Simultaneous Clustering and Tracking Unknown Number of Objects

Katsuhiko Ishiguro, Takeshi Yamada and Naonori Ueda
NTT Communication Science Laboratories
Kyoto, 619-0237, Japan
{ishiguro, yamada, ueda}@cslab.kecl.ntt.co.jp

Abstract

In this paper, we present a novel on-line probabilistic generative model that simultaneously deals with both the clustering and the tracking of an unknown number of moving objects. The proposed model assumes that i) time series data are composed of a time-varying number of objects and that ii) each object is governed by a mixture of an unknown number of different patterns of dynamics. The problem of learning patterns of dynamics is formulated as the clustering of tracked objects based on a nonparametric Bayesian model with conjugate priors, and this clustering in turn improves the tracking. We present a particle filter for posterior estimation of simultaneous clustering and tracking. Through experiments with synthetic and real movie data, we confirmed that the proposed model successfully learned the hidden cluster patterns and obtained better tracking results than conventional models without clustering.

1. Introduction

Tracking multiple objects in movie images is an important task [6, 8]. Tracking algorithms compute filtered smooth trajectories of objects from scenes, by fitting state-space dynamics models such as Kalman filters to the observations. Tracking anonymous data such as maneuver tracking on radar (i.e., we cannot directly distinguish objects from observations) [7, 11] is challenging and is mainly formalized in statistical and probabilistic manners and applied to many types of time series data including movie data.

If many target objects are moving in a scene, it is natural to assume multiple different dynamics based on their characteristics. Many existing multi-target tracking models, however, fail to address the problem of estimating multiple patterns of dynamics in tracking problems. They assume that whole trajectories of different objects can be modeled by a single general dynamics model, even though obviously target-wise tuning of dynamics provides better prediction and understanding of time series data.

We consider a multi-target tracking model that allows each target object to use multiple different patterns of dynamics. However, to construct such a model we need to solve the following new problems: i) we do not know the number of patterns in the time series data, and ii) we do not know the specification of each dynamics. The first problem can be understood as time series clustering [9, 10, 12], and the second is the estimation of the model parameters [2, 3]. If we know these patterns of dynamics in advance, then we can apply a conventional tracking algorithm. However, identifying these patterns is difficult without first segregating mixed indistinguishable target objects into target-wise trajectory data with a tracking algorithm. Unfortunately these algorithms assume that time series data are segregated by objects. It is a chicken-and-egg problem.

Figure 1. Example of system input and output. Left figure shows snapshot of a movie given to system as input. Right figure shows output snapshot obtained from the system, where different colored rectangles represent targets' ID and positions and the numbers on the rectangles denote dynamics pattern indices.

This paper proposes a probabilistic generative model that simultaneously enables both clustering patterns of dynamics and the tracking of unknown numbers of objects. We assume that time series data are generated from the hidden trajectories of target objects, each of which is governed by an unknown number of different dynamics parameters. We incorporate the Dirichlet Process Mixture (DPM) to a probabilistic multi-target tracking model to perform clustering and estimate an unknown number of Kalman filter parameters, and this clustering in turn improves tracking accuracies. A Particle filter approach is employed to make on-line (incremental) inferences of latent variables and hid-

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

den states. Experiments using synthetic and real-world data show the effectiveness of the proposed model compared with conventional models.

2. Generative Models

Figure 2 shows the graphical models of those previously studied in the literature and our new proposal. These graphical models represent dependencies between random variables and explain how the data are generated from the process characterized by these variables. In our model, data to be generated include hidden state $\boldsymbol{x}(t)$, which is a real-valued vector consisting of the actual coordinates of the tracked object, and observation $\boldsymbol{y}(t)$, which is a sensed signal that is possibly collapsed by observation noises.

2.1. Conventional Models

First, we describe the simple multi-target tracking model proposed by Särkkä et al. [11] in Fig. 2(a). Since we have multiple targets, the hidden state of the ith target object is denoted as $\boldsymbol{x}_i(t)$. We represent this by depicting a "plate" around $\boldsymbol{x}_i(t)$. Analogously, this model produces multiple observations $\boldsymbol{y}_m(t)$ indexed by m. N_t and M_t are the number of objects and observations at time t, respectively.

When we have multiple target objects (hidden states) and observations, we have to solve the correspondence between targets and observations, since we don't know which target object generates which observation. This problem, which is called data association, is essential to correctly estimate the target-wise hidden state. This model introduces two kinds of latent variables, c and j, to solve the data association problem. $c_i(t)$ represents the birth (addition) and death (deletion) of target object i and thus explains how many target objects appear and disappear. With $c_i(t) = 1$, the ith object is alive (or visible in the scene) at time t, and $c_i(t) = 0$ means the target object is now outside the scene. If a new target object is generated at time t, then new index $\hat{i}$ is produced and $c_{\hat{i}}(t) = 1$. $j_m(t)$ is the data association variable that associates $\boldsymbol{y}_m(t)$ with its source target $\boldsymbol{x}_i(t)$. If $j_m(t) = i$, then ith target object $\boldsymbol{x}_i(t)$ generates mth observation $\boldsymbol{y}_m(t)$. For the inference of such latent variables and hidden states, the Rao-Blackwellized Particle filter is employed. Note that dynamics parameters ξ and ψ are fixed constants (depicted as squares) in this model. This implies that the whole time series data are governed by one fixed dynamics. As discussed in the previous section, this assumption is not always relevant.

The model illustrated in Fig. 2(b) [2] represents the situation where one target object produces its trajectory and emits observations while occasionally changing its dynamics. In contrast to the model proposed by Särkkä et al., we have a set of parameters $\{\xi_k\}$, $\{\psi_k\}$ indexed by k. $\boldsymbol{x}(t)$ ($\boldsymbol{y}(t)$) is generated from the noise distribution parameterized by ξ_k (ψ_k). At each time step t, $z_t = k$ is sampled as an index of the dynamics patterns, and corresponding dynamics parameters ξ_k and ψ_k are used to generate the hidden states and the observations.

The novelty of their work lies in introducing the Dirichlet Process Mixture (DPM) to model these two noise distributions. DPM is a flexible Bayesian nonparametric model that can be seen as an infinite mixture of distributions. By introducing DPM, the model can infer not only the specification of dynamics parameters but also the number of mixture components. Therefore, the system can track without explicitly specifying the number of mixture components (parameters) and their characteristics. As the inference algorithm the authors present a MCMC technique for off-line inference and a Particle filter approach for on-line inference. One major drawback of this model is that it cannot handle time series data that consist of multiple objects. The time series must be segregated by the target objects in advance because the model cannot solve the data association problem.

To summarize, these two most recent models are not applicable to a time series that consists of the trajectories of multiple target objects that occasionally change their dynamics by switching between a pool of dynamics. In the next subsection, we propose a new model that features the advantages of both models.

2.2. Our Proposed Model

In this subsection, we first outline our proposed model before discussing the details in a later section. Fig. 2(c) is the graphical model of our proposed model that is designed to represent the multi-target tracking data generated from a mixture of dynamics parameters. Following the multi-target tracking model (Fig. 2(a)), the hidden state of object i is denoted as $\boldsymbol{x}_i(t)$ and the mth observation is $\boldsymbol{y}_m(t)$.

To generate $\boldsymbol{x}_i(t)$, first the model generates $c_i(t)$, which denotes the birth (addition) and death (deletion) of target objects as explained earlier. We model the process of target additions and deletions by Bernoulli trials (Eq. (7)). We can produce a time-varying number of object trajectories by controlling $c(t)$. Next $z_i(t)$ is generated as an index of the dynamics that governs the ith target at time t by following Caron et al.'s model (Fig. 2(b)). Sampling $z_i(t)$ (Eq. (5)) is modeled by the Chinese Restaurant Process [1, 5] with concentration parameter γ, combined with a Bernoulli trial parametrized by π. CRP is a realization of DPM suitable for clustering and can model multiple patterns (clusters) of dynamics without fixing the number of dynamics patterns.

For each pattern k, dynamics parameters ξ_k and ψ_k are generated from base distribution G_0 and H_0, respectively. In our model, we use a Kalman filter as the state-space model, and ξ_k and ψ_k are their normal distribution parameters. We assume that G_0 and H_0 are both Normal Inverse

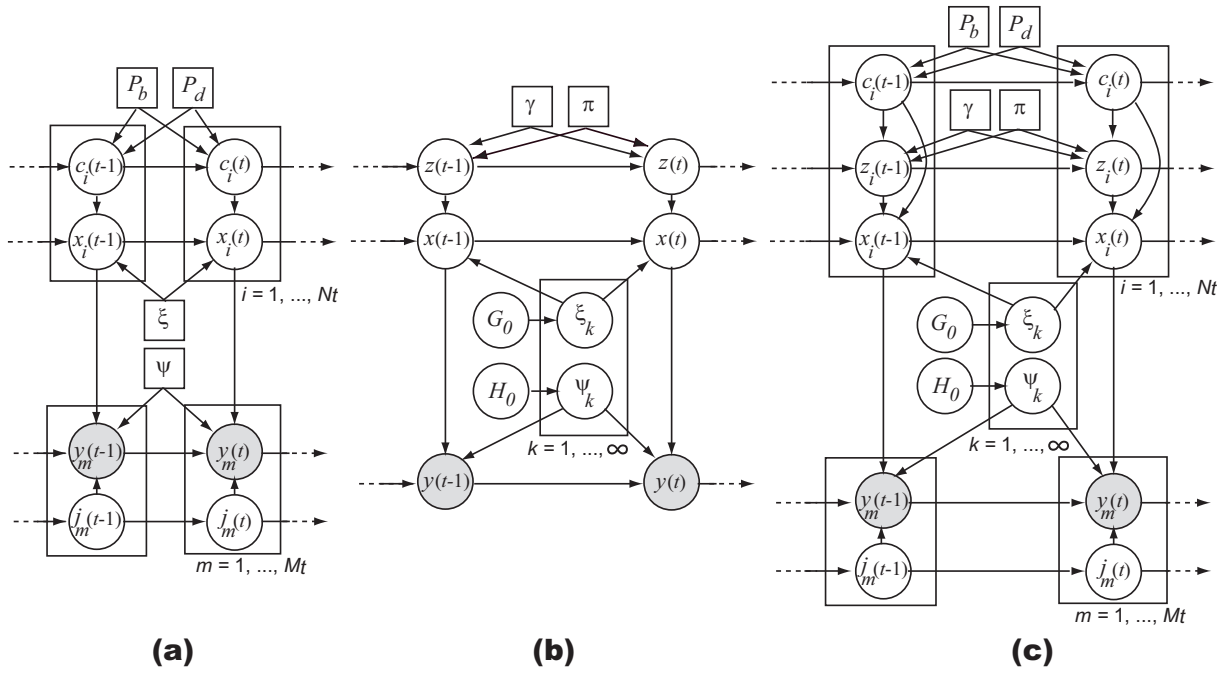

Figure 2. Graphical models of three generative models described in this paper. Circle nodes represent random variables, and squares denote fixed parameters. White nodes are latent variables, and shaded nodes are observables. Rectangles ("plates") denote replication, with the index given in rectangle's bottom. (a): model proposed in [11], (b): model proposed in [2] and (c): proposed model.

Wishart distributions (NIW) with parameters θ^{ξ} and θ^{ψ} respectively. Now we generate each of the latent variables required to produce $\boldsymbol{x}_i(t)$. The hidden state of ith object $\boldsymbol{x}_i(t)$ is produced if the object is in scene ($c_i(t) = 1$), and the generation process is governed by ξ_k that is designated by $z_i(t) = k$.

To generate observation $\boldsymbol{y}_m(t)$, we first produce latent data association variable $j_m(t)$ (Eq. (5)). As stated above $j_m(t)$ holds the association information between $\boldsymbol{x}_i(t)$ and $\boldsymbol{y}_m(t)$. If $j_m(t) = i$ then mth observation $\boldsymbol{y}_m(t)$ is generated from ith target object $\boldsymbol{x}_i(t)$ with target observation parameter ψ_k. Iterating this process for all t, we have time series data consisting of *multiple* target trajectories, each of which is governed by *multiple* different dynamics.

3. Inference

In this section, we discuss the on-line clustering and dynamics parameter learning schemes and how to infer them. Note that we represent the whole set of the variables from time step 1 to t by capitals: *e.g.* $\boldsymbol{X}(t) = \{\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(t)\}$ and $\boldsymbol{Y}(t) = \{\boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(t)\}$.

3.1. Posterior Estimation with Particle Filter

Our objective is to estimate posterior distribution $p(\boldsymbol{x}(t)|\boldsymbol{Y}_t)$. We write the set of latent variables at time step t as $\phi(t) = \{\{c_i(t)\}, \{j_m(t)\}, \{z_i(t)\}, \{\xi_k(t)\}, \{\psi_k(t)\}\}$, and $\Phi(t) = \{\phi(1), \phi(2), \ldots, \phi(t)\}$.

Using these notations, we approximate the posterior $p(\boldsymbol{x}(t)|\boldsymbol{Y}(t))$ with Particle filter technique as follows:

$$p(\boldsymbol{x}(t)|\boldsymbol{Y}(t)) = \sum_{s=1}^{S} p\left(\boldsymbol{x}(t), \Phi(t)^{(s)}|\boldsymbol{Y}(t)\right) w(t)^{(s)}, \quad (1)$$

where S denotes the number of particles. Let $q(\cdot)$ be the proposal distribution that generates $\Phi(t)^{(s)}$ as follows:

$$\phi(t)^{(s)} \sim q(\phi(t)|\Phi(t-1), \boldsymbol{Y}(t)). \quad (2)$$

For the choice of the proposal distribution, we may simply use $p(\phi(t)|\Phi(t-1))$ instead of $q(\phi(t)|\Phi(t-1), \boldsymbol{Y}(t))$. Each particle is weighted by $w(t)^{(s)}$ defined by:

$$w(t)^{(s)} = w(t-1)^{(s)} \frac{p\left(\boldsymbol{y}(t)|\phi(t)^{(s)}\right) p\left(\phi(t)^{(s)}|\Phi(t-1)\right)}{q(\phi(t)|\Phi(t-1), \boldsymbol{Y}(t))}. \quad (3)$$

For likelihood $p\left(\boldsymbol{y}(t)|\phi(t)^{(s)}\right)$, we use Kalman likelihood, as in [11]. Because we model $p(\boldsymbol{x}(t), \Phi(t)|\boldsymbol{Y}(t))$ by Kalman filter following previous researches [7, 11], we can easily compute hidden state distribution $p(\boldsymbol{x}(t), \Phi(t)|\boldsymbol{Y}(t))$ in Eq. (1), after obtaining the latent variable samples.

Based on the generative model (Fig. 2(c)), we decom-

pose conditional distribution $p(\phi(t)|\Phi(t-1))$ as follows:

$$p(\phi(t)|\Phi(t-1)) \triangleq p(c(t)|C(t-1), P_b, P_d) \quad (4)$$

$$\times \prod_m p(j_m(t)|J(t-1)) \prod_i p(z_i(t)|Z(t-1), \gamma, \pi) \quad (5)$$

$$\times \prod_k p\left(\xi_k(t)|\theta_k^{\xi}(t-1)\right) p\left(\psi_k(t)|\theta_k^{\psi}(t-1)\right) \quad (6)$$

3.2. "Birth and Death" Variable

The right term in Eq. (4) is the conditional prior of $c(t)$, which denotes the birth (addition) and death (deletion) of target objects. With $c_i(t) = 1$, the ith object is alive (or in the scene) at time t, and $c_i(t) = 0$ means the target object is now outside the scene. In [11], the time evolution of C is formalized as two-step Bernoulli trials: i) an existing object with $c_i(t-1) = 1$ disappears and $c_i(t) = 0$ with probability P_d, otherwise $c_i(t) = 1$, ii) a new target object is generated in the scene with probability P_b. We limit the number of newly born objects at every time step to 1 (as in [11]), giving us the following:

$$p(c(t)|C(t-1), P_b, P_d) = {P_d}^{n_d}(1-P_d)^{n_s}{P_b}^{n_b}(1-P_b)^{1-n_b}, \quad (7)$$

where n_s denotes the number of surviving target objects, n_d is the number of that disappeared, and the $(n_b = \{0, 1\})$ is the number of newly born objects.

3.3. Data Association Variable

For data association variable J, $p(j_m(t) \mid J(t-1))$ in the first term of Eq. (5) is assumed as uniform among all existing objects. Otherwise, if search space of J is large, we can introduce a pseudo likelihood to the proposal to reject the outliers. In this experiment, we use the following distribution for the proposal of j similar to [11]:

$$q(j_m(t) = i|J(t-1), \boldsymbol{y}_m(t)) \propto p(\boldsymbol{y}_m(t)|\hat{\boldsymbol{x}}_i(t))\, p(j_m(t) = i|J(t-1)). \quad (8)$$

This equation incorporates observation likelihoods given a "representative" state vector $\hat{\boldsymbol{x}}_i(t)$, which is the average of the predictive distribution of $\boldsymbol{x}_i(t)$.

3.4. Dynamics Cluster Index Variable

We adopt the Chinese Restaurant Process (CRP) [1] prior to generate $z(t)$ in the second term of Eq. (5). CRP is a realization of Dirichlet Process Mixture (DPM), which is a family of nonparametric Bayes, and is a distribution over partitions. We have a set of mixture components or clusters indexed by k, and write the cluster index of ith sample as z_i. The distribution over index of ith sample conditioned on the indices $z_{1:i-1}$ is

$$p(z_i = k|z_{1:i-1}) = \begin{cases} \frac{m_k}{i-1+\gamma} & \text{if } k \text{ is an existing cluster with } m_k > 0 \\ \frac{\gamma}{i-1+\gamma} & \text{if } k \text{ is a new cluster} \end{cases}, \quad (9)$$

where m_k denotes the size of the kth cluster. We write this sampling procedure as $z_i \sim CRP(\gamma)$. CPR favors a small number of clusters that can be easily understood from Eq. (9) and is consistent with the intuition of clustering. Estimating the posterior distribution of $z(t)$ can be achieved straightforwardly by combining the prior provided by CRP and an appropriately selected data likelihood.

In our model, Eq. (9) is updated to obtain the best clustering estimate at every time step. Using $Z(t-1)$ as prior information, we write the sampling of $z_i(t)$ as follows:

$$p(z_i(t) = k|Z(t-1), \gamma) = \begin{cases} \frac{m_k(t-1)}{|Z(t-1)|+\gamma} & \text{if } m_k(t-1) > 0 \\ \frac{\gamma}{|Z(t-1)|+\gamma} & \text{if } m_k(t-1) = 0 \end{cases}, \quad (10)$$

where $m_k(t-1)$ is the total population size (the number of i s.t. $z_i(t) = k$) of kth cluster up to the time step $t-1$, and $|Z(t-1)|$ denotes the sum of total population among all the clusters up to the time step $t-1$: i.e., it equals the sum of $m_k(t-1)$ for all k.

In addition, we introduce another parameter π in order to improve performance. We assume that the changes of dynamics happen only occasionally and thus the target object switches its dynamics parameters with relatively small probability π. Now we formulate the second term of Eq. (5) as follows:

$$z_i(t) \sim CRP(Z(t-1), \gamma) \quad \text{with probability} \quad \pi \quad (11)$$

$$z_i(t) = z_i(t-1) \quad \text{with probability} \quad 1-\pi \quad (12)$$

3.5. Kalman Filter Parameters

In our model, we use linear Gaussian state-space models, namely Kalman filters, for the dynamics models. If the hidden state of ith object $\boldsymbol{x}_i(t)$ and mth observation $\boldsymbol{y}_m(t)$ are generated from the kth dynamics cluster (i.e. $j_m(t) = i$ and $z_i(t) = k$), we have the following state-space model equations:

$$\boldsymbol{x}_i(t) = f(\boldsymbol{x}_i(t-1), \xi_k(t)), \quad \xi_k(t) = \{\boldsymbol{q}, \boldsymbol{Q}\} \quad (13)$$

$$\boldsymbol{y}_m(t) = h(\boldsymbol{x}_i(t), \psi_k(t)), \quad \psi_k(t) = \{\boldsymbol{r}, \boldsymbol{R}\}. \quad (14)$$

f and h are standard linear Gaussian models with normal distribution noise. $\xi_k(t) = \{\boldsymbol{q}, \boldsymbol{Q}\}$ is the mean and covariance matrix for the system model noise and $\psi_k(t) = \{\boldsymbol{r}, \boldsymbol{R}\}$ is for observation noise. To estimate these parameters, we introduce Normal Inverse Wishart distribution (NIW) as a

Table 1. Tested methods and their characteristics

method	Eqs. (17), (18)	Eqs. (10), (11), (12)
Single	no	no
Individual	yes	no
Clustered	yes	yes

prior distribution of the parameters (c.f. [2]). We assign NIW for each cluster, for the system noise and the observation noise, respectively. We parameterize the system noise NIW and the observation noise NIW by $\theta_k = \{\theta_k^{\xi}, \theta_k^{\psi}\}$.

$\xi_k(t)$ and $\psi_k(t)$ are sampled from the posterior distribution given the data up to time $t-1$ as current estimates of the true dynamics parameter ξ_k and ψ_k. The sampling distributions and their hyper parameters θ_k are also updated according to the data: $\theta_k(t-1) = \{\theta_k^{\xi}(t-1), \theta_k^{\psi}(t-1)\}$. Components in Eq. (6) will be described as follows:

$$p\left(\xi_k(t)|\theta_k^{\xi}(t-1)\right) = NIW\left(\xi_k(t); \theta_k^{\xi}(t-1)\right) \quad (15)$$

$$p\left(\psi_k(t)|\theta_k^{\psi}(t-1)\right) = NIW\left(\psi_k(t); \theta_k^{\psi}(t-1)\right). \quad (16)$$

Using sampled $\xi_k(t)$ and $\psi_k(t)$, we perform tracking.

For on-line estimation, we update hyper parameters θ to obtain better predictions of ξ and ψ after tracking. These updates can be analytically performed because of the conjugacy:

$$NIW\left(\xi_k; \theta_k^{\xi}(t)\right) \propto p\left(\boldsymbol{x}(t)|\xi_k\right) NIW\left(\xi_k; \theta_k^{\xi}(t-1)\right). \quad (17)$$

$$NIW\left(\psi_k; \theta_k^{\psi}(t)\right) \propto p\left(\boldsymbol{y}(t)|\psi_k, \boldsymbol{x}(t)\right) NIW\left(\psi_k; \theta_k^{\psi}(t-1)\right). \quad (18)$$

Updated hyper parameters $\theta(t)$ will be used in Eqs.(15) and (16) in the next time step.

4. Experiments

We tested the proposed model with both artificially generated data and real movie data.

4.1. Compared Models

We compared the proposed model with the other two more restricted models with fewer latent variables and evaluated their effect. In experiments we compared the three models summarized in Table 1. The first model (labeled as **Single** in Table 1) is a baseline that closely resembles Särkkä et al.'s model [11] in Fig. 2(a). This model does not update the dynamics hyper parameters $\theta^{\xi}(t)$ and $\theta^{\psi}(t)$ in Eqs. (17) and (18), nor select and generate clusters with CRP (Eqs. (10), (11) and (12)): all $\xi(t)$ and $\psi(t)$ are always sampled from single default hyper parameters $\theta(0)$.

The second model (**Individual**) learns hyper parameter $\theta(t)$ incrementally as described in the previous section (Eqs. (17) and (18)), but without clustering on $z_i(t)$. This model represents more complicated time series data than the first model by modeling each target object with its own dynamics parameters: state trajectories of multiple objects are governed by the multiple patterns of dynamics.

Finally, the proposed model (**Clustered**) is tested, which has clustered dynamics and hyper parameter adaptations. Compared to the second model, this model assumes a number of patterns in dynamics shared among multiple objects and also that the dynamics of an object may change temporarily.

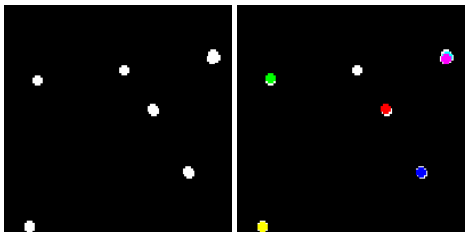

Figure 3. Example of synthetic movie data. Left: snapshot of observations. Each point denotes one observation. Right: ground truth data. Colored points are target objects, and white are noise data.

Table 2. Noise parameters used in generation of synthetic data

	$\boldsymbol{q}$	$\boldsymbol{Q}$	$\boldsymbol{R}$
I	$\{3.0, 0.0\}^T$	diag$\{1.0, 1.0\}$	diag$\{1.0, 3.0\}$
II	$\{-3.0, 0.0\}^T$	diag$\{1.5, 1.5\}$	diag$\{1.0, 2.0\}$
III	$\{0.0, 3.0\}^T$	diag$\{0.5, 0.5\}$	diag$\{2.0, 2.0\}$
IV	$\{0.0, -10.0\}^T$	diag$\{1.0, 1.0\}$	diag$\{0.5, 3.5\}$

4.2. Synthetic Data Experiment

We consider tracking and clustering moving mass points in a $[0:200] \times [0:200]$ virtual 2D space. The hidden states of the target objects are the 2D real-valued vectors representing the coordinates of the objects. The observations are also 2D real-valued vectors and the collapsed coordinates of the mass points. The observations include outputs from a "noise target", which is a false target. Each target object obeys the following random walk model:

$$\boldsymbol{x}_i(t) = \boldsymbol{x}_i(t-1) + \boldsymbol{v}(t), \quad \boldsymbol{v}(t) \sim N\left(\boldsymbol{q}, \boldsymbol{Q}\right) \quad (19)$$

$$\boldsymbol{y}_i(t) = \boldsymbol{x}_i(t) + \boldsymbol{w}(t), \quad \boldsymbol{w}(t) \sim N\left(\boldsymbol{r}, \boldsymbol{R}\right). \quad (20)$$

where $N(\cdot)$ denotes the normal distribution. In this experiment, each object alters its dynamics by changing the noise parameters selected among four patterns, which we present in Table 2. In all patterns, $\boldsymbol{r}$ is the zero vector.

The initial hyper parameters of NIW distribution $\theta(0)$ are selected so that the average of $\xi_k(t)$ and $\psi_k(t)$ takes the following values: $\boldsymbol{q} = \boldsymbol{0}$, $\boldsymbol{Q} = diag\{10.0, 10.0\}$, $\boldsymbol{r} = \boldsymbol{0}$,

Table 3. Time step wise average of data log-likelihood for synthetic data and real movie data

Method	π	synthetic	real 1	real 2
Single	-	-115.22	-95.62	-54.54
Individual	-	-107.85	-84.21	-51.94
Clustered	0.1	-96.68	-79.71	-50.30
	0.2	-96.37	-80.93	-50.74
	0.5	-103.66	-82.96	-51.86
	1.0	-110.31	-83.93	-52.13

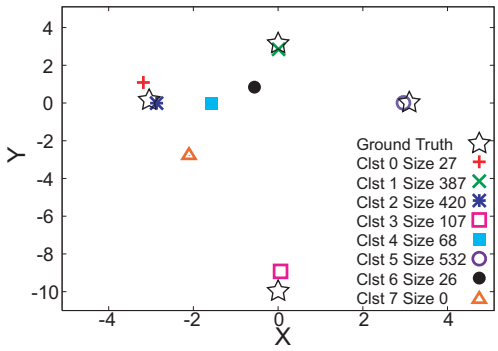

Figure 4. Plot of mean values of system noise distribution

$\boldsymbol{R} = diag\{5.0, 5.0\}$. The length of the time series data is 300 steps. The number of particles is $S = 300$, and the CRP concentration parameter is set to $\gamma = 2$. At every time step, a new object is born with the probability of $P_b = 0.1$. Each existing object ($c_i(t-1) = 1$) is deleted based on Eq. (21) with $\lambda = 0.1$. Let t_n be the last time step when at least one observation $\boldsymbol{y}_m$ exists such that $j_m(\cdot) = i$. Then the probability of the object deletion P_d is defined as follows:

$$P_d = 1 - \lambda e^{t - t_n}. \tag{21}$$

Now we analyze the results. The averaged data likelihood at each time step is shown in the third column of Table 3. This value measures how likely the obtained latent variables (namely, a model) generates the observed data. The table shows that dynamics parameter estimation and its clustering improve the log likelihood. This means our proposed model successfully finds better models and parameters through on-line tracking and clustering.

Next we present the distribution of $\boldsymbol{q}$ in Fig. 4 at the last time step ($\pi = 0.2$). $\boldsymbol{q}$ represents the averaged bias of the target velocities. The analyzed data has four patterns in $\boldsymbol{q}$, as discussed earlier. Each plot corresponds to a dynamics cluster, and the number of "size" is the data size of each cluster: how many times the objects selected this dynamics. Four dominant clusters acquire values nearly equal to the designed ground truth. Namely, our model successfully clustered the dynamics patterns and estimated their parameters.

4.3. Real Movie Data Experiment

Next we describe two experiments using two real movie data. The task was to perform clustering and tracking feature points emitted from pedestrians in movie frames recorded by a common digital camera.

Figure 5. Snapshots from the movie data used in the real movie data experiments. (a): used for the first experiment. (b): used for the second experiment.

The movie data of the first real data experiment was recorded at a shopping mall corridor (Fig. 5 (a)). The camera was placed almost right above the pedestrians. The size of each frame is 320×240 pixels. Basically, pedestrians move in vertical directions (upward or downward), and occasionally make turnarounds to enter shops.

Target feature points were extracted as follows. First, the background subtraction was performed on all frames. Second, pixels were binarized with thresholds to extract the foreground pixels. Then, we extracted the black-colored pixels and performed mean shift clustering on them at each frame. Finally, obtained means were used as the observed feature points. On average, each pedestrian corresponds to one to three feature points at each frame.

In this experiment we utilized a color model [8] for data association prior (Eq. (5)) instead of uniform distribution. We computed an 8-bit RGB histogram around the feature point. Data association prior was computed based on the Bhattacharya coefficient between the histogram around the feature point and the object's histogram obtained at the previous time step.

The state-space model is identical with the previous synthetic data experiment. The initial hyper parameters of NIW distribution $\theta(0)$ are selected so that the average of $\xi_k(t)$ and $\psi_k(t)$ takes the following values: $\boldsymbol{q} = \boldsymbol{0}$, $\boldsymbol{Q} = diag\{20.0, 20.0\}$, $\boldsymbol{r} = \boldsymbol{0}$ and $\boldsymbol{R} = diag\{20.0, 20.0\}$. The length of the time series data is 200 steps. 200 frames are extracted from 1000 frames of 30 FPS-captured movie data. The number of particles is $S = 500$, and CRP concentration parameter is set to $\gamma = 0.1$. The target birth probability $P_b = 0.1$ and target death probability P_d is calculated by Eq. (21) with $\lambda = 0.1$.

The second real movie data is a crowded walkway scene (Fig. 5 (b)). The size of each frame is 640×480 pixels.

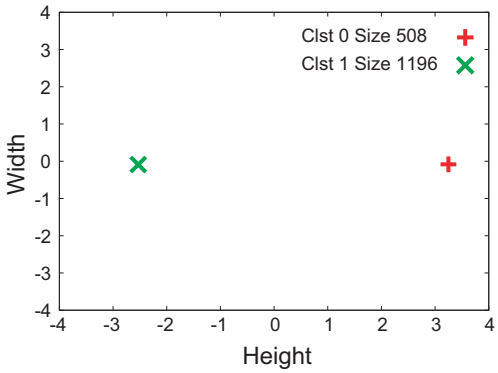

Figure 6. Plot of mean values of system noise distribution at the last frame of the first real movie data experiment

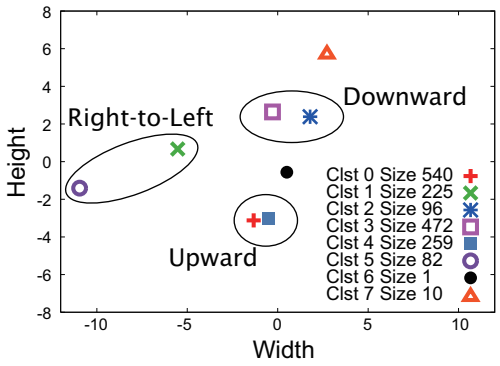

Figure 7. Plot of mean values of system noise distribution at the last frame of the second real movie data experiment

As in Fig. 5 (b), the viewpoint was set in lower angle of elevation, and occlusions were observed frequently. We observe that there are roughly three patterns in the movements of pedestrians: upward, downward and right-to-left movements. Each of three patterns may be divided into subpatterns, reflecting different crash avoidance behaviors in the crowded area.

Target feature points were extracted by employing HoG-based human shape detector [4]. We processed all the captured frames with the detector program provided by the authors. Obtained coordinates of the possible human locations are input to the tracking systems.

The state-space model is identical with the previous experiments. The initial hyper parameters of NIW distribution $\theta(0)$ are selected so that the average of $\xi_k(t)$ and $\psi_k(t)$ takes the following values: $\boldsymbol{q} = \mathbf{0}$, $\boldsymbol{Q} = diag\{25.0, 25.0\}$, $\boldsymbol{r} = \mathbf{0}$ and $\boldsymbol{R} = diag\{10.0, 10.0\}$. The length of the time series data is 300 steps that are extracted from 600 frames of 15 FPS-captured movie data. The number of particles is $S = 1000$, and CRP concentration parameter is set to $\gamma = 1.0$. The target birth probability $P_b = 0.1$ and target death probability P_d is calculated by Eq. (21) with $\lambda = 0.1$.

The two rightmost columns of Table 3 show the averaged data likelihood for the real movie data experiments. As the synthetic data experiment, our proposed model successfully improved the data log likelihood.

Next we present the distribution of $\boldsymbol{q}$ in Fig. 6 for the first real movie data, and Fig. 7 for the second real movie data, respectively ($\pi = 0.1$). Since these experiments are done on real-world data, we do not have the ground truth of the number and the specifications of the dynamics. For the first experiment (Fig. 6), we observe that the obtained two dynamics clusters are well matched to the expected result. For the second experiment (Fig. 7), the system generated two clusters for each of three directions (upward, downward and right-to-left). This result well matches to the expected result: i.e. we have roughly three dynamics patterns and each pattern is divided into smaller clusters so as to reflect the complex behaviors.

Finally we present snapshots from the tracking results in Fig. 8 for the first experiment and Fig. 9 for the second real movie data experiment, respectively. In the figures, estimated object coordinates are denoted with rectangles. The color of each rectangle denotes the target's ID (i). Numbers on the rectangles represent the indices of the pattern of dynamics (z) in Fig. 6 and Fig. 7. The results show that the proposed model is able to identify hidden states of multiple targets and estimate the unknown patterns of dynamics.

5. Conclusion

In this paper, we presented a novel on-line probabilistic generative model that simultaneously deals with both the clustering and the tracking of multiple moving objects. We assume that time series data are generated from an unknown number of hidden trajectories of target objects, each of which is governed by an unknown number of different dynamics parameters. Our model can infer the property of a mixture model whose number of mixture components is unknown through a nonparametric Bayes model with conjugate priors. The developed model can simultaneously learn and infer the hidden states of each target object as well as their patterns of dynamics and characteristics. Through experiments with synthetic and real movie data, we confirmed that the proposed model successfully learned the hidden cluster patterns and obtained better tracking results than conventional models without clustering.

In this paper, we modeled rather simple state-space schemes. One possible extension of this work is to incorporate more complex models into the learning scheme, such as object appearance or structure models.

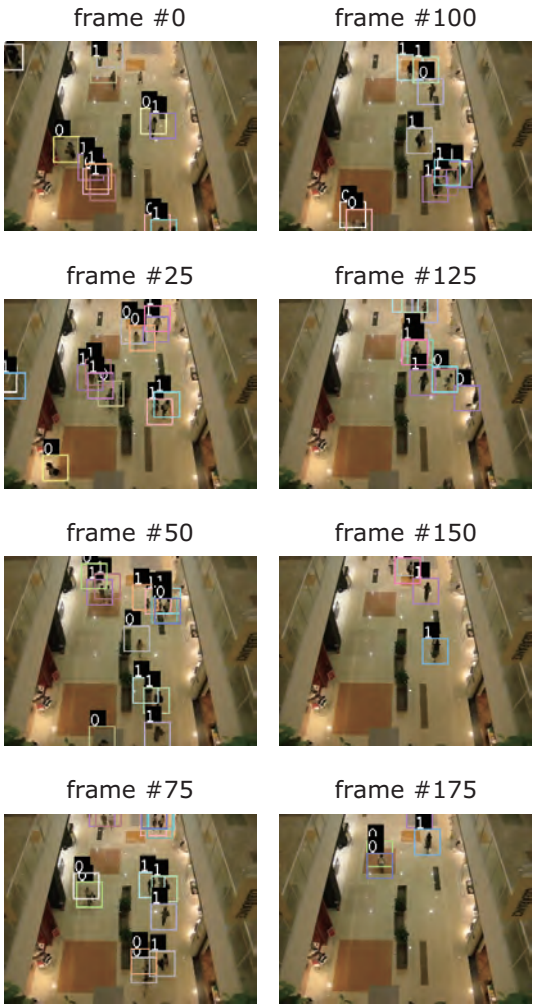

Figure 8. Tracking results of the first real movie experiment

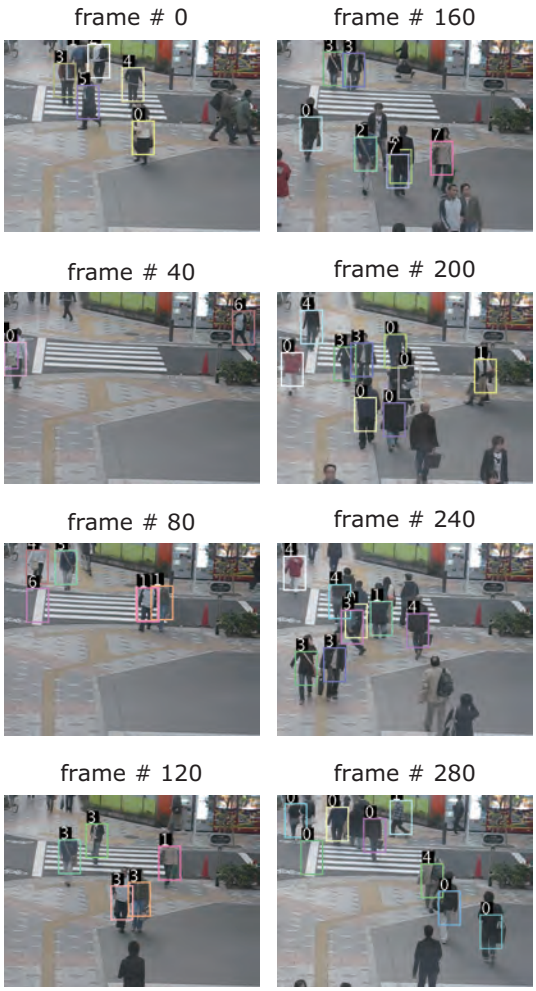

Figure 9. Tracking results of the second real movie experiment

References

[1] D. Blackwell and J. B. MacQueen. Ferguson distributions via Polya urn schemes. *The Annals of Statistics*, 1(2):353–355, 1973. 2, 4

[2] F. Caron, M. Davy, A. Doucet, E. Duflos, and P. Vanheeghe. Bayesian inference for linear dynamic models with Dirichlet process mixtures. *IEEE Trans. Signal Processing*, 56(1):71–84, 2008. 1, 2, 3, 5

[3] M. J. Cassidy and W. D. Penny. Bayesian nonstationary autoregressive models for biomedical signal analysis. *IEEE Trans. Biomedical Engineering*, 49(10):1142–1152, 2002. 1

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, volume 2, pages 886–893, 2005. 7

[5] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proc. AAAI*, 2006. 2

[6] Z. Khan, T. Balch, and F. Dellaert. A Rao-Blackwellized particle filter for eigentracking. In *Proc. CVPR*, pages 980–986, 2004. 1

[7] Z. Khan, T. Balch, and F. Dellaert. MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *IEEE Trans. PAMI*, 28(12):1960–1972, December 2006. 1, 3

[8] B. Leibe, N. Cornelis, K. Cornelis, and L. V. Gool. Dynamic 3D scene analysis from a moving vehicle. In *Proc. CVPR*, 2007. 1, 6

[9] T. W. Liao. Clustering of time series - a survey. *Pattern Recognition*, 38(11):1857–1874, 2005. 1

[10] M. Ramoni, P. Sebastiani, and P. Cohen. Bayesian clustering by dynamics. *Machine Learning*, 47(1):91–121, 2002. 1

[11] S. Särkkä, A. Vehtari, and J. Lampinen. Rao-Blackwellized particle filter for multiple target tracking. *Information Fusion*, 8(1):2–15, 2007. 1, 2, 3, 4, 5

[12] P. Smyth. Clustering sequences with hidden markov models. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 648–654. MIT Press, 1997. 1