Dirichlet Process Based Evolutionary Clustering

Tianbing Xu[1] Zhongfei (Mark) Zhang[1]
[1]Dept. of Computer Science
State Univ. of New York at Binghamton
Binghamton, NY 13902, USA
{txu,zhongfei,blong}@cs.binghamton.edu

Philip S. Yu[2] Bo Long[1]
[2]Dept. of Computer Science
Univ. of Illinois at Chicago
Chicago, IL 60607, USA
psyu@cs.uic.edu

Abstract

Evolutionary Clustering has emerged as an important research topic in recent literature of data mining, and solutions to this problem have found a wide spectrum of applications, particularly in social network analysis. In this paper, based on the recent literature on Dirichlet processes, we have developed two different and specific models as solutions to this problem: DPChain and HDP-EVO. 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 structures during the evolution. Extensive evaluations have demonstrated the effectiveness and promise of these models against the state-of-the-art literature.

1 Introduction

Evolutionary clustering is a relatively new 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 in comparison with the traditional clustering. On the other hand, solutions to the evolutionary clustering problem have found a wide spectrum of applications for trend development analysis, social network evolution analysis, and dynamic community development analysis. Potential and existing applications include daily news analysis to observe news focus change, blog analysis to observe community development, and scientific publications analysis to identify the new and hot research directions in a specific area. Due to these important applications, evolutionary clustering has recently become a very hot and focused research topic.

Statistically, each cluster is associated with a certain distribution at each time. A solution to the evolutionary clustering problem is to make an inference to a sequence of distributions from the data at different times.

A reasonable solution to the evolutionary clustering problem must have a clustering result consistent with the original data distribution. Consequently, the following two properties must be satisfied to reflect a reasonable 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.

In this paper, we propose a statistical approach to solving the evolutionary clustering problem. We assume that the cluster structure at each time follows a mixture model of the clusters for the data collection at this time; clusters at different times may share common clusters; 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 clusters change over the time. Consequently, we use Dirichlet Process (DP) [11] to model the evolutionary change of the clusters over the time. Specifically, we propose two Dirichlet process based models as two different solutions to the evolutionary clustering problem: DPChain and HDP-EVO.

DPChain is based on the Dirichlet Process Mixture (DPM) model [2, 10], which automatically learns the number of the clusters from the evolutionary data; in addition, the cluster mixture proportion information at different times is used to reflect a smooth cluster change over the time. HDP-EVO is developed based on the Hierarchical Dirichlet Process (HDP) model [21] with a set of common clusters on the top level of the hierarchy to explicitly address the cluster correspondence issue in order to solve the evolutionary

clustering problem; the middle level is for the clusters at each different time, which are considered as the subsets of the top level clusters; the relationship between the top level clusters and the middle level clusters is obtained through the statistical inference under this model, resulting in explicitly addressing the cluster correspondence issue for the clusters at different times.

The specific contributions of this work are highlighted as follows: (1) We have applied the recent literature on Dirichlet process based statistical learning to solve the evolutionary clustering problem by developing two specific models: DPChain and HDP-EVO as two different solutions. (2) Both models for evolutionary clustering substantially advance the literature in the sense that they are capable of automatically learning the number of clusters and the cluster structure at each time during the evolution, which makes these solutions practical in many evolutionary clustering applications. (3) We have demonstrated the superiority of these solutions to the existing state-of-the-art literature in both synthetic data and real Web daily news data for the evolutionary document clustering application.

2 Related Work

Evolutionary Clustering is a recently emerging research topic in data mining. Due to its very short history, there is not much literature on this topic at this time.

Chakrabarti et al. in 2006 [7] 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. The framework attempted to combine the two properties of evolutionary clustering for the development of these two algorithms; one is the snapshot quality, which measures how well the current data fit the current clustering; and other is the history quality, which measures how smooth the current clustering is with the previous clustering.

Recently, Chi et al. [8] 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 algorithms were developed by explicitly incorporating the history clustering information into the existing classic clustering algorithm, specifically, k-means, agglomerative hierarchical clustering, and spectral clustering approaches [16, 19]. 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 obviously violated.

Dirichlet Process [11] is a statistical model developed in the statistics literature to capture the distribution uncertainties in the space of probability measure. When a random measure is no longer a single distribution, but a mixture distribution, DPM [2, 10] is used to extend DP. Statistically, the clustering problem indeed fits into a mixture model, making it natural to use DPM model.

More importantly, DPM allows an infinite number of mixture components, shedding the light on solving the clustering model selection problem. Sethuraman [18] gives a constructive definition of Dirichlet distribution for an arbitrarily measurable base space. This stick-breaking construction is very useful to model the weight of mixture components in the clustering mixture model. Besides the capability of learning the number of clusters from the data automatically, HDP model [21] is further developed for sharing the mixture components across different data collections, making it possible to capture the relationship between the clusters at different times.

Recently in the machine learning community, DP related models are developed and used to solve the clustering problems such as document topic analysis [5, 21], image clustering [17], and video surveillance activity analysis [22]. Blei et al. [5] developed the Latent Dirichlet Allocation (LDA) model that automatically learns the clustering of topics given a document corpus. However, LDA assumes that the number of clusters is given in advance and is a parametric constant.

Blei et al. [4] designed a family of time series probabilistic models to analyze the evolving topics at different times; they assumed a fixed number of topics and did not consider the clusters' birth or death during the evolution. Griffiths et al. [12] studied the PANS proceedings by LDA model to identify "hot topics" and "cold topics" by examining temporal dynamics of the documents; they used Bayesian model selection to estimate the number of the topics. Wang et al. [23] presented an LDA-style topic model in which time is an observed continuous variable instead of a Markov discretization assumption. This model is able to capture the trends of the temporal topic evolution; however, the number of topics is still assumed fixed. Zhu et al. [25] further developed a time-sensitive Dirichlet process mixture model for clustering documents, which models the temporal correlations between instances. Nevertheless, a strong assumption was made that there is only one cluster and one document at each time, which is too restrictive to handle prob-

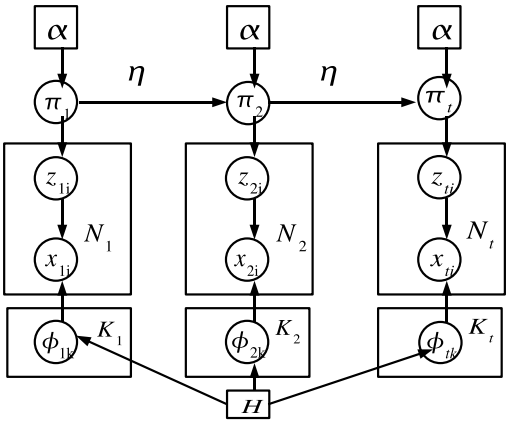


Figure 1. The DPChain Model

lems with a collection of clusters and documents at a time. More recently, Xu et al. [24] proposed a statistical model HDP-HMM to provide a solution to evolutionary clusterng, which is able to learn the number of clusters and the cluster structure transitions during the evolution.

3 Dirichlet Process Mixture Chain (DPChain)

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 to a defined variable means the whole scope the variable is defined for except for the item indicated as s.

The first model we propose is based on the DPM model [2, 10], 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.

3.1 DPChain Representation

Figure 1 illustrates the DPChain model. We use the indicator variable to represent the DPChain model. First we introduce the notations. α denotes the concentration parameter for a Dirichlet distribution. H denotes the base measure of a Dirichlet distribution with the pdf as h. F denotes the distribution of the data with the pdf as 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 mixutre 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 $stick(\alpha)$ [18] which is described as follows.

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

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 is smoothly changed 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 is an indicator for the population of a cluster. 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 $\boldsymbol{\pi}_t$. With a cluster smooth change, more recent history has more influence on the current clustering than less recent history. Thus, a cluster with a higher mixture proportion at the present 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.

$$\boldsymbol{\pi}_t = \sum_{\tau=1}^{t} \exp\{-\eta(t-\tau)\}\boldsymbol{\pi}_\tau \tag{2}$$

where η is a smooth parameter.

This relationship is further illustrated by an extended Chinese Restaurant Process (CRP) [3, 1]. We denote $n_{t,k}$ as the number of data items in cluster k at time t, and $n_{t,k}^{-i}$ as the number of data items belonging to cluster k except $x_{t,i}$; $w_{t,k}$ is the smooth prior weight for cluster k at the beginning of time t. According to (2), $w_{t,k}$ has the relationship to $n_{\tau,k}$ at the previous time τ:

$$w_{t,k} = \sum_{\tau=1}^{t-1} \exp\{-\eta(t-\tau)\}n_{\tau,k} \tag{3}$$

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

$$p(z_{t,i} = k|\mathbf{z}_1, \dots \mathbf{z}_{t-1}, \mathbf{z}_{t,-i}) \propto \begin{cases} \frac{w_{t,k}+n_{t,k}^{-i}}{\alpha+\sum_{j=1}^{K_t} w_{t,j}+n_t-1} & \text{if } k \text{ is an existing cluster} \\ \frac{\alpha}{\alpha+\sum_{j=1}^{K_t} w_{t,j}+n_t-1} & \text{if } k \text{ is a new cluster} \end{cases} \tag{4}$$

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 (4), an existing cluster appears again with a probability proportional to $w_{t,k} + n_{t,k}^{-i}$, while a new cluster appears at 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 next time, which leads to a higher probability of becoming death for cluster k. Consequently, this model has the capability to describe the birth or death of a cluster over the evolution. The data item generation process for 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 cluster 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 (2).
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 cluster index variable $z_{t,i}$ and 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 modelled as a random variable varying with time, as the rate of generating a new cluster may change over the time.

3.2 DPChain Inference

Given the DPChain model, we use Markov Chain Monte Carlo (MCMC) method [14] to sample the cluster assignment $z_{t,i}$ for each data item at time t. Specifically, following Gibbs Sampling [6], 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 all the data at time t except for $x_{t,i}$. The posterior of the cluster assignment is determined by Bayes rule:

$$p(z_{t,i} = k|\mathbf{x}_t, \mathbf{z}_{t,-i}, \mathbf{z}_1, \dots \mathbf{z}_{t-1}) \propto p(x_{t,i}|\mathbf{z}_{t,-i}, \mathbf{z}_1, \dots \mathbf{z}_{t-1}, {\mathbf{x}_k}^{-i}) p(z_{t,i} = k|\mathbf{z}_1, \dots \mathbf{z}_{t-1}, \mathbf{z}_{t,-i}) \tag{5}$$

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 indenpent 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 of the right hand side of (5). Further, denote $f_k^{-i}(x_{t,i})$ as the first term of the right hand side of (5), 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:

$$f_k^{-i}(x_{t,i}) = \int f(x_{t,i}|\phi_{t,k}) \cdot h(\phi_{t,k}|\{x_{t,j} : z_{t,j} = k, j \neq i\}) d\phi_{t,k} \tag{6}$$

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 [15].

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

$$f_k^{-i}(x_{t,i}) = \int f(x_{t,i}|\phi_{t,k}) dH(\phi_{t,k}) \tag{7}$$

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

$$p(z_{t,i} = k|\mathbf{x}_t, \mathbf{z}_{t,-i}, \mathbf{z}_1, \dots \mathbf{z}_{t-1}) \propto \begin{cases} \frac{w_{t,k} + n_{t,k}^{-i}}{\alpha + \sum_{j=1}^{K_t} w_{t,j} + n_t - 1} f_k^{-i}(x_{t,i}) & \text{if } k \text{ is an existing cluster} \\ \frac{\alpha}{\alpha + \sum_{j=1}^{K_t} w_{t,j} + n_t - 1} f_k^{-i}(x_{t,i}) & \text{if } k \text{ is a new cluster} \end{cases} \tag{8}$$

4 HDP Evolutionary Clustering Model (HDP-EVO)

While DPChain model advances the existing literature on evolutionary clustering in the sense that it is capable of learning the cluster numbers over the time, this model fails to have an explicit representation on the cluster correspondence over the time. In order to explicitly capture the cluster correspondence between the data collections of different times, we further develop the HDP Evolutionary Clustering model, which we call HDP-EVO.

4.1 HDP-EVO Representation

HDP-EVO model is illustrated in Figure 2. Again, we use the indicator variable representation to describe the HDP-EVO model. First, we introduce the notations. γ is the concentration parameter of the Dirichlet distribution of

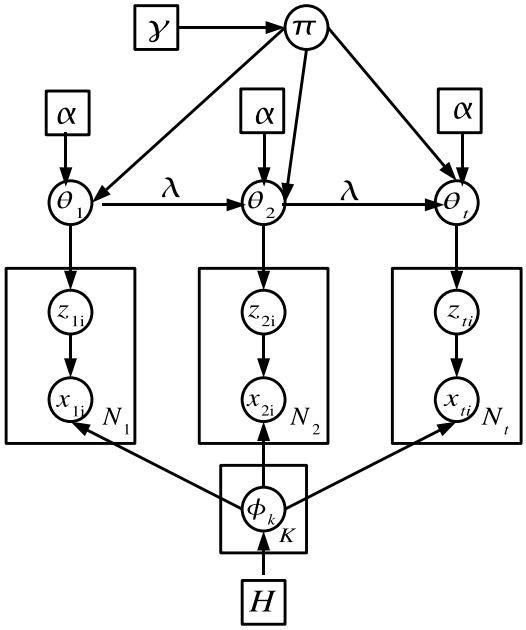


Figure 2. The HDP-EVO Model

$\boldsymbol{\pi}$. Common clusters for all the collections at different times are shared with the global cluster set with mixture proportion vector $\boldsymbol{\pi}$.

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

ϕ_k is the parameter for a cluster with i.i.d. sampled from a distribution H.

$$\phi_k|H \sim H$$

The clusters appearing at time t are a subset of the common clusters with a local cluster mixture parameter vector $\boldsymbol{\theta}_t$.

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

where α is the concentration parameter. At different times, a different $\boldsymbol{\theta}_t$ shares the common global clusters which establish the correspondence between the local clusters at different times and the global clusters.

Similar to DPChain model, the mixture proportion of the clusters evolves over the time, favoring recent history. We assume again an exponential smooth transition:

$$\boldsymbol{\theta}_t = \sum_{\tau=1}^{t} \exp\{-\lambda(t-\tau)\}\boldsymbol{\theta}_\tau \tag{9}$$

where λ is a smooth parameter. We denote $z_{t,i}$ as the cluster assignment at time t for the data item $x_{t,i}$, and follows a multinomial distribution of $\boldsymbol{\theta}_t$.

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

Finally, $x_{t,i}$ is modeled as being drawn from the distribution F with the parameter ϕ_k under cluster k.

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

Now, the data generation process is described as follows.

1. The common global clusters' parameter vector $\boldsymbol{\phi}$ is sampled from distribution H. The number of the cluster is not a fixed prior but is decided by the data when a new cluster is needed.

2. Sample global cluster mixture proportion $\boldsymbol{\pi}$ from $stick(\gamma)$.

3. At time t, first sample the local clusters' mixture proportion vector $\boldsymbol{\theta}_t$ from $DP(\alpha, \boldsymbol{\pi})$; then do smoothly weighted sum according to (9).

4. $z_{t,i}$, the assignment of the cluster for $x_{t,i}$, is sampled from the multinomial distribution with parameter $\boldsymbol{\theta}_t$.

5. Finally, we sample $x_{t,i}$ from distribution F with parameter ϕ_k, given the cluster assignment $z_{t,i} = k$.

Based on the above generation process, the cluster number can be automatically learned through the inference from the data at each time. All the local clusters at different times are capable of establishing a correspondence relationship among themselves from the top level of the commonly shared global clusters. With the introduction of the exponentially weighted smoothness of the mixture proportion vector at different times, the cluster may smoothly evolve over the time.

4.2 Two-Level CRP for HDP-EVO

The indicator variable representation of HDP-EVO directly assigns clusters to data. In order to design the Gibbs sampling process for HDP-EVO, we further illustrate HDP-EVO model as a 2-level CRP.

Under the standard CRP model [3, 1], each table corresponds to one cluster. Here, we further categorize the clusters into a higher level, global clusters that are commonly shared across all data collections at different times, and the lower lever, local clusters, i.e., the tables of a Chinese Restaurant with data items sitting around, at each time. We use k to denote the k-th global cluster and use tab to denote the tab-th local cluster (Figure 3).

At each time t, the data collection is modeled as being generated from the local clusters with the parameters $\{\psi_{t,1}, \ldots, \psi_{t,tab}, \ldots\}$, each of which is sampled from the commonly shared global clusters with parameters $\{\phi_1, \ldots, \phi_k, \ldots\}$ in the CRP style [3, 1]. We use $tab_{t,i}$ to denote the table (i.e., the local cluster) at time t for $x_{t,i}$. We assign global cluster k to table tab, if all the data clustered

into local cluster tab at time t are distributed with parameter ϕ_k. We explicitly introduce $k_{t,tab}$ to represent this mapping relationship. Similarly, we introduce $tab_{t,i}$ to denote the mapping that $x_{t,i}$ is clustered into table tab at time t. Let $n_{t,tab}$ be the number of the data items at table tab at time t, $n_{t,tab}^{-i}$ be the number of the data items in table tab except for $x_{t,i}$, and n_t be the total number of the data items at time t. Let $m_{t,k}$ be the number of the tables at time t belonging to the global cluster k, $m_{t,k}^{-tab}$ be number of the tables in cluster k except for tab, and m_t be the total number of the tables at time t,

Under the 2-level CRP, at time t, we first sample which table tab $x_{t,i}$ belongs to, given the history $\{tab_{t,1}, \ldots, tab_{t,i-1}\}$ in which by the exchangeability $x_{t,i}$ may be considered as the last data item at time t:

$$p(tab_{t,i}|tab_{t,1}, \ldots, tab_{t,i-1}) \propto \begin{cases} \frac{n_{t,tab}^{-i}}{\alpha+n_t-1} & \text{if } tab \text{ is an existing table} \\ \frac{\alpha}{\alpha+n_t-1} & \text{if } tab \text{ is a new table} \end{cases} \quad (10)$$

where α is the concentration parameter. To ensure the smooth transition over the history, we also denote $w_{t,k}$ as the smooth prior weight for cluster k at time t. Thus, we have

$$w_{t,k} = \sum_{\tau=1}^{t-1} \exp\{-\lambda(t-\tau)\} m_{\tau,k} \quad (11)$$

Denoting $\mathbf{K}$ as the all the history global cluster assignment mapping up to time t inclusive, the likelihood of having the assignment mapping $k_{t,tab}$ is:

$$p(k_{t,tab}|\mathbf{K} \setminus k_{t,tab}) \propto \begin{cases} \frac{m_{t,k}^{-tab}+w_{t,k}}{\gamma+m_t-1+\sum_{j=1}^{K_t} w_{t,j}} & \text{if } k \text{ is an existing cluster} \\ \frac{\gamma}{\gamma+m_t-1+\sum_{j=1}^{K_t} w_{t,j}} & \text{if } k \text{ is a new cluster} \end{cases} \quad (12)$$

where γ is the concentration parameter.

4.3 HDP-EVO Inference

Again we use Gibbs Sampling [6] for the 2-level CRP for HDP-EVO inference. First, we specify how to assign $x_{t,i}$ (which may be considered as the last data item by the exchangability) to tab:

$$p(tab_{t,i}|\mathbf{x}_t, tab_{t,1}, \ldots, tab_{t,i-1}, \mathbf{K}) \propto p(tab_{t,i}|tab_{t,1}, \ldots, tab_{t,i-1}) p(x_{t,i}|x_{t,-i}, tab_{t,1}, \ldots, tab_{t,i-1}, \mathbf{K}) \quad (13)$$

For the second level CRP, We denote the conditional likelihood $f_{k_{t,tab}}^{-i}(x_{t,i})$ as the second term of the right hand side of (13).

For an existing table tab which belongs to global cluster $k_{t,tab}$, the conditional likelihood of $x_{t,i}$ given other data under cluster $k_{t,tab}$ indexed from tab,$f_{k_{t,tab}}^{-i}(x_{t,i})$ is the same as that is Eq. (6) with cluster k replaced with $k_{t,tab}$.

For a new table tab, we first sample the table from the global cluster, the conditional likelihood of $x_{t,i}$ under the cluster k becomes:

$$f_{k_{t,tab}}^{-i}(x_{t,i}) = \begin{cases} \frac{m_{t,k}^{-tab}+w_{t,k}}{\gamma+m_t-1+\sum_{j=1}^{K_t} w_{t,j}} f_k^{-i}(x_{t,i}) & k \text{ is an existing cluster} \\ \frac{\gamma}{\gamma+m_t-1+\sum_{j=1}^{K_t} w_{t,j}} f_k^{-i}(x_{t,i}) & k \text{ is a new cluster} \end{cases} \quad (14)$$

where $f_k^{-i}(x_{t,i})$ under new cluster k is the marginal likelihood for a new global cluster k from Eq. (7) with $\phi_{t,k}$ replaced with ϕ_k.

Finally, we sample $x_{t,i}$ from table tab as follows:

$$p(tab_{t,i}|\mathbf{x}_t, tab_{t,1}, \ldots, tab_{t,i-1}, \mathbf{K}) \propto \begin{cases} \frac{n_{t,tab}^{-i}}{\alpha+n_t-1} f_{k_{t,tab}}^{-i}(x_{t,i}) & \text{if } tab \text{ is an existing table} \\ \frac{\alpha}{\alpha+n_t-1} f_{k_{t,tab}}^{-i}(x_{t,i}) & \text{if } tab \text{ is a new table} \end{cases} \quad (15)$$

Similarly, to sample a table tab from a global cluster k, we have:

$$p(k_{t,tab}|\mathbf{x}_t, tab_{t,1}, \ldots, tab_{t,i}, \mathbf{K} \setminus k_{t,tab}) \propto p(k_{t,tab}|\mathbf{K} \setminus k_{t,tab}) p(\mathbf{x}_{t,tab}|\mathbf{x}_{t,-tab}, k_{t,tab}, \mathbf{K} \setminus k_{t,tab}) \quad (16)$$

where $\mathbf{x}_{t,tab}$ denotes all the data belonging to table tab at time t, and $\mathbf{x}_{t,-tab} = \mathbf{x}_t \setminus \mathbf{x}_{t,tab}$ denotes the remaining data except those in table tab.

We denote the second term of the right hand side of (16) as $f_k^{-tab}(\mathbf{x}_{t,tab})$, which means the conditional likelihood of all the data in table tab, given other tables' data at time t, under cluster k.

For an existing global and new cluster k, we have the likelihood :

$$f_k^{-tab}(\mathbf{x}_{t,tab}) = \prod_{i:x_{t,i} \in tab} f_k^{-i}(x_{t,i}) \quad (17)$$

Finally, we assign a table tab to a global cluster k as follows:

$$p(k_{t,tab}|\mathbf{x}_t, \mathbf{tab}_t, \mathbf{K} \setminus k_{t,tab}) \propto \begin{cases} \frac{m_{t,k}^{-tab}+w_{t,k}}{\gamma+m_t-1+\sum_{j=1}^{K_t} w_{t,j}} f_k^{-tab}(\mathbf{x}_{t,tab}) & \text{if } k \text{ is an existing cluster} \\ \frac{\gamma}{\gamma+m_t-1+\sum_{j=1}^{K_t} w_{t,j}} f_k^{-tab}(\mathbf{x}_{t,tab}) & \text{if } k \text{ is a new cluster} \end{cases} \quad (18)$$

where $\mathbf{tab}_t$ is the set of all the tables at time t.

5 Parameter Learning

For both models we have developed in this paper, there are hyperparameters that must be estimated. We use the EM method [9] to learn these parameters. Specifically, for DPChain, the hyperparameters are (α, η). According to (3),

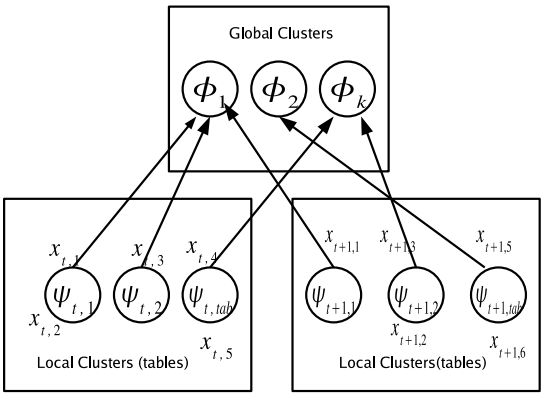


Figure 3. The illustrated example of global and local cluster correspondence

updating η results directly in updating $w_{t,k}$. Consequently, we actually update the hyperparameters $\Theta = (\alpha, w_{t,k})$. Following [10], α is sampled from the Gamma Distribution at each iteration in the Gibbs sampling in the E-step. In the M-step, similar to [25], we update $w_{t,k}$ by maximizing the cluster assignment likelihood. Suppose that, at an iteration, there are K clusters.

$$w_{t,k}^{new} = \frac{n_{t,k}}{\alpha + n_t - 1} \cdot \sum_{j=1}^{K} w_{t,j}^{old} \tag{19}$$

Thus, the EM framework is as follows:

- At time t, initialize parameters Θ and $z_{t,i}$
- E-Step: Sample α from Gamma Distribution. Sample cluster assignment $z_{t,i}$ for data item $x_{t,i}$ by (8);
- M-Step: Update $w_{t,k}$ by (19).
- Iterate the E-Step and the M-Step until the EM converges.

For HDP-EVO, the hyperparameters are $\Theta = (\alpha, \gamma, \lambda)$, Similar parameter learning may be obtained using an EM again.

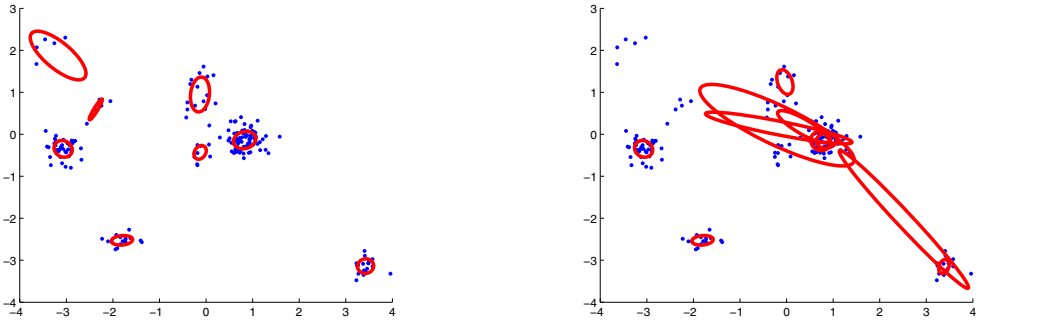
6 Experimental Evaluations

We have extensively evaluated the two models in comparison with the state-of-the-art literature, the PCM and PCQ algorithms developed in [8]. For the experiments in text data evolutionary clustering, we have also evaluated the two models in comparison with LDA [5, 13] in addition. The evaluations are performed in three datasets, a synthetic dataset, the 20 NewsGroups dataset, and a Google daily news dataset we have collected over a period of 5 continuous days.

6.1 Synthetic Dataset

We have generated a synthetic dataset according to our assumption of the evoluationary data. At each time, the data are a collection of mixture models with the number of the clusters as an unknown prior; the data evolve over the time under a smooth transition. Specifically, in the dataset, we have 10 different data collections corresponding to 10 different times, with each collection according to the DPM model with 200 2-dimensional Gaussian distribution points. 10 Gaussian points in $\mathbf{N}(\mathbf{0}, \mathbf{2I})$ are set as the 10 global clusters' mean parameters ϕ; then 200 Gaussian points within a cluster are sampled with this cluster's mean parameter and deviation parameter sampling from $\mathbf{N}(\mathbf{0}, \mathbf{0.2I})$, where $\mathbf{I}$ is identify matrix. At each time, part of the clusters are chosen from the previous collections, with a weight inversely proportional to their difference in time; other clusters are sampled from the multinomial distribution with the current mixture proportion vector, which is a sample from a symmetric DP with parameter 0.1. Consequently, each time, we sample 200 2-dimensional data points from Gaussian distribution according to the corresponding cluster parameters ϕ we have chosen at time t. Thus, new and existing clusters of Gaussian distribution appear at the coming times, according to their history. After the generation of such dataset, we obtain the number of the clusters and the cluster assignment as the ground truth. We intentionally generate different numbers of the clusters at different times, as shown in Figure 6.

In the inference process, we tune the hyperparameters as follows. In each iteration, we use vague gamma priors [10] to update α and γ from $\Gamma(1, 1)$. Smoothing parameter λ (consequently $\mathbf{w}_t$ as well) is updated according to (19). Figure 4 shows an example of the clustering results between HDP-EVO and PCQ at time 8 for the synthetic data. Clearly, HDP-EVO has a much better performance than PCQ in this synthetic data. For a more systematic evaluation on this synthetic dataset, we use NMI (Normalized Mutual Information) [20] to qantitatively compare the clustering performances among all the four algorithms (DPChain, HDP-EVO, PCM, and PCQ). Figure 5 documents the performance comparison. From this figure, the average NMI values across the 10 times for DPChain and HDP-EVO are 0.74 and 0.85, respectively, while those for PCQ and PCM are 0.70 and 0.71, respectively. DPChain works worse than HDP-EVO for the synthetic data. The reason is that DPChain model is unable to accurately capture the cluster correspondence among the data collections across the time in this case, but still performs better than PCQ and PCM. Since one of the advantages of the two pro-



(a) (b)

Figure 4. Clustering results of HDP-EVO (a) and PCQ (b) for the synthetic data

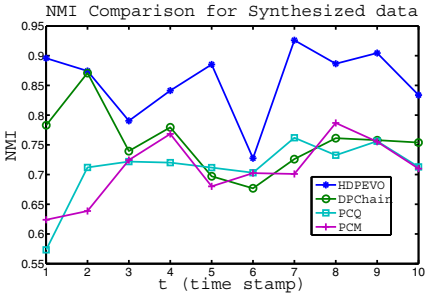


Figure 5. The NMI performance comparison of the four algorithms on the synthetic dataset

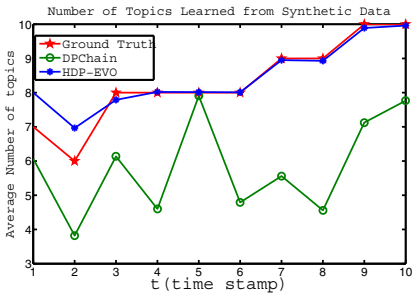


Figure 6. The cluster number learning performance of the two proposed models on the synthetic dataset

posed models is to be able to learn the number of of the clusters and the clustering structures during the evolution, we report this performance for the two models on this synthetic dataset in Figure 6. 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, both models are able to learn the cluster numbers, with HDP-EVO better in performance than DPChain. Since both PCQ and PCM do not have this capability, they are not included in this evaluation.

6.2 Real Dataset

In order to demonstrate and evaluate the proposed models on a real dataset, we construct a real dataset based on a subset of the 20 Newsgroups data [1]. We intentionally set the number of the clusters at each time the same number to accommodate the comparing algorithms PCQ and PCM which have this assumption of the same cluster number over the evolution. In order to compare the text clustering capability of LDA [5, 13] with a known topic number, we here set the topic number for LDA at each time collection as the ground truth 10. Consequently, we select 10 clusters (i.e., topics) from the dataset (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 dataset into 5 different collections, each of which has 20 documents randomly selected from each clusters. Consequently, each collection at a time has 10 topics to generate words. All the documents are preprocessed using the standard text processing techniques for removing the stop words and stemming the remaining words.

To apply the DPChain and HDP-EVO models, a symmetric Dirichlet distribution is used with the parameter 0.2 for the prior base distribution H. In each iteration, we update α and γ from the gamma priors $\Gamma(0.1, 0.1)$, λ (or $\boldsymbol{w}_t$) from (19). 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 with a known cluster number, in order

[1] http:kdd.ics.uci.edu/databases/20newsgroups/

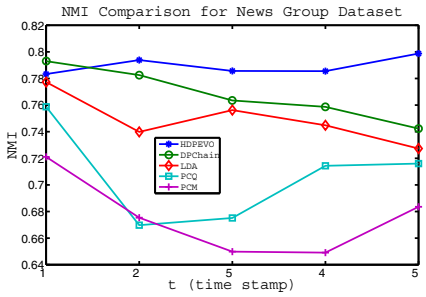


Figure 7. The NMI performance comparison of the five algorithms on the 20 Newsgroups dataset

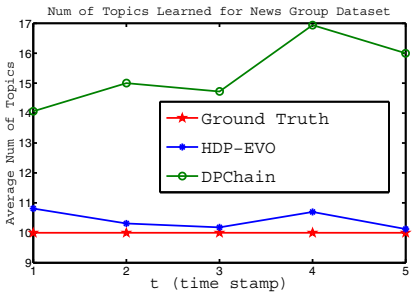


Figure 8. Cluster number learning performance of the two models on the 20 Newsgroups dataset

to compare with LDA, we explicitly apply LDA to the data collection with the ground truth cluster number as input at each time.

Figure 7 reports the overall performance comparison among all the five methods using NMI metric again. Clearly both proposed models substantially outperform PCQ, PCM, and LDA almost at all the times. HDP-EVO has a better performance than DPChain except at time 1 where there is no history information. Figure 8 further reports the performance on learning the cluster numbers at different times for the two proposed models. Both models have a reasonble performance in automatically learning the cluster number at each time in comparison with the ground truth. Again, HDP-EVO has a better performance than DPChain.

In order to truly demonstrate the performance of the proposed models in comparison with the state-of-the-art literature on a real evolutionary clustering scenario, we have manually collected Google News articles for a continuous window of five days (Feb. 10 - 14, 2008) where both the data items (i.e., words in the articles) and the clusters (i.e., the news topics) evolve over the time. We select a series number of clusters (ground truth in Table 1) at each day to reflect the evolving process of the clusters. We select 10 documents for each cluster everyday. Again, in order to compare the text clustering capability of LDA [5, 13] with a known topic number, 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 newsgroup dataset.

Figure 9 reports the NMI based performance evaluations among the five algorithms. Again, both proposed methods substantially outperform PCQ, PCM, and LDA in average, and HDP-EVO has a better performance than DPChain, except for at time 1 where there is no history information; 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 10 further reports the performance on learning

Table 1. Ground Truth of Google News Dataset

Day	1	2	3	4	5
Num. Clusters	5	6	5	6	6
Num. Documents	50	60	50	60	60
Num. Words	6113	6356	7063	7762	8035

the cluster numbers for different times for the two proposed models. Again, HDP-EVO has a much better performance than DPChain even though both methods are able to learn the cluster numbers automatically.

7 Conclusions

In this paper, we have addressed the evolutionary clustering problem. Based on the recent literature on DP based models, we have developed two separate models as two different solutions to this problem: DPChain and HDP-EVO. Both models substantially advance the evolutionary clustering literature in the sense that they not only perform better

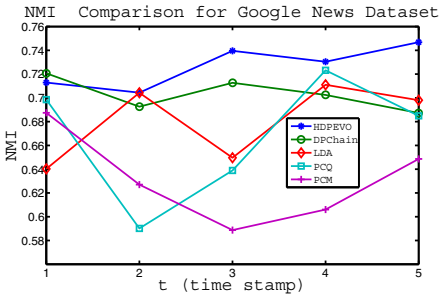


Figure 9. The NMI performance comparison for all the five algorithms on the Google News dataset

than the existing literature, but also are able to automatically learn the dynamic cluster numbers and the dynamic clustering structures during the evolution, which is a common scenario in many real evolutionary clustering applications. Extensive evaluations demonstrate the effectiveness of these models as well as their promise in comparison with the state-of-the-art literature.

8 Acknowledgement

This work is supported in part by NSF (IIS-0535162 and IIS-0812114). 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 NSF.

References

[1] D. Aldous. Exchangeability and related topics. *Ecole de Probabilites de Saint-Flour*, (XIII):1–198, 1983.

[2] C. Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

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

[4] D. Blei and J. Lafferty. Dynamic topic models. In *In Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[5] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2003.

[6] G. Casella and E. I. George. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, Aug. 1992.

[7] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 554–560, 2006.

[8] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 153–162, 2007.

[9] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[10] M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *The Annals of Statistics*, 90:577–588, 1995.

[11] T. S. Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

[12] T. L. Griffiths and M. Steyvers. Finding scientific topics. In *Proceedings of the National Academy of Sciences*, pages 5228–5235. Feb., 2004.

[13] G. Heinrich. Parameter estimation for text analysis. *Technical Report*, 2004.

[14] R. M. Neal. Probabilistic inference using markov chain monte carlo methods. *Technical Report*, (CRG-TR-93-1), 1993.

[15] R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, June 2000.

[16] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS 14*, 2002.

[17] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised activity perception by hierarchical bayesian models. In *British Machine Vision Conference (BMVC)*, 2006.

[18] J. Sethuraman. A constructive definition of dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[19] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8), August 2000.

[20] A. Strehl and J. Ghosh. Cluster ensembles a knowledge reuse framework for combining partitionings. In *Proceedings of AAAI*, 2002.

[21] Y. Teh, M. B. M. Jordan, and D. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2007.

[22] X. Wang, X. Ma, and E. Grimson. Unsupervised activity perception by hierarchical bayesian models. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Patter Recognition (CVPR)*, 2007.

[23] X. Wang and A. McCallum. 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*, pages 424–433, 2006.

[24] T. Xu, Z. Zhang, P. Yu, and B. Long. Evolutionary clustering by hierarchical dirichlet process with hidden markov state. In *ICDM*, 2008.

[25] X. Zhu, Z. Ghahramani, and J. Lafferty. Time-sensitive dirichlet process mixture models. *Technical Report*, (CMU-CALD-05-104), May 2005.

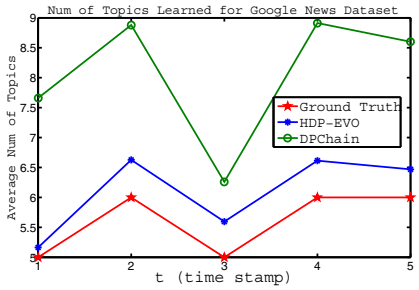


Figure 10. The cluster number learning performance of the two models on the Google News dataset