Generative Models for Evolutionary Clustering

TIANBING XU (1) and ZHONGFEI ZHANG (1,2),
(1) Department of Computer Science, State University of New York at Binghamton
(2)Zhejiang Provincial Key Lab of Information Network Technology,
Dept. of Information Science and Electronics Engineering, Zhejiang University, China
PHILIP S. YU, Department of Computer Science, University of Illinois at Chicago
BO LONG, Yahoo! Inc.

This paper studies evolutionary clustering, a recently emerged hot topic with many important applications, noticeably in dynamic social network analysis. In this paper, based on the recent literature on Nonparametric Bayesian models, we have developed two generative models DPChain and HDP-HTM. DPChain is derived from the Dirichlet Process Mixture (DPM) model with an exponential decaying component along with the time. HDP-HTM combines the Hierarchical Dirichlet Process (HDP) with a Hierarchical Transition Matrix (HTM) based on the proposed Infinite Hierarchical Markov State model (iHMS). Both models substantially advance the literature on evolutionary clustering in the sense that not only they both perform better than the existing literature, but more importantly they are capable of automatically learning the cluster numbers and explicitly addressing the correspondence issues over the evolution. Extensive evaluations have demonstrated the effectiveness and the promise of these two solutions against the state-of-the-art literature.

Categories and Subject Descriptors: H.2.8 [**Database Applications**]: Data mining; G.3 [**Probability and Statistics**]: Nonparametric statistics; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*

General Terms: Algorithms

Additional Key Words and Phrases: Evolutionary Clustering, DPChain, HDP-HTM, iHMS, Hierarchical Transition Matrix

1. INTRODUCTION

Evolutionary clustering has a wide spectrum of applications, such as daily news analysis to observe the changing news foci, blog analysis to observe the community development and evolution, and scientific publication analysis to identify the new and hot research directions in a specific area. As a result, evolutionary clustering research has recently emerged as a hot and active research topic in data mining. Evolutionary clustering refers to the scenario where a collection of data evolves over the time; at each time, the collection of the data has a number of clusters; when the collection of the data evolves from one time to another, new data items may join the collection and existing data items may disappear; similarly, new clusters may appear and at the same time existing clusters may disappear. Consequently, both the data items and the clusters of the collection may change over the time, which poses a great challenge to the problem of evolutionary clustering as the model selection problem in the traditional clustering is still an open problem.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

Chakrabarti et al. in 2006 [Chakrabarti et al. 2006] were probably considered as the first to address the evolutionary clustering problem in the data mining literature. In their work, a general framework was proposed and two specific clustering algorithms within this framework were developed — evolutionary k-means and evolutionary agglomerative hierarchical clustering. Recently, Chi et al. [Chi et al. 2007] presented an evolutionary spectral clustering approach by incorporating the temporal smoothness constraint into the solution. In order to fit the current data well into the clustering but at the same time not to deviate the clustering from the history too dramatically, the temporal smoothness constraint is incorporated into the overall measure of the clustering quality. Based on the spectral clustering approach, two specific algorithms, PCM and PCQ, were proposed.

These two efforts were developed by explicitly incorporating the history clustering information into the existing classic clustering algorithms, specifically, k-means, agglomerative hierarchical clustering, and spectral clustering approaches [Ng et al. 2002; Shi and Malik 2000]. While incorporating the history information into the evolutionary clustering certainly advances the literature on this topic, there is a very restrictive assumption in their work – it is assumed that the number of the clusters over the time stays the same. It is clear that in many applications of evolutionary clustering, this assumption is violated.

From the statistical point of view, we may first model the data collection as a generative process in order to describe the generation of a sample or data point at each time; then a solution to the evolutionary clustering problem may be made as an inference to learn the distribution of the data at different times consistent with the original data distribution. Consequently, the following two properties are natural to a typical evolutionary clustering problem: (1) The number of clusters as well as the clustering structures at different evolutionary times may change. (2) The clusters of the data between neighboring times should stay the same or have a smooth change; but after a long time, clusters may drift substantially.

Since some clusters at different times might be the same while others may be different, another challenging problem is the correspondence problem, which refers to the correspondence among different local clusters across different times, resulting in the cluster-cluster correspondence and the cluster transition correspondence issues. We assume that the cluster structure at each time follows a mixture model of the clusters for the data collection at this time. Thus, clusters at different times may share common clusters, resulting in explicitly addressing the cluster-cluster correspondence issue. Further, these clusters evolve over the time and some may become more popular while others may become outdated, making the cluster structures and the number of the clusters change over the time.

Consequently, we propose two statistical models as the two solutions to the evolutionary clustering problem — DPChain and HDP-HTM [Xu et al. 2008a; 2008b]. The DPChain model is based on the Dirichlet Process Mixture (DPM) model [Antoniak 1974; Escobar and West 1995], which automatically learns the number of the clusters from the evolutionary data; in addition, the exponential decaying trend is used to model the change of the cluster mixture proportion over the time. In the HDP-HTM model, we apply the Hierarchical Dirichlet Processes (HDP) [Teh et al. 2007] to handle the global and local cluster correspondence problems; further, we develop the state transition matrix to explicitly reflect the cluster-cluster transitions between different times. These solutions are proven to work well in different real-world evolutionary clustering applications. They are capable of automatically learning the number of the clusters at each time during the evolution. In addition, the clustering performances are more accurate than those in the existing literature.

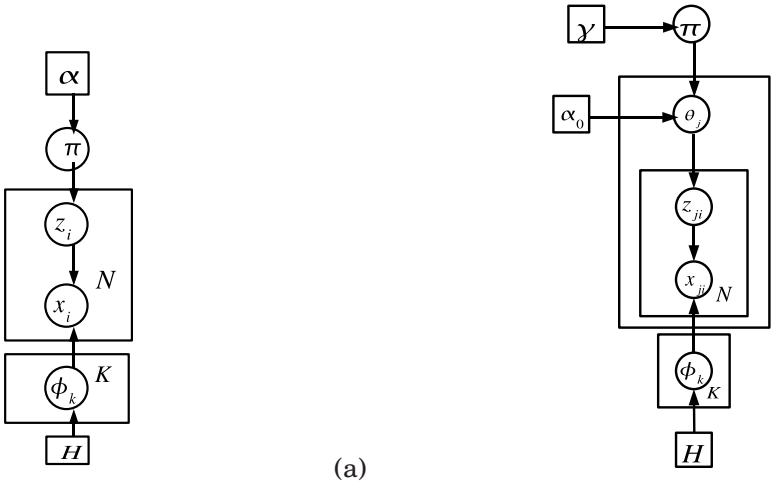

Fig. 1. Graphical Models (a) DPM model (b) HDP model

In the following text, boldface symbols are used to denote vectors or matrices, and non-boldface symbols are used to denote scalar variables. Also for all the variables we have defined, adding a symbol $-s$ either in the subscript or in the superscript of a variable defines the whole range of the variable except for the item indicated as s.

The remainder of this paper is outlined as follows. Section 2 reviews the related statistical background and Section 3 further reviews the related work in the literature. In Section 4 we introduce the first model DPChain and how to learn that model. Sections 5 and 6 describe the iHMS model as the Hierarchical Transition Matrix for HDP-HTM and the representation and inference method of HDP-HTM model. Section 7 reports the experimental results on three data sets for the proposed DPChain and HDP-HTM models against the exisiting evolutionary clustering algorithms (PCQ and PCM) and the related models LDA and HDP from the literature. In Section 7.3, we discuss the potential application of HDP-HTM on the community discovery of dynamic social networks. Finally, in Section 8, we conclude the paper.

2. RELATED STATISTICAL BACKGROUND

The Dirichlet process (DP) [Ferguson 1973] is a distribution over distributions, and is usually used as a prior in Nonparametric Bayesian models. The definition of a Dirchelt process follows [Teh 2007].

Definition 2.1. Let base distribution H be a distribution over a parameter space and α be a positive real number. For any finite measurable partition $A_1, A_2, ...A_k$ of the parameter space, the vector $(G(A_1), G(A_2), ...G(A_k))$ is a random vector. We denote G as the Dirichlet Process with parameter α and base measure H with $G \sim \mathcal{DP}(\alpha, H)$ if

$$(G(A_1), G(A_2), ...G(A_k)) \sim Dir(\alpha H(A_1), \alpha H(A_2), ...\alpha(H(A_k)))$$

The Dirichlet Process can be represented as a Chinese Restaurant Process (CRP) [Aldous 1983] — a distribution over partitions. Suppose that we have a Chinese restaurant with an infinite number of tables; each table is capable of holding an infinite number of customers. The first customer comes in and sits at the first table; the subsequent customers may sit either randomly at a table already with customers with a

probability proportional to the number of the customers already at that table, or at a new table with a probability proportional to a constant parameter.

Draws from $\mathcal{DP}$ can also be represented as a weighted sum of point masses. The stick breaking process (also known as GEM) [Sethuraman 1994] provides a constructive definition of Dirichlet Process as follows:

$$G = \sum_{k=0}^{\infty} \pi_k \delta_{\phi_k} \quad \phi_k \sim H$$
$$\pi_k = \beta_k \prod_{l=1}^{k-1} (1 - \beta_l) \quad \beta_k \sim Beta(1, \alpha) \tag{1}$$

Here we define $\boldsymbol{\pi}$ as $stick(\alpha)$ distributed in terms of Eq. 1. π_k is the weight proportion of cluster k. The overall proportions of an infinite number of the clusters are summed to 1.

Different from the traditional finite mixture models with a finite number of clusters, Dirichlet Process Mixture model [Antoniak 1974] is able to represent an infinite number of clusters. According to the stick breaking process representation of the Dirichlet Process, we have a description of DPM shown in Figure 1(a).

$$\begin{aligned} \phi_k|H &\sim H \\ \boldsymbol{\pi}|\alpha &\sim stick(\alpha) \\ z_i|\boldsymbol{\pi} &\sim Mult(\boldsymbol{\pi}) \\ x_i|z_i, \{\phi_t\} &\sim F(\phi_{z_i}) \end{aligned}$$

Here z_i is the cluster assignment taking cluster k with probability π_k; H is the prior distribution for the cluster parameters.

HDP [Teh et al. 2007] is a hierarchical extension of DPM as is shown in Figure 1(b). The process defines a global random probability measure G_0 distributed as a Dirichlet process, and a set of random probability measures G_j, each of which forms a Dirichlet process controlled by the global measure G_0.

Taking advantage of the strick breaking representation of Dirichlet process, we have the representation of HDP:

$$\begin{aligned} \phi_k|H &\sim H \\ \boldsymbol{\pi}|\gamma &\sim stick(\gamma) \\ \boldsymbol{\theta}_j|\alpha_0, \boldsymbol{\pi} &\sim \mathcal{DP}(\alpha_0, \boldsymbol{\pi}) \\ z_{j,i}|\boldsymbol{\theta}_j &\sim Mult(\boldsymbol{\theta}_j) \\ x_{j,i}|z_{j,i}, \{\phi_k\} &\sim F(\phi_{z_{j,i}}) \end{aligned}$$

To extend Hidden Markov Model (HMM) [Rabiner 1989] to an infinitely countable number of states, Beal et. al. [Beal et al. 2002] proposed the Infinite Hidden Markov Model (iHMM), which has an infinitely coutable number of states in space $S = \{1, 2, ..., k, ...\}$. The Dirichlet process (represented as a stick-breaking process) is adopted to model the probabilities. The initial state probabilities for each state $\boldsymbol{\pi}$ are the stick breaking strengths:

$$\boldsymbol{\pi}|\alpha \sim stick(\alpha)$$

Each row of the transition matrix can also be constructed as a Dirichlet process with probability

$$\pi_{i\to j} = p(s_{t+1} = j | s_t = i) = \begin{cases} \frac{n_{i\to j}}{(\sum_{j=1}^{K} n_{i\to j} + \beta)} & \text{if } k \in \{1, 2, ..., K\} \\ \frac{\beta}{(\sum_{j=1}^{K} n_{i\to j} + \beta)} & \text{if } k \text{ is a new state} \end{cases}$$

Here we assume that K states have appeared at the current time; β is the concentration parameter; $n_{i\to j}$ is the expected number of transitions from state i to state j.

Beal et al. [Beal et al. 2002] go further resulting in an HDP to model the transition matrix. Similarly, the emission matrix is also constructed by HDP. Recently, Fox et al. [Fox et al. 2007] have revisited the HDP-HMM model and have developed methods which allow more efficient and effective learning from realistic time series data. Ni et al. [Ni et al. 2007] have proposed a new hierarchical Nonparametric Bayesian model by imposing a nested Dirichlet process prior to the base distributions of iHMMs to learn the sequential data. More recently, Gael et al. [Gael et al. 2008] have introduced a new inference algorithm for iHMM called the beam sampling algorithm which is also more efficient and robust. There are also many interesting efforts on topic modeling beyond the bag of words approaches [Griffiths et al. 2005; Boyd-Graber and Blei 2008; Gruber et al. 2007; Wallach 2006]. Work in [Griffiths et al. 2005; Boyd-Graber and Blei 2008; Gruber et al. 2007] mainly combines latent topic modeling for document semantic information and Markov modeling for sequential information. The paper [Wallach 2006] incoporates latent variables and n-gram statistics to form a hierarchical Dirichlet bigram language model.

3. FURTHER RELATED WORK

There are many noticeable applications of the Dirichlet process based models in text analysis and topic modeling. Blei et al. [Blei et al. 2003] proposed the well-known Latent Dirichlet Allocation (LDA) model for text modeling and clustering with an assumed known constant number of the topics set in advance. For the topic evolution analysis, Blei and his colleagues [Blei and Lafferty 2006; Wang et al. 2008] have designed the probabilistic models to develop effective solutions. Based on LDA, Griffiths et al. [Griffiths and Steyvers 2004] tried to identify "hot topics" and "cold topics" by the text temporal dynamics. The number of the topics was decided by a Bayesian model selection. Wang et al. [Wang and McCallum 2006] introduced an LDA-style topic model to represent the time as an observed continuous variable attempting to capture the topic evolutionary trends. Zhu et al. [Zhu et al. 2005] further developed a time-sensitive Dirichlet process mixture model for clustering documents with the temporal correlations between time instances considered. However, all these models fail to automatically learn the number of the topics (i.e., the clusters). Further, they also fail to address the correspondence issues during the evolution.

There are also many methods developed for community discovery based on graph partitioning such as [Flake et al. 2000] and [Abou-Rjeili and Karypis 2006]. Recently, statistical graphical models provide new promising solutions to this problem. McCallum et al. [Mccallum et al. 2005] proposed the Author-Recipient-Topic (ART) model for social network analysis to learn the topic distribution based on LDA and AT (Author-Topic [Rosen-Zvi et al. 2004]) model. Zhang et al. [Zhang et al. 2007; Zhang et al. 2007; Zhang et al. 2007] proposed a series generative models SSN-LDA, GWN-LDA, and HSN-PAM to address this problem. In SSN-LDA (Simple Social Network LDA), communities are modeled as the latent variables in the model and are defined as the distributions over the social actor space. GWN-LDA (Generic weighted network-LDA) is a hierarchical Bayesian model derived from the LDA model, for discovering the

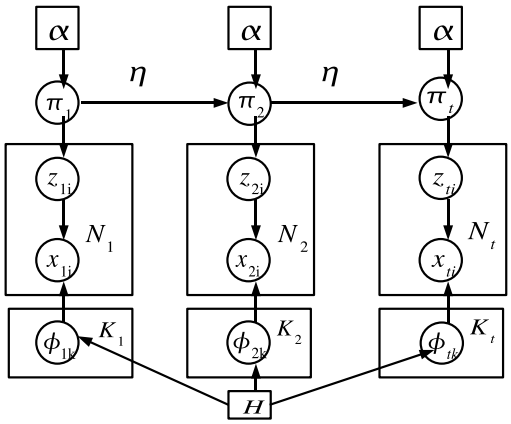

Fig. 2. The DPChain Model

probabilistic community profiles in social networks based on two different network encoding approaches. In HSN-PAM (Hierarchical Social Network-Pachinko Allocation Model) communities are classified into two categories: super-communities and regular-communities; the model is able to discover not only the correlations among the social actors but also the correlations among the hidden groups. Xu et al. [Xu et al. 2008] proposed the infinite hidden relational model (IHRM) for social network modeling and analysis. In order to incorporate both the link and content information into the analysis, Pathak et al. [Pathak et al. 2008] presented a social topic model CART (community-Author-Receipient-Topic) for the community extraction. However, all these models only consider the static social networks and ignore the dynamics of social networks when the evolution is in consideration. Furthermore, these models do not consider how to automatically learn and track the number of the social communities. More recently, to do statitiscal inference for diverse networks ranging from biology to social sciences, Ahmed et al. [Ahmed and Xing 2009] presented a machine learning method built upon a temporally smoothed l_1-regularized logistic regression formalism as a convex-optimization problem which may be sovled efficiently and is scalable to large networks. Fortunato et al. [Fortunato 2010] gave a comprehensive review on the community detection in graphs, including traditional and state-of-the-art techniques such as graph partitioning, spectral algorithms, hierarchical clustering, and probabilistic graphical models (including generative models).

4. DIRICHLET PROCESS MIXTURE CHAIN (DPCHAIN)

The first model we propose is based on the DPM [Antoniak 1974; Escobar and West 1995], which is called DPChain model in this paper. For DPChain model, we assume that at each time t a collection of data has K_t clusters and each cluster is derived from a unique distribution. K_t is unknown and is learned from the data. We denote N_t as the number of the data items in this collection at time t.

4.1. DPChain Representation

Figure 2 illustrates the DPChain model. We incorporate the indicator variables to represent the DPChain model. First we introduce the notations. α denotes the concentration parameter for a Dirichlet Process. H denotes the base measure of a Dirichlet distribution with the pdf h. F denotes the distribution of the data with the pdf f. $\phi_{t,k}$ denotes the parameter of cluster k of the data at time t. At time t, $\phi_{t,k}$ is a sample from

distribution H, represented as a parameter of F.

$$\phi_{t,k}|H \sim H$$

$\boldsymbol{\pi}_t$ is the cluster mixture proportion vector at time t. $\pi_{t,k}$ is the weight of the corresponding cluster k at time t. Consequently, $\boldsymbol{\pi}_t$ is distributed as $stick(\alpha)$ [Sethuraman 1994] which is described as follows.

$$\boldsymbol{\pi}_t = (\pi_{t,k})_{k=1}^{\infty} \quad \pi_{t,k} = \pi_{t,k}{}' \prod_{l=1}^{k-1}(1-\pi_{t,l}) \quad \pi_{t,k}{}' \sim Beta(1,\alpha) \tag{2}$$

Let $z_{t,i}$ be the cluster indicator at time t for data item i. $z_{t,i}$ follows a multinomial distribution with parameter $\boldsymbol{\pi}_t$.

$$z_{t,i}|\boldsymbol{\pi}_t \sim Mult(\boldsymbol{\pi}_t)$$

Let $x_{t,i}$ denote data item i from the collection at time t. $x_{t,i}$ is modeled as being generated from F with parameter $\phi_{t,k}$ by the assignment $z_{t,i}$.

$$x_{t,i} \mid z_{t,i}, (\phi_{t,k})_{k=1}^{\infty} \sim f(x|\phi_{t,z_{t,i}})$$

In evolutionary clustering, cluster k smoothly changes from time $t-1$ to t. With this change of the clustering, the number of the data items in each cluster may also change. Consequently, the cluster mixture proportion as an indicator for the population of a cluster also changes accordingly. In the classic DPM model, $\boldsymbol{\pi}_t$ represents the cluster mixture. We extend the classic DPM model to the DPChain model by incorporating the temporal information into $\boldsymbol{\pi}_t$. When a cluster smoothly changes, more recent history has more influence on the current clustering than less recent history. Thus, a cluster with a higher mixture proportion at the current time is more likely to have a higher proportion at the next time. Hence, the cluster mixture at time t may be constructed as follows.

$$\boldsymbol{\pi}_t = \sum_{\tau=1}^{t} \exp\{-\eta(t-\tau)\}\boldsymbol{\pi}_\tau \tag{3}$$

where η is a smooth parameter.

This relationship is further illustrated by an extended CRP [Blackwell and MacQueen 1973; Aldous 1983]. We denote $n_{t,k}$ as the number of the data items in cluster k at time t, and $n_{t,k}^{-i}$ as the number of the data items belonging to cluster k except $x_{t,i}$; $w_{t,k}$ is the prior smooth weight for cluster k at the beginning of time t. According to Eq. 3, $w_{t,k}$ has the relationship to $n_{\tau,k}$ at the previous time τ:

$$w_{t,k} = \sum_{\tau=1}^{t-1} \exp\{-\eta(t-\tau)\}n_{\tau,k} \tag{4}$$

Then, similar to CRP, the prior probability to sample a data item from cluster k given the history assignment $\{\mathbf{z}_1 \ldots \mathbf{z}_{t-1}\}$ and the other assignment at time t, $\mathbf{z}_{t,-i} = \mathbf{z}_t \setminus z_{t,i}$ is defined as follows.

$$p(z_{t,i} = k|\mathbf{z}_1, \ldots \mathbf{z}_{t-1}, \mathbf{z}_{t,-i}) \propto \begin{cases} \frac{w_{t,k}+n_{t,k}^{-i}}{\alpha+\sum_{j=1}^{K_t} w_{t,j}+n_t-1} & \text{if } k \text{ is an existing cluster} \\ \frac{\alpha}{\alpha+\sum_{j=1}^{K_t} w_{t,j}+n_t-1} & \text{if } k \text{ is a new cluster} \end{cases} \tag{5}$$

where $n_t - 1$ is the number of the data items at time t except for $x_{t,i}$, and $x_{t,i}$ is considered as the last data item in the collection at time t. With Eq. 5, an existing cluster

appears again with a probability proportional to $w_{t,k} + n_{t,k}^{-i}$, while a new cluster appears as the first time with a probability proportional to α. If at time t as well as the times before t, the data of cluster k appear infrequently, cluster k has a relatively small weight to appear again in the future time, which leads to a higher probability of becoming death for cluster k. Consequently, this model has the capability to describe the birth and death of a cluster over the evolution. The data item generation process for the DPChain model is listed as follows.

(1) Sample cluster parameter $\phi_{t,k}$ from the base measure H at each time. The number of the clusters is not a fixed prior parameter but is decided by the data when a new cluster is needed.
(2) First, sample the cluster mixture vector $\boldsymbol{\pi}_t$ from $stick(\alpha)$ at each time; then, $\boldsymbol{\pi}_t$ is further smoothly weighted from the exponential sum according to Eq. 3.
(3) At time t, sample the cluster assignment $z_{t,i}$ for data item $x_{t,i}$ from the multinomial distribution with parameter $\boldsymbol{\pi}_t$.
(4) Finally, a data item $x_{t,i}$ is generated from distribution $f(x|\phi_{t,z_{t,i}})$ given the cluster index variable $z_{t,i}$ and the cluster parameter $\phi_{t,k}$.

At each time t, the concentration parameter α may be different. In the sampling process, we just sample α from a Gamma distribution at each iteration. For a more sophisticated model, α may be modeled as a random variable varying with time, as the rate of generating a new cluster may change over the time.

4.2. DPChain Inference

Given the DPChain model, we use Markov Chain Monte Carlo (MCMC) method [Neal 1993] to sample the cluster assignment $z_{t,i}$ for each data item at time t. Specifically, following the Gibbs sampling [Casella and George 1992], the aim is to sample the posterior cluster assignment $z_{t,i}$, given the whole data collection $\mathbf{x}_t$ at time t, the history assignment $\{\mathbf{z}_1 \dots \mathbf{z}_{t-1}\}$, and other assignment $\mathbf{z}_{t,-i}$ at the current time.

We denote $\mathbf{x}_{t,-i}$ as the whole data collection at time t except for $x_{t,i}$. The posterior of the cluster assignment is determined by Bayes rule:

$$\begin{aligned} &p(z_{t,i} = k|\mathbf{x}_t, \mathbf{z}_{t,-i}, \mathbf{z}_1, \dots \mathbf{z}_{t-1}) \propto \\ &p(x_{t,i}|\mathbf{z}_{t,-i}, \mathbf{z}_1, \dots \mathbf{z}_{t-1}, {\mathbf{x}_k}^{-i}) p(z_{t,i} = k|\mathbf{z}_1, \dots \mathbf{z}_{t-1}, \mathbf{z}_{t,-i}) \end{aligned} \tag{6}$$

where ${\mathbf{x}_k}^{-i} = \{x_{t,j} : z_{t,j} = k, j \neq i\}$ donates all the data at time t assigned to cluster k except for $x_{t,i}$.

Since $z_{t,i}$ is conditionally independent of $\mathbf{x}_{t,-i}$ given all the history assignment and the current time assignment except for $x_{t,i}$, we omit $\mathbf{x}_{t,-i}$ at the second term in the right hand side of Eq. 6. Further, denote $f_k^{-i}(x_{t,i})$ as the first term in the right hand side of Eq. 6, which is the conditional likelihood of $x_{t,i}$ on cluster k, given the other data associated with k and other cluster assignment.

If k is an existing cluster:

$$f_k^{-i}(x_{t,i}) = \int f(x_{t,i}|\phi_{t,k}) \cdot h(\phi_{t,k}|\{x_{t,j} : z_{t,j} = k, j \neq i\}) d\phi_{t,k} \tag{7}$$

where $h(\phi_{t,k}|\{x_j : z_{t,j} = k, j \neq i\})$ is the posterior distribution of parameter $\phi_{t,k}$ given observation $\{x_{t,j} : z_{t,j} = k, j \neq i\}$. If F is conjugate to H, the posterior of $\phi_{t,k}$ is still in the distribution family of H. Then we can integrate out $\phi_{t,k}$ to compute $f_k^{-i}(x_{t,i})$. Here we only consider the conjugate case because our experiments reported in this paper are based on this case. For the non-conjugate case, a similar inference method may be obtained [Neal 2000].

For a new cluster k, it is equivalent to computing the marginal likelihood of $x_{t,i}$ by integrating out all the parameters sampled from H.

$$f_k^{-i}(x_{t,i}) = \int f(x_{t,i}|\phi_{t,k})dH(\phi_{t,k}) \tag{8}$$

Finally, the posterior cluster assignment in the conjugate case is given as:

$$p(z_{t,i} = k|\mathbf{x}_t, \mathbf{z}_{t,-i}, \mathbf{z}_1, \ldots \mathbf{z}_{t-1}) \propto \begin{cases} \frac{w_{t,k}+n_{t,k}^{-i}}{\alpha+\sum_{j=1}^{K_t} w_{t,j}+n_t-1} f_k^{-i}(x_{t,i}) & \text{if } k \text{ is an existing cluster} \\ \frac{\alpha}{\alpha+\sum_{j=1}^{K_t} w_{t,j}+n_t-1} f_k^{-i}(x_{t,i}) & \text{if } k \text{ is a new cluster} \end{cases} \tag{9}$$

4.3. Parameter Learning

We use the EM method [Dempster et al. 1977] to learn hyperparameters (α, η). According to Eq. 4, updating η results directly in updating $w_{t,k}$. Consequently, we actually update the hyperparameters $\Theta = (\alpha, w_{t,k})$. Following [Escobar and West 1995], α is sampled from the Gamma distribution at each iteration in the Gibbs sampling in the E-step. In the M-step, similar to [Zhu et al. 2005], we update $w_{t,k}$ by maximizing the cluster assignment likelihood. Suppose that, at an iteration, there are K clusters.

$$w_{t,k}^{new} = \frac{n_{t,k}}{\alpha + n_t - 1} \cdot \sum_{j=1}^{K} w_{t,j}^{old} \tag{10}$$

The EM framework works as follows:

— At time t, initialize parameters Θ and $z_{t,i}$
— E-Step: Sample α from a Gamma distribution. Sample cluster assignment $z_{t,i}$ for data item $x_{t,i}$ by Eq. 9.
— M-Step: Update $w_{t,k}$ by Eq. 10.
— Iterate the E-Step and the M-Step until the EM converges.

5. INFINITE HIERARCHICAL MARKOV STATE MODEL

The previous section focuses on DPChain model which does not explicitly address the cluster correspondence issue; thus we introduce another model to explicitly address this issue. Here, we propose a new infinite hierarchical hidden Markov state model (iHMS) to construct the Hierarchical Transition Matrix (HTM) and to provide a posterior inference scheme for HTM in the new model (covered in detail in Section 6).

5.1. Hierarchical Transition Matrix

Traditionally, HMM has a *finite* state space with K hidden states, say $\{1, 2, \ldots K\}$. For the hidden state sequence $\{s_1, s_2, \ldots s_T\}$ up to time T, there is a K by K state transition probability matrix $\mathbf{\Pi}$ governed by Markov dynamics with all the elements $\pi_{i\to j}$ of each row $\boldsymbol{\pi}_i$ summed to 1.

$$\pi_{i\to j} = p(s_t = j|s_{t-1} = i)$$

Here we elect to use s as another notation for a state in order to differentiate from the other state notation z. The initial state probability for state i is $p(s_1 = i)$ with the summation of all the initial probabilities equal to 1. For observation x_t in the observation sequence $\{x_1, x_2, \ldots x_T\}$, given state $s_t \in \{1, 2, \ldots, K\}$, there is a parameter ϕ_{s_t} drawn from the base measure H which parameterizes the observation likelihood probability.

$$x_t|s_t \sim F(\phi_{s_t})$$

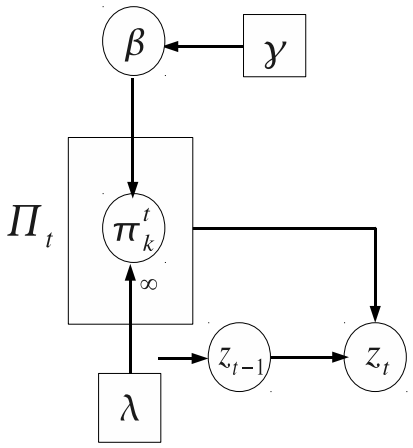

Fig. 3. The Hierarchical Transition Matrix by iHMS

However, when dealing with a countably *infinite* state space, $\{1, 2, \ldots K, \ldots\}$, we must adopt a new model similar to that in [Beal et al. 2002] for a state transition probability matrix with an *infinite* matrix dimension. $\boldsymbol{\pi}_i$, the ith row of the transition probability matrix $\mathbf{\Pi}$, may be represented as the mixing proportions for all the next infinite states, given the current state. As a result, we model it as a Dirichlet process with an infinite dimension with the summation of all the elements in a row to 1, which leads to an infinite number of DPs' construction for an infinite number of rows of a transition probability matrix.

With no further prior knowledge on the state sequence, a typical prior for the transition probability may be the symmetric Dirichlet distributions. Similar to [Teh et al. 2007], we intend to construct a hierarchical Dirichlet model to keep different rows of the transition probability matrix to share part of the prior mixing proportions of each state at the top level. Consequently, we adopt a new state model, Infinite Hierarchical Markov State model (iHMS), to construct the Infinite Transition Probability Matrix which is called the Hierarchical Transition Matrix (HTM).

Similar to HDP [Teh et al. 2007], we draw a random probability measure on the infinite state space $\boldsymbol{\beta}$ as the top level prior from $stick(\gamma)$ represented as the mixing proportions of each state.

$$\boldsymbol{\beta} \sim stick(\gamma)$$

The mixing proportion of state k, β_k, may also be interpreted as the prior mean of the transition probabilities leading to state k. Hence, $\boldsymbol{\beta}$ may be represented as the prior for Dirichlet process measure of a transition probability.

For the ith row of the transition matrix, $\boldsymbol{\pi}_i$, we sample it from $\mathcal{DP}(\lambda, \boldsymbol{\beta})$ with a smaller concentration parameter λ implying a larger variability around the mean measure $\boldsymbol{\beta}$. $\boldsymbol{\pi}_i$ is distributed as $stick(\lambda)$ similar to Eq. 2:

$$\boldsymbol{\pi}_i \sim stick(\lambda)$$

Specifically, the jth element $\pi_{i\to j}$ is the state transition probability from the previous state i to the current state j as $p(s_t = j|s_{t-1} = i)$.

Now, each row of the transition probability matrix is represented as a Dirichlet process which shares the same reasonable prior to the mixing proportions of the states. For a new row corresponding to a new state k, we simply draw a transition probability vector $\boldsymbol{\pi}_k$ from $\mathcal{DP}(\lambda, \boldsymbol{\beta})$ as a row, resulting in constructing a countably infinite number of the rows of the transition probability matrix.

5.2. Extension of iHMS

From now on we use the notations more clearly for state vairables in the rest of the paper by introducing the time dimension. Let $z_{t,i}$ represent the state for data item i at time t, and $\boldsymbol{z}_t = \{z_{t,i}\}$ is a collection of the states at time t. The transition probability constructed by iHMS may be further extended to the scenario where there is more than one state at each time. Suppose that there is a countably infinite global state space $\boldsymbol{S} = \{1, 2, \ldots, K, \ldots\}$ including states in all the state space $\boldsymbol{S}_t$ at each time t, where $\boldsymbol{S}_t \subseteq \boldsymbol{S}$. Figure 3 shows our extended iHMS model to construct Hierarchical Transition Matrix (HTM) Π_t at each time t. For any state $z_{t,i} \in \boldsymbol{S}_t$ at time t and state $z_{t-1,i} \in \boldsymbol{S}_{t-1}$ at time $t-1$, we may adopt $\pi^t_{j\to k}$ to represent $p(z_t = k|z_{t-1} = j, \Pi_t) = \sum_i p(z_{t,i} = k|z_{t-1,i} = j)$ as the transition probability from state j to state k between times $t-1$ and t. This state transistion probability describes the relations between the states at adjacent times, not on the individual data item. Transition probability $\{\pi^t_{j\to k}\}$ connects the states $\boldsymbol{z}_{t-1}$ and $\boldsymbol{z}_t$ at adjacent times, and has a natural tendency for a state transition to appear more frequently if we have already encountered many such transitions. Thus, it is reasonable to model a row of transitions as a Dirichlet process. We will discuss this extension in detail later.

5.3. MAP Estimation of HTM

Let $\boldsymbol{X}$ be an observation sequence, which includes all the observations $\boldsymbol{x}_t$ at each time t, where $\boldsymbol{x}_t \in \boldsymbol{X}$. Now, the question is how to represent the countably infinite state space in a hierarchical state transition matrix (HTM). Note that, at each time, there is in fact a finite number of observations $\boldsymbol{x}_t$; the state space $\boldsymbol{S}_t$ at each time t must be arbitrarily finite even though conceptually the global state space $\boldsymbol{S}$ may be considered countably infinite. Further, we adopt the stick-breaking representation for the Dirichlet process [Teh et al. 2007; Ishwaran and James 2001] to iteratively handle an arbitrary number of the states and accordingly the transition probability matrix up to time t.

Suppose that up to time t there are K current states and we use $K+1$ to index a potentially new state. Then $\boldsymbol{\beta}$ may be represented as:

$$\boldsymbol{\beta} = \{\beta_1, \ldots \beta_K, \beta_u\} \quad \beta_u = \sum_{k=K+1}^{\infty} \beta_k \quad \sum_{k=1}^{K} \beta_k + \beta_u = 1 \tag{11}$$

Given $\boldsymbol{\beta}$, the Dirichlet prior measure of the jth row of the transition probability matrix $\boldsymbol{\pi}^t_j$ has the dimension $K+1$. The last element β_u is the prior measure of the transition probability from state j to an unrepresented state u.

When a new state is instantiated, we sample b from $Beta(1, \gamma)$, and set the new proportions for the new state $K^{new} = K+1$ and another potentially new state $K^{new}+1$ as:

$$\beta_{K^{new}} = b\beta_u \quad \beta^{new}_u = (1-b)\beta_u \tag{12}$$

Now, K is updated as K^{new}, β_u as β_u^{new}, and the number of the states may continue to increase if yet another new state is instantiated, resulting in a countably infinite transition probability matrix.

After we have observed data colletions $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_t$ at the two adjacent times and the state transtition correspondence inferred from the state indicators $\boldsymbol{z}_{t-1}$ and $\boldsymbol{z}_t$, respectively, the MAP (Maximum a posteriori) estimation of the transition matrix $\{\pi_{j\to k}^t\}$ at time t is (see Section Appendix):

$$\begin{aligned}\pi_{j\to k}^t &= p(z_t = k | z_{t-1} = j, \boldsymbol{z}_{t-1}, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t) \\ &= \begin{cases} \frac{n_{j\to k}^t - 1 + \lambda\beta_k}{n_j^t - K - 1 + \lambda} & k \text{ is an exisiting state} \\ \frac{\lambda\beta_u}{n_j^t - K - 1 + \lambda} & k \text{ is a new state} \end{cases}\end{aligned} \tag{13}$$

where $n_{j\to k}^t$ is the expected number of the transitions from state j to state k between the previous and the current times, and n_j^t is the expected number of the transitions out of state j. We run several monte carlo iterations to approximate the state transition counts.

$$n_{j\to k}^t \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{i\in\boldsymbol{x}_{t-1}\cap\boldsymbol{x}_t}\delta(z_{t-1,i}=j, z_{t,i}=k) \quad n_j^t = \sum_{k=1}^{K} n_{j\to k}^t$$

Here we use the Kronecker-delta function ($\delta(a,b) = 1$ iff $a = b$ and 0 otherwise) to count the number of the state transitions for all the common observations.

Intuitively, in Eq. 13 we may consider $\lambda\beta_k$ as the pseudo-observation of the transition from state j to k (i.e., the strength of the belief for the prior state transition), and $\lambda\beta_u$ as the probability of a new state transferred from state j. Besides $\boldsymbol{\beta}$ and λ, the transition matrices at different times are determined completely "given" the previous states $\boldsymbol{z}_{t-1}$, $\boldsymbol{z}_t$, and the data collections at times $t-1$ and t. Since the mean mixture proportion $\boldsymbol{\omega}_t$ (discussed in detail in the next section) is not able to provide the complete information as we cannot obtain $n_{j\to k}^t$ only from $\boldsymbol{\omega}_t$, $\boldsymbol{\omega}_t$ is dropped in Eq. 13.

6. HDP INCORPORATED WITH HTM (HDP-HTM)

To capture the state (cluster) transition correspondence during the evolution at different times, we propose HTM; at the same time, we must capture the state-state (cluster-cluster) correspondence, which may be handled by a hierarchical model with the top level corresponding to the global states [1] and the lower level corresponding to the local states, where it is natural to model the statistical process as HDP [Teh et al. 2007]. Consequently, we propose to combine HDP and HTM as a new HDP-HTM model, illustrated in Figure 4. At time 0, there are no state transitions, ω_t (explained later) is simply β, and this model in Figure 4 collapses into HDP.

6.1. HDP-HTM Representation

Let the global state space $\boldsymbol{S}$ denote the global cluster set, which includes all the states $\boldsymbol{S}_t \subseteq \boldsymbol{S}$ at all the times t. The global observation set $\boldsymbol{X}$ includes all the observations $\boldsymbol{x}_t$ at each time t, of which each data item i is denoted as $x_{t,i}$.

We draw the global mixture proportion from the global states $\boldsymbol{\beta}$ with the stick-breaking representation using the concentration parameter γ. The global measure G_0 may be represented as:

$$G_0 = \sum_{k=1}^{\infty}\beta_k\delta_{\phi_k}$$

[1]Each state is represented as a distinct cluster.

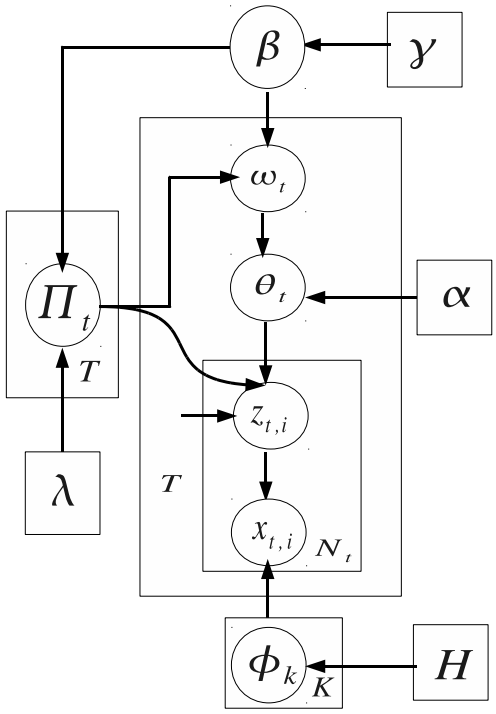

Fig. 4. The HDP-HTM Model

where ϕ_k is drawn from the base probability measure H with pdf h, and δ_{ϕ_k} is the concentration measure on ϕ_k.

Different from HDP, here we must consider the evolution of the data and the states (i.e., the clusters). The distribution of the clusters at time t is not only governed by the global measure G_0, but also is controlled by the data and cluster evolution in the history. Consequently, we make an assumption that the data and the clusters at time t are generated based on the previous data and cluster information, according to the mixture proportions of each cluster and the transition probability matrix. The global prior mixture proportions for the clusters are $\boldsymbol{\beta}$, and the state transition matrix $\boldsymbol{\Pi}_t$ provides the information of the state evolution between times $t-1$ and t. Now, the expected number of the data items generated by cluster k is proportional to the number of the data items in the clusters in the history multiplied by the transition probabilities from these clusters to state k; specifically, the mean mixture proportion for cluster k at time t, $\boldsymbol{\omega}_t$, is defined as follows:

$$\omega_{t,k} = \sum_{j=1}^{\infty} \beta_j \pi_{j\to k}^{t}$$

More precisely, $\boldsymbol{\omega}_t$ is further obtained by:

$$\boldsymbol{\omega}_t = \boldsymbol{\beta} \cdot \boldsymbol{\Pi}_t \tag{14}$$

Clearly, by the transition probability property, $\sum_{k=1}^{\infty} \omega_{t,k} = 1$, $\sum_{k=1}^{\infty} \pi_{j\to k}^{t} = 1$ and the stick-breaking property $\sum_{j=1}^{\infty} \beta_j = 1$.

$$\sum_{k=1}^{\infty} \omega_{t,k} = \sum_{k=1}^{\infty}\sum_{j=1}^{\infty} \beta_j \pi_{j\to k}^{t} = \sum_{j=1}^{\infty} \beta_j \sum_{k=1}^{\infty} \pi_{j\to k}^{t} = \sum_{j=1}^{\infty} \beta_j = 1$$

Thus, the mean mixture proportion $\boldsymbol{\omega}_t$ may be taken as the new probability measure at time t on the global cluster set. With the concentration parameter α, we draw the mixture proportion vector $\boldsymbol{\theta}_t$ from $\mathcal{DP}(\alpha, \boldsymbol{\omega}_t)$.

$$\boldsymbol{\theta}_t | \alpha, \boldsymbol{\omega}_t \sim \mathcal{DP}(\alpha, \boldsymbol{\omega}_t)$$

Now, at time t, the local measure G_t shares the global clusters parameterized by $\boldsymbol{\phi} = (\phi_k)_{k=1}^{\infty}$ with the mixture proportion vector $\boldsymbol{\theta}_t$.

$$G_t = \sum_{k=1}^{\infty} \theta_{t,k} \delta_{\phi_k}$$

At time t, given the mixture proportion of the clusters $\boldsymbol{\theta}_t$, the previous cluster assignments $\boldsymbol{z}_{t-1}$ and the transition probability Π_t, we draw a cluster indicator $z_{t,i}$ for data item $x_{t,i}$. For the simplification of the inference, we assume a multinomial distribution for $z_{t,i}$ with parameter $\boldsymbol{\theta}_t$ as an approximation. Intuitively, this is a local approximation assumption: at each time t, once we have a topic mixing proportion, we may simply draw a topic assignment z_t, while the history cluster information and the transition probability would also have an influence on $z_{t,i}$ through $\boldsymbol{\omega}_t$ to $\boldsymbol{\theta}_t$.

Once we have the cluster indicator $z_{t,i}$, data item $x_{t,i}$ may be drawn from distribution F with pdf f, parameterized by $\boldsymbol{\phi}$ from the base measure H.

$$x_{t,i} | z_{t,i}, \boldsymbol{\phi} \sim f(x | \phi_{z_{t,i}})$$

Finally, we summarize the data generation process for HDP-HTM as follows.

(1) Sample the cluster parameter vector $\boldsymbol{\phi}$ from the base measure H. The number of the parameters is unknown *a priori*, but is determined by the data when a new cluster is needed.
(2) Sample the global cluster mixture vector $\boldsymbol{\beta}$ from $stick(\gamma)$.
(3) Generate hierarchical transition matrix $\boldsymbol{\Pi}_t$ at time t from $\mathcal{DP}(\lambda, \boldsymbol{\beta})$.
(4) At time t, compute the mean measure $\boldsymbol{\omega}_t$ for the global cluster set by $\boldsymbol{\beta}$ and $\boldsymbol{\Pi}_t$ according to Eq. 14.
(5) At time t, sample the local mixture proportion $\boldsymbol{\theta}_t$ by $\mathcal{DP}(\alpha, \boldsymbol{\omega}_t)$.
(6) At time t, sample the cluster indicator $z_{t,i}$ approximated from $Mult(\boldsymbol{\theta}_t)$ for data item $x_{t,i}$.
(7) At time t, sample data item $x_{t,i}$ from $f(x|\phi_{z_{t,i}})$ given cluster indicator $z_{t,i}$ and parameter vector $\boldsymbol{\phi}$.

6.2. Inference for HDP-HTM

We denote $n_{i\to j}^{t}$ as the number of the state transitions from states i to j between two adjacent times $t-1$ and t. Let $n_{t,k}$ be the number of the data items belonging to cluster k at time t, $n_{t,k}^{-i}$ be the number of the data items belonging to cluster k except $x_{t,i}$ at time t, and n_t be the number of all the data items at time t. Similar to HDP [Teh et al. 2007], let $m_{t,k}$ be the number of the tables (i.e., the local clusters) belonging to the global cluster k at time t, and m_k be the number of the tables (i.e., the local clusters) belonging to the global cluster k across all the times. Finally, let $\boldsymbol{x}_t$ be the data collection at time t.

In order to handle an infinite or arbitrary number of the states (i.e., clusters), we adopt the stick-breaking mechanism similar to what we have done in Section 5.3. Assume that there are K existing clusters. The global mixture proportion $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_K, \beta_u\}$ with β_u being the proportion for an unrepresented cluster; when a new cluster is instantiated, the vector $\boldsymbol{\beta}$ is updated according to the stick-breaking construction in Eq. 12 to ensure the summation equal to 1. In addition, the transition probability matrix is in the dimension of $K+1$ by $K+1$, resulting in $\boldsymbol{\omega}_t$ also in dimension of 1 by $K+1$ with the last element $\omega_{t,u}$ as the proportion of the unrepresented cluster.

The main sampling produre is similar to the direct assignment posterior sampling in HDP as our HDP-HTM is similar to HDP in the sense that $\boldsymbol{\omega}_t$ corresponds to $\boldsymbol{\pi}$ in Figure 1(b) and the difference is how we generate $\boldsymbol{\omega}_t$ from the hierarchical transition matrix and the global mixture proportions $\boldsymbol{\beta}$. The idea to sample $\boldsymbol{\beta}$ is also simliar to that in HDP; we first introduce an auxiliary variable $\boldsymbol{m}$ with $m_{t,k}$ as the number of the tables on cluster k at time t as mentioned above; then the global mixture proportion $\boldsymbol{\beta}$ is sampled from $\boldsymbol{m}$. We obtain the posterior of the transition probability matrix by the counter statistic $n^t_{i\to j}$ at time t according to Eq. 13 given the previous and the current state indicators. In the direct assignment of the posterior sampling scheme, we no longer need to sample $\boldsymbol{\theta}_t$ because we may just sample the cluster assignment $\boldsymbol{z}_t$ at time t by integrating out $\boldsymbol{\theta}_t$. Similarly, by the conjugacy of h and f, it is not necessary to sample parameter ϕ_k for cluster k.

Sampling $\boldsymbol{z}_t$

At time t, all data items and their state indicator assignments at this time are exchangeable; thus, the conditional probability of the current cluster assignment $z_{t,i}$ for the current data item $x_{t,i}$ given the other assignments $\boldsymbol{z}_{t,-i} = \boldsymbol{z}_t \setminus z_{t,i}$ and the Dirichlet process parameters $\boldsymbol{\omega}_t$ and α is:

$$p(z_{t,i} = k | \boldsymbol{z}_{t,-i}, \boldsymbol{\omega}_t, \alpha) = \begin{cases} \frac{n^{-i}_{t,k} + \alpha\omega_{t,k}}{n_t - 1 + \alpha} & \text{if } k \text{ is an existing cluster} \\ \frac{\alpha\omega_{t,u}}{n_t - 1 + \alpha} & \text{if } k \text{ is a new cluster} \end{cases} \tag{15}$$

After we have the observation of the data items, we compute the posterior conditional probability of $z_{t,i}$ given the other cluster assignment $\boldsymbol{z}_{t,-i}$, the observation $\boldsymbol{x}_t$ at time t, the parameters $\boldsymbol{\omega}_t$ and α and transition probability Π_t.

$$\begin{aligned} &p(z_{t,i} = k | \boldsymbol{z}_{t-1}, \Pi_t, \boldsymbol{z}_{t,-i}, \boldsymbol{x_t}, \boldsymbol{\omega}_t, \alpha) \propto p(z_{t,i} = k | x_{t,i}, \boldsymbol{z}_{t,-i}, \boldsymbol{\omega}_t, \alpha) p(x_{t,i} | \boldsymbol{x}^{-i}_k, \boldsymbol{z}_{t,-i}) \\ &= \begin{cases} \frac{n^{-i}_{t,k} + \alpha\omega_{t,k}}{n_t - 1 + \alpha} f^{-i}_k(x_{t,i}) & \text{if } k \text{ is an existing cluster} \\ \frac{\alpha\omega_{t,u}}{n_t - 1 + \alpha} f^{-i}_k(x_{t,i}) & \text{if } k \text{ is a new cluster} \end{cases} \end{aligned} \tag{16}$$

where $\boldsymbol{x}^{-i}_k = \{x_{t,j}, z_{t,j} = k, j \neq i\}$; $f^{-i}_k(x_{t,i})$ is the conditional likelihood of $x_{t,i}$ given the other data items $\boldsymbol{x}_{t,-i}$ under cluster k, which by the conjugacy property of h and f may be computed by integrating out the cluster parameter ϕ_k for cluster k similar to Eq. 7 or Eq. 8 with ϕ_k replaced with $\phi_{t,k}$. We may drop $\boldsymbol{z}_{t-1}$ and Π_t in the right hand side of Eq.16 as $z_{t,i}$ is approximated as a multinomial distribution only dependent on $\boldsymbol{\theta}_t$. Consequently, we do posterior Gibbs Sampling [Casella and George 1992] to infer the state indicator for data item $x_{t,i}$.

Estimate the Transition Matrix $\boldsymbol{\Pi}_t$

After we have the knowledge of the sequence of the states at adjacent times and the observations at different times, the state transition statistics $n^t_{i\to j}$ at time t is updated; we may estimate the MAP of transition probability matrix $\boldsymbol{\Pi}_t$ at time t according to Eq. 13.

Sampling $\boldsymbol{m}$

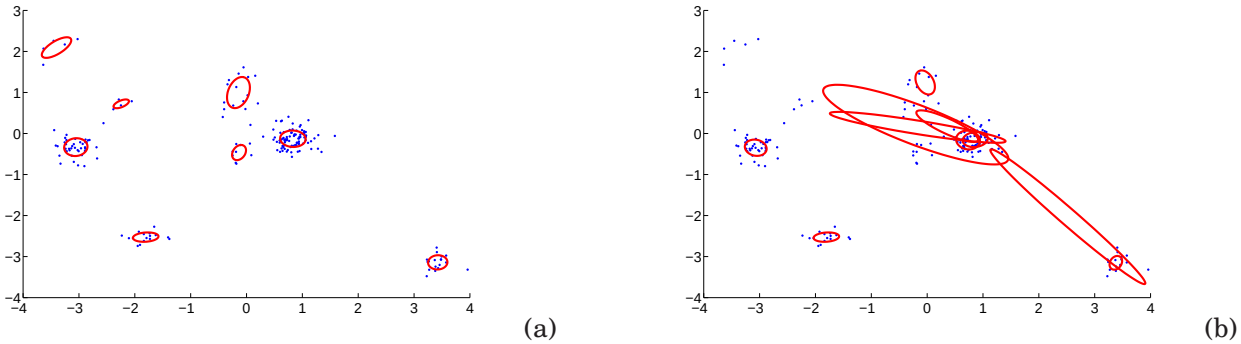

Fig. 5. Clustering results of HDP-HTM (a) and PCQ (b) for the synthetic data

Again similar to HDP, in order to sample $\boldsymbol{m}$, we must first sample $\boldsymbol{m}_t$, the number of the tables (i.e., the local clusters) for the global clusters at time t [Teh et al. 2007]. After the sampling of $\boldsymbol{z}_t$, $n_{t,k}$ is updated accordingly. By [Antoniak 1974; Teh et al. 2007], we may sample $\boldsymbol{m}$ as follows:

$$p(m_{t,k} = m|\boldsymbol{z}_t, \boldsymbol{m}^{-t,k}, \boldsymbol{\beta}, \alpha) = \frac{\Gamma(\alpha\beta_k)}{\Gamma(\alpha\beta_k + n_{t,k})} S(n_{t,k}, m)(\alpha\beta_k)^m \tag{17}$$

where $\boldsymbol{m}^{-t,k} = \boldsymbol{m} \setminus m_{t,k}$

Sampling $\boldsymbol{\beta}$

Given $\boldsymbol{m}$, the posterior distribution of $\boldsymbol{\beta}$ is:

$$\boldsymbol{\beta}|\boldsymbol{m}, \gamma \sim Dir(m_1, \ldots, m_K, \gamma) \tag{18}$$

where K is the number of the existing clusters up to time t. Consequently, it is trivial to sample $\boldsymbol{\beta}$ according to Eq. 18.

Hyperparameter Sampling

In the HDP-HTM model, there are the concentration hyperparameters $\Theta = \{\alpha, \gamma, \lambda\}$. According to [Teh et al. 2007; Escobar and West 1995], we may sample these parameters by the Gamma distribution with the constant Gamma parameters discussed in detail in Section 7.

Finally, we summarize the sampling framework at time t as follows:

(1) Initialize the transition matrix $\boldsymbol{\Pi}_t$, as well as $\boldsymbol{\beta}$, $\boldsymbol{m}$, and $\boldsymbol{z}_t$; compute $\boldsymbol{\omega}_t$ by taking the product of $\boldsymbol{\Pi}_t$ and $\boldsymbol{\beta}$.
(2) Sample the hyperparameters α, γ, and λ from the Γ (Gamma) distribution.
(3) Sample $\boldsymbol{m}$ based on $\boldsymbol{z}_t$, $\boldsymbol{\beta}$ and α according to Eq. 17.
(4) Sample $\boldsymbol{\beta}$ based on $\boldsymbol{m}$ and γ according to Eq. 18.
(5) Estimate $\boldsymbol{\Pi}_t$ at time t based on $\boldsymbol{z}_{t-1}$, $\boldsymbol{z}_t$, $\boldsymbol{\beta}$, λ and data collections $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_t$ according to Eq. 13.
(6) Sample $\boldsymbol{z}_t$ based on $\boldsymbol{\Pi}_t$, $\boldsymbol{\beta}$, and α according to Eq. 16.
(7) Iterate between 2 and 6 until convergence.

7. EXPERIMENTAL EVALUATIONS

We have evaluated our models DPChain and HDP-HTM extensively against the evolutionary spectral clustering algorithms PCM and PCQ [Chi et al. 2007] and HDP [Teh et al. 2007] for the synthetic data set and the real data sets in the application of document evolutionary clustering; for the experiments in text data evolutionary clustering,

we have also added LDA [Blei et al. 2003; Heinrich 2004] into the comparison. In particular, the evaluations are performed in three data sets, a synthetic data set, the 20 Newsgroups data set, and a Google daily news data set we have collected over a period of five continuous days. We report the evaluations both in performance and in running time for the real data sets. At the end of this section, we also report a case study of applying HDP-HTM to solving the dynamic social network community discovery and tracking problem.

7.1. Synthetic Dataset

We have generated a synthetic data set in a scenario of evolutionary development. The data set is a collection of mixture models with the number of the clusters unknown *a priori* with a smooth transition over the time during the evolution. Specifically, we simulate the scenario of the evolution over 10 different times with each time's collection according to a DPM model with 200 2-dimensional Gaussian distribution points. At each time, part of the clusters are chosen from the previous collections; other clusters are sampled from the multinomial distribution with mixture proportion vectors sampled from a symmetric Dirichlet process. 10 Gaussian points in $\mathbf{N}(\mathbf{0}, \mathbf{2I})$ are set as the 10 global clusters' mean parameters. Then 200 Gaussian points within a cluster are sampled with this cluster's mean parameter and deviation parameter sampled from $\mathbf{N}(\mathbf{0}, \mathbf{0.2I})$, where $\mathbf{I}$ is an identity matrix. After the generation of such a data set, we obtain the number of the clusters and the cluster assignments as the ground truth. We intentionally generate different numbers of the clusters at different times, as shown in Figure 7.

In the inference process, we tune the hyperparameters as follows. In each iteration, we use the vague Gamma priors [Escobar and West 1995] to update α, λ, and γ from $\Gamma(1, 1)$. Figure 5 shows an example of the clustering results between HDP-HTM and PCQ at time 8 for the synthetic data. Clearly, HDP-HTM has a much better performance than PCQ for this synthetic data set.

For a more systematic evaluation on this synthetic data set, we use the Normalized Mutual Information (NMI) [Strehl and Ghosh 2002] to quantitatively compare the clustering performances among all the five algorithms (DPChain, HDP-HTM, HDP, PCM, and PCQ). The reason why NMI is elected to use is that it is considered as one of the commonly used quantitative metrics for clustering in the literature. NMI measures how much information two random distribution variables (the computed clustering assignment and the groundtruth clustering assignment) share; the larger the better with 1 as the maximum normalized value. Figure 6 documents the performance comparison. From this figure, the average NMI values across the 10 times for HDP-HTM, HDP, and DPChain are 0.86, 0.78, and 0.74, respectively, while those for PCQ and PCM are 0.70 and 0.71, respectively. DPChain works worse than HDP-HTM for the synthetic data. The reason is that the DPChain model is unable to capture the cluster correspondence during the evolution among the data collections across the time while HDP-HTM is able to explicitly solve the correspondence problem; on the other hand, DPChain still performs better than PCQ and PCM in avearage.

Since one of the advantages of the HDP-HTM and DPChain models is the capability to learn the number of the clusters during the evolution, we report this performance for HDP-HTM and DPChain compared with HDP on this synthetic data set in Figure 7. Here, we define the expected number of the clusters at each time as the average number of the clusters in all the posterior sampling iterations after the burn-in period. Thus, these numbers are not necessarily integers. Clearly, all the three models are able to learn the cluster numbers, with HDP-HTM having a better performance than HDP since it is able to learn more accurate state transition information while DPChain is

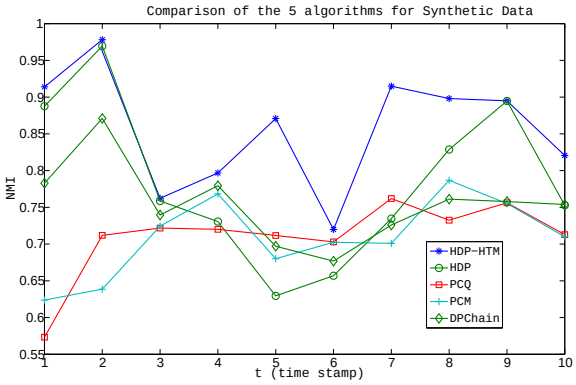

Fig. 6. The NMI performance comparison of the five algorithms on the synthetic data set

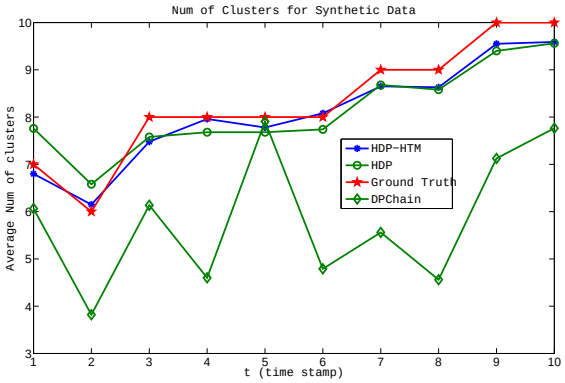

Fig. 7. The cluster number learning performance comparison on the synthetic data set

the worst as DPChain is unable to learn any cluster correspondence information. Since PCQ and PCM do not have this capability, they are not included in this evaluation.

7.2. Real Dataset

In order to showcase the performance of the DPChain and HDP-HTM models on real data applications, we apply them to a subset of the 20 Newsgroups data [2]. We intentionally set the number of the clusters at each time as the same number to accommodate the comparing algorithms PCQ and PCM which have this assumption of the same cluster number over the evolution. Also we select 10 clusters from the data set (alt.atheism, comp.graphics, rec.autos, rec.sport.baseball, sci.crypt, sci.electronics, sci.med, sci.space, soc.religion.christian, talk.politics.mideast), with each having 100 documents. To "simulate" the corresponding 5 different times, we then split the data set into 5 different collections, each of which has 20 documents randomly selected from each cluster. Thus, each collection at a time has 10 clusters to generate words. We have preprocessed all the documents with the standard text processing for removing the stop words and stemming the remaining words.

[2]http://kdd.ics.uci.edu/databases/20newsgroups/mini_newsgroups.tar.gz

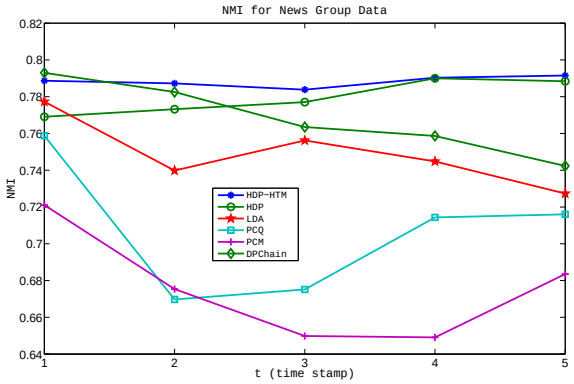

Fig. 8. The NMI performance comparison among the six algorithms on the 20 Newsgroups data set

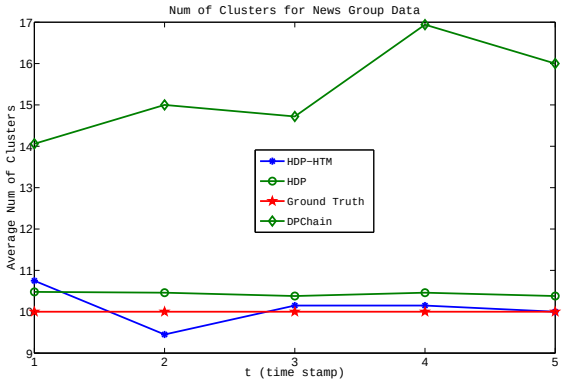

Fig. 9. Cluster number learning performance comparison on the 20 Newsgroups data set

To apply the HDP-HTM and DPChain models, a symmetric Dirichlet distribution is used with the parameter 0.5 for the prior base distribution H. In each iteration, we update α, γ, and λ in HDP-HTM, from the Gamma priors $\Gamma(0.1, 0.1)$. For LDA, α is set 0.1 and the prior distribution of the topics on the words is a symmetric Dirichlet distribution with concentration parameter 1. Since LDA only works for one data collection and requires a known cluster number in advance, we explicitly apply LDA to the data collection with the ground truth cluster number as input at each time.

Figure 8 reports the overall performance comparison among all the six methods using NMI metric again. Clearly HDP-HTM outperforms PCQ, PCM, DPChain, HDP, and LDA at almost all the times; DPChain is better than LDA, PCQ, and PCM; in particular, the difference is substantial for PCQ and PCM. Figure 9 further reports the performance on learning the cluster numbers at different times for HDP-HTM and DPChain in comparison with HDP. All the models have a reasonable performance in automatically learning the cluster number at each time in comparison with the ground truth, with HDP-HTM having the best performance in average and DPChain the worst, which is consistent with the cluster number learning result in the synthetic data set.

In order to truly demonstrate the performance of HDP-HTM in comparison with the state-of-the-art literature on a real evolutionary clustering scenario, we have manually collected Google News articles for a period of five continuous days with both the data items and the clusters evolving over the time. All the documents are selected from six categories of Google News including Business, Sci/Tech, Sports, Entertainment, Health and World[3]. The evolutionary ground truth for this data set is as follows. For each of the five continuous days, we have the number of the words, the number of the clusters, the number of the documents as (6113, 5, 50), (6356, 6, 60), (7063, 5, 50), (7762, 6, 60), and (8035, 6, 60), respectively. In order to accommodate the assumption of PCM and PCQ that the cluster number stays the same during the evolution, but at the same time in order to demonstrate the capability of HDP-HTM and DPChain to automatically learn the cluster number at each evolutionary time, we intentionally set the news cluster number at each day's collection to have a small variation deviation during the evolution. Again, in order to compare the text clustering capability of LDA [Blei et al. 2003; Heinrich 2004] with a known topic number in advance, we use the ground truth cluster number at each time as the input to LDA. The parameter tuning process is similar to that in the experiment using the 20 Newsgroups data set.

Figure 10 reports the NMI based performance evaluations among the six algorithms. Again, HDP-HTM outperforms PCQ, PCM, DPChain, HDP, and LDA at all the times, especially being substantially better than PCQ, PCM, and LDA. DPChain is better than PCQ and PCM. PCQ and PCM fail completely in most of the cases as they assume that the number of the clusters remains the same during the evolution, which is not true in this scenario.

Figure 11 further reports the performance on learning the cluster numbers for different times for HDP-HTM and DPChain compared with HDP model. Again HDP-HTM has the best performance to learn the cluster numbers automatically at all the times while DPChain is the worst, consistent with the previous experiments.

In Figure 12, we report the running time comparision for the three models (LDA, DPChain, and HDP-HTM) on the Google news and the 20 Newsgroups data sets. The two real data sets have different scales with a smaller scale for the Google news data set having about 300 documents and a larger scale for the 20 Newsgroups data set having about 1000 documents. The running times for the three models in the smaller Google news data set are comparable, with DPChain running slightly the fastest and HDP-HTM slightly the slowest. When the data set scales up to the 20 Newsgroups data set, the difference of the running times among the three models becomes clearly more obvious, with LDA the fastest and HDP-HTM the slowest. Furthermore, when the data set scales up from the Google news data set to the 20 Newsgroups data set, LDA has the smallest increase of the running time while HDP-HTM has the largest increase of the running time. This shows that from the scalability consideration, LDA is the best while HDP-HTM is the worst. This observation is consistent with the intuition. In consideration of the spectrum of the model complexity, LDA is the simplest resulting in the best scalability while HDP-HTM is the most complicated resulting in the worst scalability.

We have also conducted sensitivity analysis for DPChain and HDP-HTM models. In all the above experiments, in each iteration when we run the models, we sample the hyperparameters (i.e., α, γ, and λ) from the Gamma distribution with different values. We have observed that the experimental results are not sensitive to these hyperparameters.

[3]http://news.google.com/

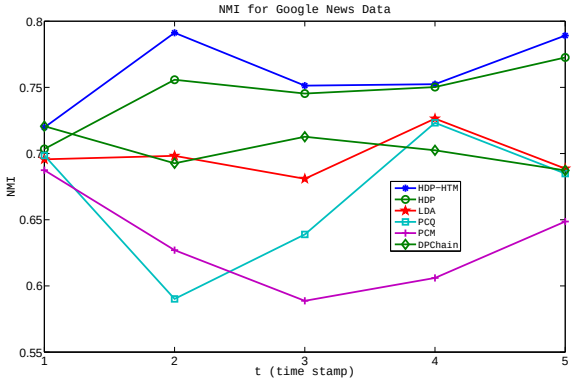

Fig. 10. The NMI performance comparison for all the six algorithms on the Google news data set

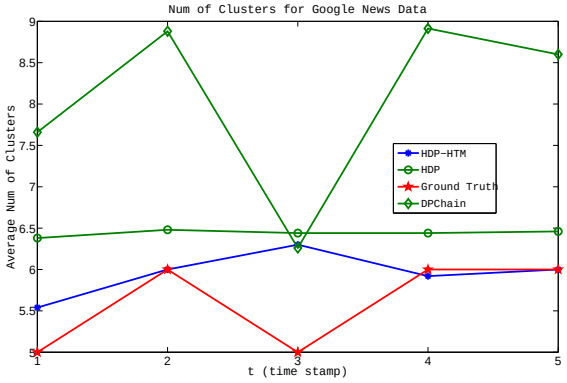

Fig. 11. The cluster number learning performance comparison on the Google news data set

7.3. A Case Study on Dynamic Social Network Analysis

From the statistical perspective, a cluster (community) is a collection of samples associated with a certain distribution with distinctive parameters. As an example of the wide spectrum of the applications of the two solutions, both DPChain and HDP-HTM are able to model the dynamic social network evolution for community discovery and tracking. In this section, we show a case study of applying HDP-HTM to solve for this problem.

At each time t, we have a vector ω_t representing the mean mixture proportions for each cluster (community) which is generated based on the historical community transition probability matrix and the top level prior mixture proportion β. Based on ω_t, we are able to sample mixture proportions θ_t for each community and consequently the hidden community indicator z_t can be generated. Each data item may not necessarily belong to only one community; it may join different communities according to the communities' distributions and the evolution. Thus, θ_t is a probability vector and indicates the likelihood that the data items would belong to the corresponding communities. Furthermore, θ_t also changes according to different historical community transition information under the evolution.

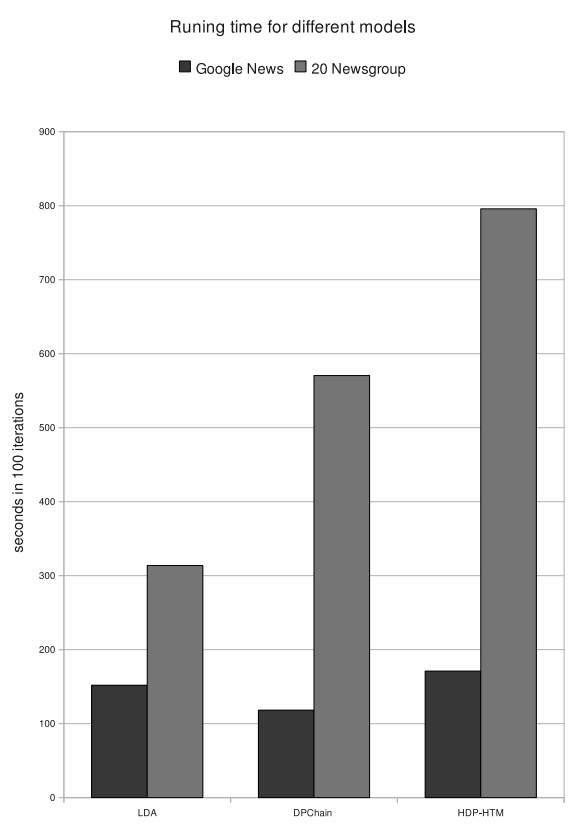

Fig. 12. Runing time comparison for LDA, DPChain, and HDP-HTM on the two real data sets

To illustrate a general scenario for the community discovery and tracking in dynamic social networks, we generate a synthetic data set similar to that reported in Section 7.1. Here, we have 4 time stamps; at each time there is a collection of a different number of the communities (represented as either ellipses if there are more than two points or lines if there are only two points in Figure 13) with different numbers of the social actors (points). The true numbers of the communities for all the four times are 6, 6, 7, and 7, respectively. HDP-HTM is able to learn correctly almost all the actual community numbers (5,6,6,7). Clearly, the community structures and the numbers evolve over the time. For example, it is interesting to note that the largest community at time 1 (Figure 13(a)) splits into two communities at time 2 (Figure 13(b)). From time 2 (Figure 13(b)) to time 3 (Figure 13(c)), the community above those two communities grows with the nearby social actors according to the new distribution. From time 3 (Figure 13(c)) to time 4 (Figure 13(d)), the number of social actors belonging to those two communities decreases as those two communities are becoming unpopular. This experiment further demonstrates that HDP-HTM is capable of discovering communities in dynamic social networks and at the same time of learning and tracking the community development during the evolution.

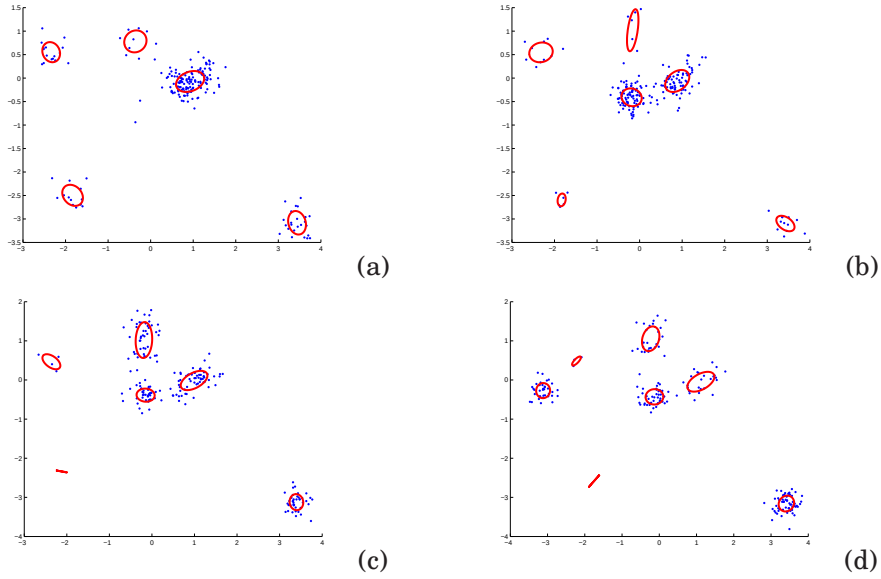

Fig. 13. Community Discovery by HDP-HTM (a) time 1 (b) time 2 (c) time 3 (d) time 4

8. CONCLUSIONS

In this paper, we have addressed the evolutionary clustering problem. Based on the recent literature on the Dirichlet process based models and HMM, we have developed the DPChain and HDP-HTM models as two effective nonparametric Bayesian learning solutions to this problem. Different from the traditional matrix decomposition based clustering solutions, both models substantially advance the evolutionary clustering literature in the sense that they not only perform better than the existing evolutionary clustering algorithms, but more importantly they are able to automatically learn the dynamic cluster numbers and the dynamic clustering structures during the evolution, which are typically expected in many real evolutionary clustering applications but are not available in the existing literature. In addition, HDP-HTM also explicitly addresses the correspondence issues whereas all the existing solutions do not. Extensive evaluations against the state-of-the-art literature demonstrate the effectiveness and the promise of the models. Furthermore, the HDP-HTM model is promising in the application on community discovery of dynamic social networks.

APPENDIX

Here, we derive the MAP (Maximum a posteriori) estimation of the transition probabilities Π_t at time t for Eq.13. For each time t, we only consider the state transition correspondence for two consecutive time stamps; the likelihood we are interested in are data items $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_t$ at the two adjacent times. The topic assignments $\boldsymbol{z}_{t-1}$ and $\boldsymbol{z}_t$ at these two times are hidden variables coupled with the data. The posterior of the transition probability Π_t is the product of the likelihood of the complete data (including the data items and the hidden topic assignments) and the prior of Π_t. As the number of the topics changes, it is assumed that we currently have K topics and the $K+1$st topic is unseen yet. This analysis is able to handle a countable number of the topics as we did for the transition probability by the stick-breaking process in Section 5.3. The

joint distribution of the data and the topics are:

$$P(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t, \boldsymbol{z}_{t-1}, \boldsymbol{z}_t|\Pi_t) = p(\boldsymbol{x}_t|\boldsymbol{z}_t)p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \Pi_t)p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{t-1})p(\boldsymbol{z}_{t-1}) \tag{19}$$

The posterior of Π_t is,

$$P(\Pi_t|\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \boldsymbol{z}_t, \boldsymbol{z}_{t-1}) = P(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t, \boldsymbol{z}_{t-1}, \boldsymbol{z}_t|\Pi_t)p(\Pi_t) \tag{20}$$

The objective here is to maximize the posterior of Π_t given the obversation $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$ and the missing (hidden) variables $\boldsymbol{z}_t$ and $\boldsymbol{z}_{t-1}$; consequently, the EM [Dempster et al. 1977], [Bishop 2007], [MacKay 1997] is the natural choice.

Here, the distribution of the data items given the topics at time t is,

$$p(\boldsymbol{x}_t|\boldsymbol{z}_t) = \prod_i f(x_{t,i}|\phi_{z_{t,i}})$$

Similarly for $p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{t-1})$.

The transition probability has the prior Dirichlet distribution (in finite case of DP)

$$p(\Pi_t) = \prod_{j=1}^{K+1}\prod_{k=1}^{K+1}(\pi_{j\to k}^t)^{\lambda\beta_k - 1}$$

The state transitions between times $t-1$ and t is multinomially distributed as

$$p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \Pi_t) = \prod_{j=1}^{K+1}\prod_{k=1}^{K+1}(\pi_{j\to k}^t)^{\sum_i \delta(z_{t-1,i}=j, z_{t,i}=k)}$$

Here $\delta(a,b) = 1$ iff $a = b$ and 0 otherwise, considering the transition of the states at the adjacent times only.

Finally, $p(\boldsymbol{z}_{t-1})$ is not dependent upon parameter Π_t. Therefore, we may rewrite the posterior of Π_t with the parts dependent upon Π_t as

$$P(\Pi_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_t, \boldsymbol{z}_{t-1}, \boldsymbol{z}_t) = const(\Pi_t)\prod_{j=1}^{K+1}\prod_{k=1}^{K+1}(\pi_{j\to k}^t)^{\lambda\beta_k - 1 + \sum_i \delta(z_{t-1,i}=j, z_{t,i}=k)} \tag{21}$$

where $const(\Pi_t)$ means that this term is independent of Π_t.

To maximize the penalized likelihood Eq. 21, we need to sum over the exponential state configurations for $\boldsymbol{z}_{t-1}$ and $\boldsymbol{z}_t$; therefore, we turn to EM [Dempster et al. 1977], [Bishop 2007], [MacKay 1997] to obtain the MAP estimation for Π_t. In the E step, we compute the posterior distribution of the latent variables $\boldsymbol{z}_t$ and $\boldsymbol{z}_{t-1}$ given the old parameters Π_t^{old} in the previous iteration. In the M step, we evaluate the expecation of the posterior of Π_t given the complete-data (including the latent variables) under the posterior distribution of the latent variables we have already obtained and to maximize this expectation defined as $Q(\Pi_t, \Pi_t^{old})$.

$$Q(\Pi_t, \Pi_t^{old}) = \sum_{\boldsymbol{z}_{t-1}, \boldsymbol{z}_t} p(\boldsymbol{z}_{t-1}, \boldsymbol{z}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_t, \Pi_t^{old}) \log P(\Pi_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_t, \boldsymbol{z}_{t-1}, \boldsymbol{z}_t) \tag{22}$$

Now we denote $\varepsilon(\boldsymbol{z}_{t-1}, \boldsymbol{z}_t|\Pi_t^{old}) = p(\boldsymbol{z}_{t-1}, \boldsymbol{z}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_t, \Pi_t^{old})$ as the joint posterior distribution of the states between times $t-1$ and t under the old parameters Π_t^{old}. The expection number of the state transitions from states j to k for all the state configura-

tions $\boldsymbol{z}_{t-1}$ and $\boldsymbol{z}_t$ is

$$n^t_{j\to k} = \sum_{\boldsymbol{z}_{t-1},\boldsymbol{z}_t} p(\boldsymbol{z}_{t-1},\boldsymbol{z}_t|\boldsymbol{x}_{t-1},\boldsymbol{x}_t,\Pi^{old}_t)\sum_i \delta(z_{t-1,i}=j,z_{t,i}=k)$$

$$= \mathbb{E}(\sum_i \delta(z_{t-1,i}=j,z_{t,i}=k)) \approx \frac{1}{N}\sum_{n=1}^{N}\sum_i \delta(z_{t-1,i}=j,z_{t,i}=k)$$

To compute the expectation under the posterior state configurations with the monte carlo approximation, we run N iterations to average the transition counts. We use this sufficient statistic to derive the MAP estimation of $\pi^t_{j\to k}$ in the M-Step from $Q(\Pi_t,\Pi^{old}_t)$ Eq. 22.

$$Q(\Pi_t,\Pi^{old}_t) = \sum_{\boldsymbol{z}_{t-1},\boldsymbol{z}_t} \varepsilon(\boldsymbol{z}_{t-1},\boldsymbol{z}_t|\Pi^{old}_t)\log(const(\Pi_t)))$$

$$+\sum_{\boldsymbol{z}_{t-1},\boldsymbol{z}_t} \varepsilon(\boldsymbol{z}_{t-1},\boldsymbol{z}_t|\Pi^{old}_t)\sum_j\sum_k(\lambda\beta_k - 1 + \sum_i \delta(z_{t-1,i}=j,z_{t,i}=k))\log\pi^t_{j\to k} \quad (23)$$

The first term in Eq. 23 is a constant w.r.t. Π_t under the distribution function with parameters Π^{old}_t; we simply denote it as $const$; the second term can be further derived as

$$\sum_j\sum_k(\lambda\beta_k - 1 + n^t_{j\to k})\log(\pi^t_{j\to k})$$

With a Lagrange multiplier for the constraint$\sum_k \pi^t_{j\to k}=1$, the maximization of $Q(\Pi_t,\Pi^{old}_t)$ can be represented as

$$\max_{\pi^t_{j\to k},\zeta_j} \sum_j\sum_k(\lambda\beta_k - 1 + n^t_{j\to k})log(\pi^t_{j\to k}) + \sum_j \zeta_j(\sum_k \pi^t_{j\to k} - 1)$$

$$\text{s.t.}\quad \zeta_j > 0 \quad \forall j \quad (24)$$

From the maximization Eq. 24, the solution of $\pi^t_{j\to k}$ for topic k already existing is

$$\pi^t_{j\to k} = \frac{n^t_{j\to k} - 1 + \lambda\beta_k}{n^t_j - K - 1 + \lambda} \quad (25)$$

For the new $(K+1)$st topic, The transition probability mass is

$$\pi^t_{j\to K+1} = \frac{\lambda\beta_u}{n^t_j - K - 1 + \lambda} \quad (26)$$

ACKNOWLEDGMENTS

This work is supported in part by NSF (IIS-0535162, IIS-0812114, CCF-1017828, IIS 0905215, IIS-0914934, DBI-0960443, OISE-0968341, CNS-1115234, and OIA-0963278), Google Mobile 2014 Program, and National Basic Research Program of China (973 Program)(2012CB316406). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

ABOU-RJEILI, A. AND KARYPIS, G. 2006. Multilevel algorithms for partitioning power-law graphs. *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*.

AHMED, A. AND XING, E. P. 2009. Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences 106,* 29, 11878–11883.

ALDOUS, D. 1983. Exchangeability and related topics. *Ecole de Probabilites de Saint-Flour* XIII, 1–198.

ANTONIAK, C. 1974. Mixtures of Dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics 2,* 6, 1152–1174.

BEAL, M. J., GHAHRAMANI, Z., AND RASMUSSEN, C. E. 2002. The Infinite Hidden Markov Model. In *NIPS 14*.

BISHOP, C. M. 2007. *Pattern Recognition and Machine Learning*. Springer.

BLACKWELL, D. AND MACQUEEN, J. 1973. Ferguson distributions via plya urn schemes. *The Annals of Statistics 1,* 2, 353–355.

BLEI, D. AND LAFFERTY, J. 2006. Dynamic topic models. In *In Proceedings of the 23rd International Conference on Machine Learning*.

BLEI, D., NG, A., AND JORDAN, M. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research 3*, 993–1022.

BOYD-GRABER, J. AND BLEI, D. M. 2008. Syntactic topic models. In *Neural Information Processing Systems*.

CASELLA, G. AND GEORGE, E. I. 1992. Explaining the Gibbs sampler. *The American Statistician 46,* 3, 167–174.

CHAKRABARTI, D., KUMAR, R., AND TOMKINS, A. 2006. Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 554–560.

CHI, Y., SONG, X., ZHOU, D., HINO, K., AND TSENG, B. L. 2007. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 153–162.

DEMPSTER, A., LAIRD, N., AND RUBIN, D. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society 39,* 1, 1–38.

ESCOBAR, M. D. AND WEST, M. 1995. Bayesian density estimation and inference using mixtures. *The Annals of Statistics 90*, 577–588.

FERGUSON, T. S. 1973. A bayesian analysis of some nonparametric problems. *The Annals of Statistics 1,* 2, 209–230.

FLAKE, G. W., LAWRENCE, S., AND GILES, C. L. 2000. Efficient identification of web communities. In *In Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 150–160.

FORTUNATO, S. 2010. Community detection in graphs. `http://arxiv.org/abs/0906.0612`.

FOX, E. B., SUDDERTH, E. B., JORDAN, M. I., AND WILLSKY, A. S. 2007. Developing a tempered HDP-HMM for systems with state persistence. *Technical Report*.

GAEL, J. V., SAATCI, Y., TEH, Y. W., AND GHAHRAMANI, Z. 2008. Beam sampling for the Infinite Hidden Markov model. In *25th International Conference on Machine Learning*.

GRIFFITHS, T. L. AND STEYVERS, M. 2004. Finding scientific topics. In *Proceedings of the National Academy of Sciences*. Feb., 5228–5235.

GRIFFITHS, T. L., STEYVERS, M., BLEI, D. M., AND TENENBAUM, J. B. 2005. Integrating topics and syntax. In *Advances in Neural Information Processing Systems 17*. 537–544.

GRUBER, A., ROSEN-ZVI, M., AND WEISS, Y. 2007. Hidden topic Markov models. In *Artificial Intelligence and Statistics*.

HEINRICH, G. 2004. Parameter estimation for text analysis. *Technical Report*.

ISHWARAN, H. AND JAMES, L. F. 2001. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association 96,* 453, 161–.

MACKAY, D. J. 1997. Ensemble learning for hidden Markov models. *Technical Report*.

MCCALLUM, A., CORRADA-EMMANUEL, A., AND WANG, X. 2005. Topic and role discovery in social networks. In *Proceedings of 19th International Joint Conference on Artificial Intelligence*. 786–791.

NEAL, R. M. 1993. Probabilistic inference using Markov Chain Monte Carlo methods. *Technical Report* CRG-TR-93-1.

NEAL, R. M. 2000. Markov Chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics 9,* 2, 249–265.

NG, A. Y., JORDAN, M. I., AND WEISS, Y. 2002. On spectral clustering: Analysis and an algorithm. In *NIPS 14*.

NI, K., CARIN, L., AND DUNSON, D. 2007. Multi-task learning for sequential data via iHMMs and the nested Dirichlet process. In *ICML*. 689–696.

PATHAK, N., DELONG, C., BANERJEE, A., AND ERICKSON, K. 2008. Social topic models for community extraction. In *In 2nd ACM Workshop on Social Network Mining and Analysis (SNA-KDD 2008)*. ACM.

RABINER, L. 1989. A tutorial on hidden Markov models and selected applications inspeech recognition. In *Proceedings of the IEEE*. 257–286.

ROSEN-ZVI, M., GRIFFITHS, T., STEYVERS, M., AND SMYTH, P. 2004. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 487–494.

SETHURAMAN, J. 1994. A constructive definition of Dirichlet priors. *Statistica Sinica 4*, 639–650.

SHI, J. AND MALIK, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence 22,* 8.

STREHL, A. AND GHOSH, J. 2002. Cluster ensembles - a knowledge reuse framework for combining partitionings. In *Proceedings of AAAI*.

TEH, Y., M. JORDAN, M. B., AND BLEI, D. 2007. Hierarchical Dirichlet processes. *Journal of the American Statistical Association 101,* 476, 1566–1581.

TEH, Y. W. 2007. Dirichlet processes. In *Encyclopedia of Machine Learning*.

WALLACH, H. M. 2006. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*. 977 – 984.

WANG, C., BLEI, D., AND HECKERMAN, D. 2008. Continuous time dynamic topic models. In *In Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI)*.

WANG, X. AND MCCALLUM, A. 2006. Topics over time: A non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 424–433.

XU, T., ZHANG, Z., YU, P., AND LONG, B. 2008a. Dirichlet process based evolutionary clustering. In *ICDM*.

XU, T., ZHANG, Z., YU, P., AND LONG, B. 2008b. Evolutionary clustering by hierarchical Dirichlet process with hidden Markov state. In *ICDM*.

XU, Z., TRESP, V., YU, S., AND YU, K. 2008. Nonparametric relational learning for social network analysis. In *In 2nd ACM Workshop on Social Network Mining and Analysis (SNA-KDD 2008)*. ACM.

ZHANG, H., GILES, C. L., FOLEY, H. C., AND YEN, J. 2007. Probabilistic community discovery using hierarchical latent gaussian mixture model. In *In AAAI*. 663–668.

ZHANG, H., LI, W., WANG, X., GILES, C. L., FOLEY, H. C., AND YEN, J. 2007. HSN-PAM: Finding hierarchical probabilistic groups from large-scale networks. In *ICDM Workshops 2007. Seventh IEEE International Conference on Data Mining*. 27–32.

ZHANG, H., QIU, B., GILES, C. L., FOLEY, H. C., AND YEN, J. 2007. An lda-based community structure discovery approach for large-scale social networks. In *In IEEE International Conference on Intelligence and Security Informatics*. 200–207.

ZHU, X., GHAHRAMANI, Z., AND LAFFERTY, J. 2005. Time-sensitive Dirichlet process mixture models. *Technical Report* CMU-CALD-05-104.

Received July 2009; revised July 2010, December 2010; accepted August 2011