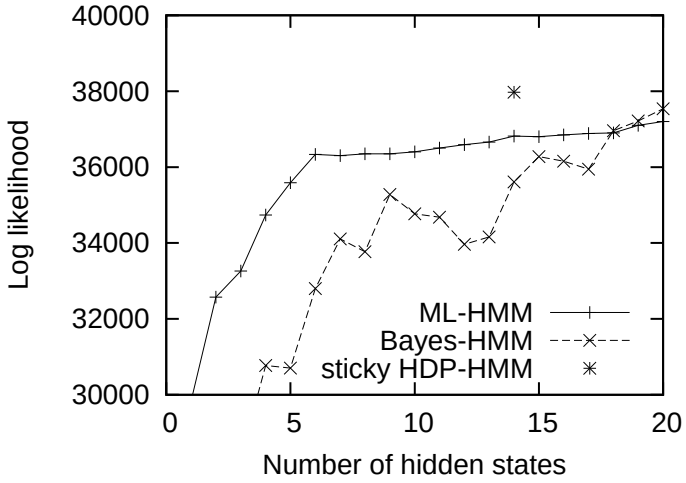


Figure 15: Comparison of log-likelihood of sticky HDP-HMM and HMM. Bayes-HMM means HMM using Gibbs sampler for learning algorithm. ML-HMM uses Baum-Welch algorithm for learning algorithm whose evaluation criteria is maximizing likelihood.

ever, the proper number of hidden states can be estimated with just a little number of trials by using the sticky HDP-HMM. In contrast, ML-HMM and Bayes-HMM require so many trials to estimate the proper number of hidden states. This leads to the huge computational cost. The use of sticky HDP-HMM reduces the computational cost of motion segmentation from the viewpoint of model selection. However, our proposed method cannot work for real-time motion segmentation. In order to achieve a real-time motion segmentation and extraction of unit motions, we have to more efficient learning algorithm for HDP-HMM and unsupervised chunking method.

However, the entire learning process is still separated into two smaller learning processes, i.e., motion modeling with sticky HDP-HMM and chunking method. To optimize the entire learning architecture, integration of the two learning processes is important for formalizing the entire unsegmented imitation learning process as a single generative model.

Several researchers have already proposed an unsupervised morphological analysis method for segmenting written documents by using nonparametric Bayesian language models [49, 50]. Combining these two processes based on the framework of nonparametric Bayesian approach is one of the most promising approaches for autonomous robots to learning by imitating unsegmented human motion. This is for our future work.

## Acknowledgement

This research is supported by a Grant-in-Aid Creative Scientific Research 2007-2011 (19GS0208) funded by the Ministry of Education, Culture, Sports, Science and Technology, Japan.

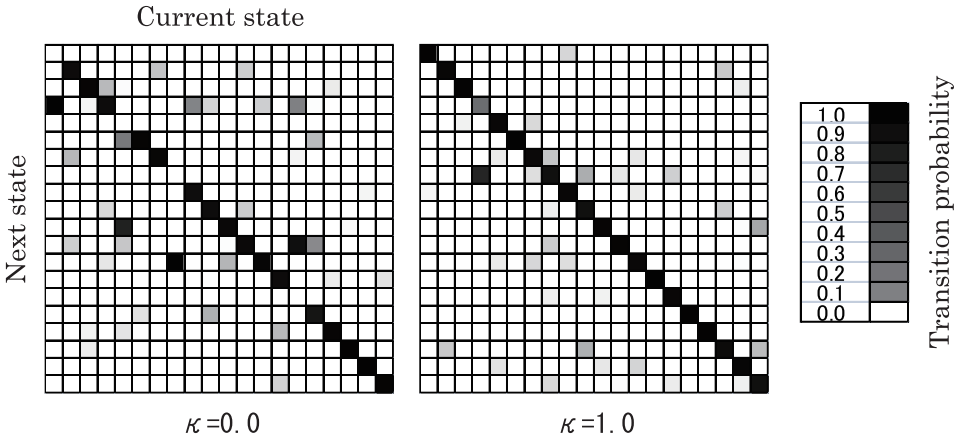


Figure 16: Transition probability depending on stickiness parameter $k$ (left) $\kappa = 0$, and (right) $\kappa = 1.0$

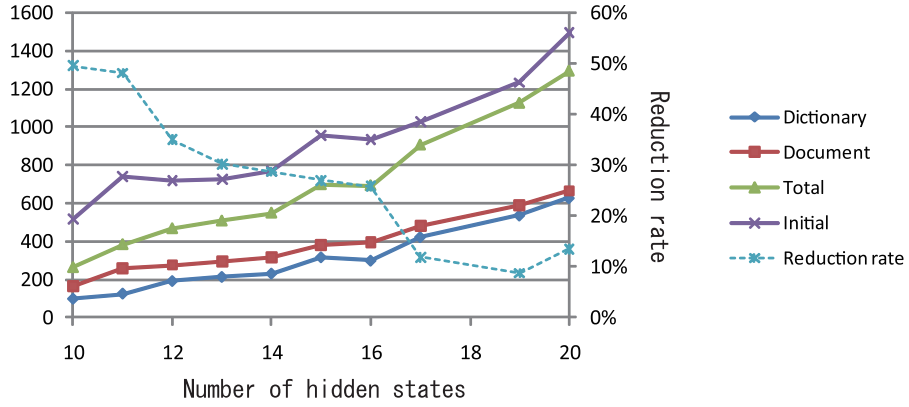


Figure 17: Initial and minimal description lengths obtained in chunking experiment. Reduction rate = 1− minimal description length / initial description length.

# REFERENCES

[1] Y. Kuniyosh, M. Inaba, and H. Inoue. Learning by watching: extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 6, pp. 799–822, 1994.

[2] V. Krüger, D. Kragic, A. Ude, and C. Geib. The meaning of action: a review on action recognition and mapping. *Advanced Robotics*, Vol. 21, No. 13, pp. 1473–1501, 2007.

[3] S. Schaal, A. Ijspeert, and A. Billard. Computational approaches to motor learning by imitation. *Philosophical Transaction of the Royal Society of London: Series B, Biological Sciences*, Vol. 1431, pp. 537–547, 2003.

[4] A. Alissandrakis, C.L. Nehaniv, and K. Dautenhahn. Imitation with alice: Learning to imitate corresponding actions across dissimilar embodiments. *Systems, Man and Cybernetics, Part A:*

*Systems and Humans, IEEE Transactions on*, Vol. 32, No. 4, pp. 482–496, 2002.

[5] C.L. Nehaniv and K. Dautenhahn. The correspondence problem. *Imitation in Animals and Artifacts*, pp. 41–61, 2002.

[6] A. Billard, Y. Epars, S. Calinon, S. Schaal, and G. Cheng. Discovering optimal imitation strategies. *Robotics and Autonomous Systems*, Vol. 47, No. 2-3, pp. 69–77, 2004.

[7] Komei Sugiura, Naoto Iwahashi, Hideki Kashioka, and Satoshi Nakamura. Learning, generation, and recognition of motions by reference-point-dependent probabilistic models. *Advanced Robotics*, Vol. 25, No. 6-7, pp. 825–848, 2011.

[8] J.M. Rubin and Richards. Boundaries of visual motion. *A.I. Memo 835, Massachusetts Institute of Technology, Artificial Intelligence Laboratory*, 1985.

[9] 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.

[10] H. Kawashima and T. Matsuyama. Multiphase learning for an interval-based hybrid dynamical system. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, Vol. E88-A, No. 11, pp. 3022–3035, 2005.

[11] K. Samejima, K Katagiri, K. Doya, and M. Kawato. Symbolization and imitation learning of motion sequence using competitive modules. *IEICE Transactions on Information and Systems*, Vol. 85, No. 1, pp. 90–100, 2002.

[12] J. Barbič, A. Safonova, J.Y. Pan, C. Faloutsos, J.K. Hodgins, and N.S. Pollard. Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface 2004*, pp. 185–194, London, Ontario, Canada, 2004.

[13] Y. Li, T. Wang, and H.Y. Shum. Motion texture: a two-level statistical model for character motion synthesis. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 465–472, 2002.

[14] J. Kohlmorgen and S. Lemm. A dynamic hmm for on-line segmentation of sequential data. *Advances in neural information processing systems*, Vol. 1, pp. 793–800, 2002.

[15] T. Ogata, S. Matsumoto, J. Tani, K. Komatani, and H.G. Okuno. Human-robot cooperation using quasi-symbols generated by rnnpb model. In *Robotics and Automation, 2007 IEEE International Conference on*, pp. 2156–2161, Roma, Italy, 2007. IEEE.

[16] R. Yokoya, T. Ogata, J. Tani, K. Komatani, and H.G. Okuno. Experience Based Imitation Using RNNPB. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 3669–3674, Beijing, China, 2006.

[17] J. Tani, M. Ito, and Y. Sugita. Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using RNNPB. *Neural Networks*, Vol. 17, No. 8-9, pp. 1273–1289, 2004.

[18] W. Hinoshita, H. Arie, J. Tani, H.G. Okuno, and T. Ogata. Emergence of hierarchical structure mirroring linguistic composition in a recurrent neural network. *Neural Networks*, Vol. 24, pp. 311–320, 2011.

[19] J. Tani and S. Nol. Learning to perceive the world as articulated: An approach for. *Neural Networks*, Vol. 12, pp. 1131–1141, 1999.

[20] M. Okada, D. Nakamura, and Y. Nakamura. Selforganizing Symbol Acquisition and Motion Generation based on Dynamics-based Information Processing System. In *Proc. of the second International Workshop on Man-Machine Symbiotic Systems*, pp. 219–229, Kyoto, Japan, 2004.

[21] Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models. In *Advances in neural information processing systems 18*, pp. 1441–1448, Vancouver, British Columbia, Canada, 2006. MIT Press.

[22] Hideki Kadone and Yoshihiko Nakamura. Segmentation, memorization, recognition and abstraction of humanoid motions based on correlations and associative memory. In *IEEE-RAS International Conference on Humanoid Robotics*, pp. 1–6, Genova, Italy, 2006.

[23] Silvia Chiappa and Jan Peters. Movement extraction by detecting dynamics switches and repetitions. In *Advances in Neural Information Processing Systems 23*, pp. 388–396. Vancouver, British Columbia, Canada, 2010.

[24] 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. Technical Report 2777, MIT Laboratory for Information and Decision Systems, 2007.

[25] T. Taniguchi, N. Iwahashi, K. Sugiura, and T. Sawaragi. Constructive approach to role-reversal imitation through unsegmented interactions. *Journal ref: Journal of Robotics and Mechatronics*, Vol. 20, No. 4, pp. 567–577, 2008.

[26] K.P. Murphy. Switching Kalman filters. *Technical reports, DEC/ Compaq Cambridge Research Labs*, 1998.

[27] S.L. Scott. Bayesian methods for hidden markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, Vol. 97, No. 457, pp. 337–351, 2002.

[28] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura. Embodied symbol emergence based on mimesis theory. *International Journal of Robotics Research*, Vol. 23, No. 4, pp. 363–377, 2004.

[29] W. Takano, H. Imagawa, D. Kulic, and Y. Nakamura. What do you expect from a robot that tells your future? the crystal ball. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1780–1785, Taipei, Taiwan, 2010.

[30] Yasuyuki Kihara and Yasuyoshi Yokokohji. Skill transfer from human to robot by direct teaching and task sharing -a case study with origami folding task-. In *11th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems (IFACHMS 2010)*, Valenciennnes, France, 2010.

[31] K.A. Heller, Y.W. Teh, D. Görür, and G. Unit. Infinite hierarchical hidden markov models. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Vol. 5, Florida, USA, 2009.

[32] T.S. Ferguson. A Bayesian analysis of some nonparametric problems. *The annals of statistics*, Vol. 1, No. 2, pp. 209–230, 1973.

[33] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, Vol. 4, No. 2, pp. 639–650, 1994.

[34] J. Pitman. Poisson-dirichlet and gem invariant distributions for split-and-merge transformations of an interval partition. *Combinatorics, Probability and Computing*, Vol. 11, pp. 501–514, 2002.

[35] D. Blackwell and J.B. MacQueen. Ferguson distributions via Pólya urn schemes. *The annals of statistics*, Vol. 1, No. 2, pp. 353–355, 1973.

[36] 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.

[37] Y.W. Teh and M.I. Jordan. Hierarchical bayesian nonparametric models with applications. *Bayesian Nonparametrics*, Vol. 28, , 2009.

[38] M.J. Beal, Z. Ghahramani, and C.E. Rasmussen. The infinite hidden Markov model. *Advances in Neural Information Processing Systems*, Vol. 1, pp. 577–584, 2002.

[39] J. Van Gael, Y. Saatci, Y.W. Teh, and Z. Ghahramani. Beam sampling for the infinite hidden markov model. In *Proceedings of the 25th international conference on Machine learning*, pp. 1088–1095, Helsinki, Finland, 2008.

[40] T. Taniguchi and N. Iwahashi. Computational model of role reversal imitation through continuous human-robot interaction. In *Proceedings of the 2007 workshop on Multimodal interfaces in semantic interaction*, pp. 25–31, Nagoya, Aichi, Japan, 2007.

[41] Kyoji Umemura. Related word-pairs extraction without dictionaries. Technical report, IPA Exploratory Software Project development result report,

http://www.ipa.go.jp/archive/NBP/12nendo/12mito/ mdata/10-36h/10-36h.pdf (in Japanese), 2000.

[42] Y. Tanaka, K. Iwamoto, and K. Uehara. Discovery of Time-Series Motif from Multi-Dimensional Data Based on MDL Principle. *Machine Learning*, Vol. 58, No. 2, pp. 269–300, 2005.

[43] T. Taniguchi and N. Iwahashi. Imitation learning from unsegmented human motion based on n-gram statistics of linear prediction models. *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics*, Vol. 21, No. 6, pp. 1143–1154, 2009.

[44] T. Taniguchi and N. Iwahashi. Imitation learning from unsegmented human motion using switching autoregressive model and singular vector decomposition. In *Advances in Neuro-Information Processing*, No. 5506, pp. 953–961. Springer, 2009.

[45] N.D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in neural information processing systems 16*, Vol. 16, pp. 329–336. The MIT Press, 2004.

[46] B. Schölkopf, A. Smola, and K.R. Müller. Kernel principal component analysis. *Artificial Neural Networks (ICANN'97)*, pp. 583–588, 1997.

[47] T. Funato, S. Aoi, H. Oshima, and K. Tsuchiya. Variant and invariant patterns embedded in human locomotion through whole body kinematic coordination. *Experimental brain research*, pp. 1–15, 2010.

[48] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[49] D. Mochihashi, T. Yamada, and N. Ueda. Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*, pp. 100–108, Suntec, Singapore, 2009.

[50] Y.W. Teh. A hierarchical bayesian language model based on pitman-yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 985–992, Sydney, Australia, 2006. Association for Computational Linguistics.