Dynamic Non-Parametric Mixture Models and The Recurrent Chinese Restaurant Process

Amr Ahmed Eric P. Xing

July 2007
CMU-ML-07-116

Carnegie Mellon

Dynamic Non-Parametric Mixture Models and The Recurrent Chinese Restaurant Process[a]

Amr Ahmed[†] **Eric P. Xing**[†]

July 2007
CMU-ML-07-116

[a]last modified on Jan. 2008

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[†]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

Dirichlet process mixture models provide a flexible Bayesian framework for density estimation; however they are inadequate with respect to modeling sequential data due to the full exchangeability assumption they employ. In this paper we present the temporal Dirichlet process mixture model (TDPM) as a framework for modeling complex longitudinal data. In a TDPM, the data is divided into epochs; all data points inside the same epoch are fully exchangeable, whereas the temporal order is maintained across epochs. Moreover, The number of mixture components in each epoch is unbounded: the components can retain, die out or emerge over time, and the actual parameterization of each component can also evolve over time in a Markovian fashion. We give three equivalent construction of this process as well as a Gibbs sampling algorithm to carry out posterior inference. We demonstrate our model by using it to build an infinite dynamic mixture of Gaussian factors, and a simple non-parametric dynamic topic model applied to the NIPS12 collection.

Keywords: Dirichlet Processes, Dynamic systems, Topic Models, Clustering

1 Introduction

Dirichlet process mixture models provide a flexible Bayesian framework for estimating a distribution as an infinite mixture of simpler distributions that could identify latent classes in the data [1]. However the full exchangeability assumption they employ makes them an unappealing choice for modeling longitudinal data such as text, audio and video streams that can arrive or accumulate as epochs, where data points inside the same epoch can be assumed to be fully exchangeable, whereas across the epochs both the structure (i.e., the number of mixture components) and the parameterizations of the data distributions can evolve and therefore unexchangeable. In this paper, we present the temporal Dirichlet process mixture model (TDPM) as a framework for modeling these complex longitudinal data, in which the number of mixture components at each time point is unbounded; the components themselves can retain, die out or emerge over time; and the actual parameterization of each component can also evolve over time in a Markovian fashion. In the context of text-stream model, each component can thus be considered as a common themes or latent class that spans consecutive time points. For instance, when modeling the temporal stream of news articles on a, say, weekly basis, moving from week to week, some old themes could fade out (e.g., the mid-term election is now over in US), while new topics could appear over time (e.g., the presidential debate is currently taking place). Moreover, the specific content of the lasting themes could also change over time (e.g, the war in Iraq is developing with some slight shift of focus). The rest of this paper is organized as follows. First in section 2 we review the Dirichlet process mixture model, and use it to motivate the TDPM model which we introduce in section 3. Section 3 also gives three different, and equivalent, constructions for the TDPM model: via the recurrent Chinese restaurant process (section 3.1), as the infinite limit of a finite dynamic mixture model (section 3.2), and finally via a temporally dependent random measures (section 3.3). In section 4 we give a Gibbs sampling algorithm for posterior inference. Section 5 extends the construction to higher order dependencies. In Section 6 we use the TDPM to built 1 an infinite dynamic mixture of Gaussian factors (i.e., an infinite mixture Kalman filters of different life-spans) and illustrate it on simulated data. Then in section 7, we give a simple non-parametric topic model on top of the TDPM and use it to analyze the NIPS12 collection. In section 8, we discuss relation to related work and in Section 9 we conclude and discuss possible future problems.

2 The Dirichlet Process Mixture Model

In this section we introduce the basic and well-known DPM model via three constructions. First, as a distribution over distributions, then via the intuitive Chinese restaurant process (CRP), and finally as a the limit of a finite mixture model. All of these views are equivalent, however, each one provides a different view of the same process, and some of them might be easier to follow, especially the CRP metaphor.

Technically, the Dirichlet process (DP) is a distribution over distributions [2]. A DP, denoted by $DP(G_0, \alpha)$, is parameterized by a base measure, G_0, and a concentration parameter, α. We write $G \sim DP(G_0, \alpha)$ for a draw of a distribution G from the Dirichlet process. G itself is a distribution over a given parameter space, θ, therefore we can draw parameters $\theta_{1:N}$ from G. The parameters drawn from G follow a Polya-urn scheme [3], also known as the Chinese restaurant process (CRP), in which previously drawn values of θ have strictly positive proba-

bility of being redrawn again, thus making the underlying probability measure G discrete with probability one [2]. By using the DP at the top of a hierarchical model, one obtains the Dirichlet process mixture model, DPM for non-parametric clustering [4]. The generative process for the DPM proceeds as follows:

$$\begin{aligned} G \mid \alpha, G_0 &\sim DP(\alpha, G_0), \\ \theta_n | G &\sim G, \\ \theta_n &\sim F(.|\theta_n), \end{aligned} \tag{1}$$

where F is a given likelihood function parameterized by θ, for instance in the cause of a Gaussian emission, F is the normal pdf, and θ is its mean and covariance. The clustering property of the DP prefers that fewer than N distinct θ are used. This notion is made explicit via the equivalent CRP metaphor. In the CRP metaphore, there exists a Chinese restaurant with an infinite numbers of tables. Customer x_i enters the restaurant and sits on table k that has n_k customers with probability $\frac{n_k}{i-1+\alpha}$, and shares the dish (parameter), ϕ_k, served there, or picks a new table with probability $\frac{\alpha}{i-1+\alpha}$, and orders a new dish sampled from G_0. Putting everything together, we have:

$$\theta_i | \theta_{1:i-1}, G_0, \alpha \sim \sum_k \frac{n_k}{i-1+\alpha} \delta(\phi_k) + \frac{\alpha}{i-1+\alpha} G_0. \tag{2}$$

Equation 1 can also be obtained by integrating out G from the equations in (2), and this shows the equivalence of the two schemes. Finally, the DPM can be arrived at if we consider a fixed K-dimensional mixture model, like K-means, and then take the limit as K $\rightarrow \infty$. In other words, a DPM can potentially model an infinite-dimensional mixture model and thus has the desirable property of extending the number of clusters with the arrival of new data (which is made explicit using the CRP metaphor — a new customer can start a new cluster by picking an unoccupied table). This flexibility allows the DPM to achieve model selection automatically. However, it is vitally *important* now to clear some myths with regards to the DPM. While DPM is known as a non-parametric model, it still does have parameters, namely, α, albeit being dubbed as *hyperparmaters*. The reason for calling α a hyperparmater is to distinguish between it and between the *effective* parameters of the model which are: K, the number of mixture components and their associated parameters, like mean and covariance in the of case a mixture of Gaussian distributions. These effective parameters need not be specified for a DPM model, but must be specified for any parametric model like K-means; Hence came the name non-parametric. In essence, the role played by the hyper-parameter α is to specify the *rate* at which the effective parameters of the model *grow* with the data. Hyperparmaters can be either supplied by the user to encode their prior knowledge, or desirable outcome (finer vs. coarser clustering), or can be learnt automatically using an EM-like algorithm called empirical Bayes [20].

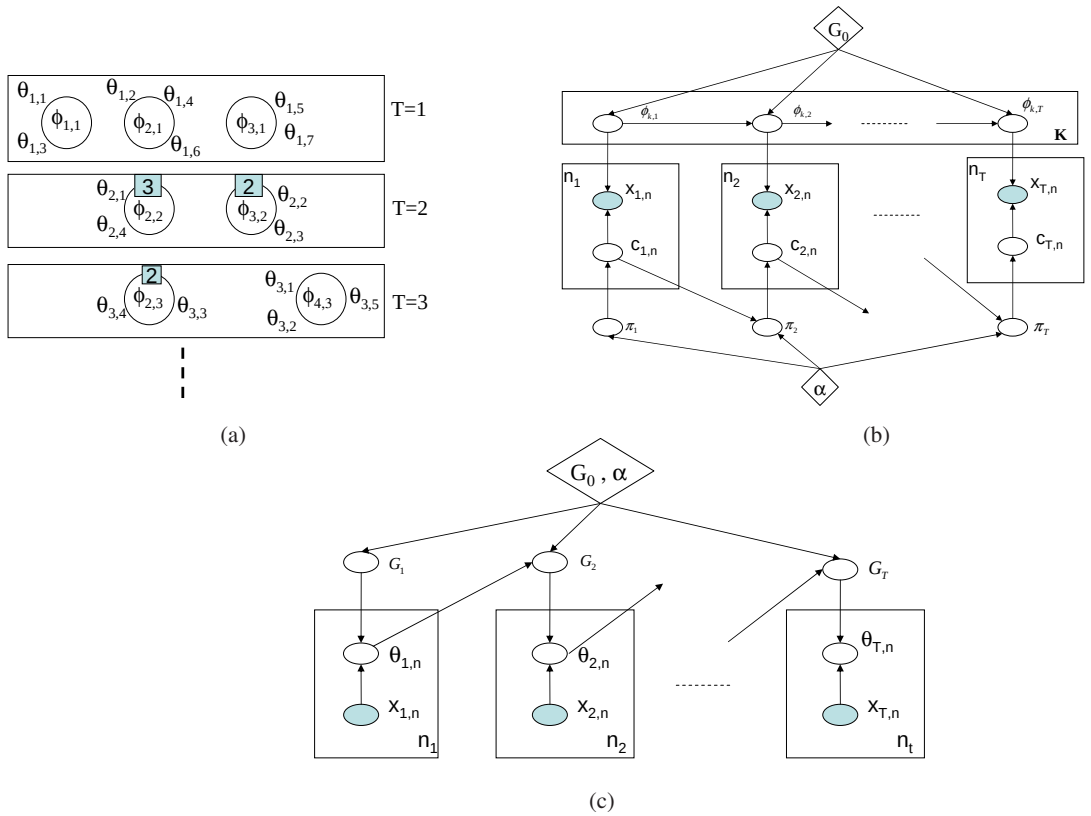

Figure 1: Three constructions for the TDPM. a) The recurrent Chinese restaurant process and b) The infinite limit of a finite dynamic mixture model. In b, diamonds represent hyper-parameters, shaded circles are observed variables, and unshaded ones are hidden variables, plates denote replications, where the number of replica is written inside the plate, for instance n_1. (C)Time dependent random measure construction of the TDPM.

3 The Temporal Dirichlet Process Mixture Model

In many domains, data items are not fully exchangeable, but rather partially exchangeable at best. In the TDPM model[1] to be presented, data are assumed to arrive in T consecutive epochs, and inside the same epoch all objects are fully exchangeable.

Intuitively the TDPM seek to model cluster parameters evolution over time using any time series model, and to capture cluster popularity evolution over time via the rich-gets-richer effect, i.e. the popularity of cluster k at time t is proportionable to how many data points were associated with cluster k at time $t-1$. In the following subsections, we will formalize these notions by giving three equivalent constructions for the TDPM as summarized in Figure 1. However, before giving these constructions that parallel those given for the DPM in section 3, we first start by specifying some notations.

3.1 Notations and Conventions:

We let n_t denotes the number of data points in the t^{th} epoch, and $x_{t,i}$ denotes the i^{th} point in epoch t. The mixture components (clusters) that generate the data can emerge, die out, or

[1] A preliminary earlier version of the TDPM model first appeared in [19]

evolve its parametrization evolve over time in a Markovian fashion, therefore, we generalize the notion of a mixture into a chain that links the parameters of the mixture component over time. We let ϕ_k denote chain k, $\phi_{k,t}$ denoted the state (parameter value) of chain k at time t, and $n_{k,t}$ denotes the number of data points associated with chain k at time t. Moreover, we use $n_{k,t}^{(i)}$ to denote the same quantity just before the arrival of datum $x_{t,i}$. Note that the chains need not have the same life span; however, once retained over time they keep the same chain index. Moreover, the set of chain indexes available at time t might not be contiguous (because some chains may have died out). Therefore, we define I_t to denote the set of chain indexes available at time t. We sometimes overload notation and use $I_t^{(i)}$ to denote the same quantity just before the arrival of datum $x_{t,i}$. Each data item $x_{t,i}$ is generated from a mixture with parameter $\theta_{t,i}$, if we let $c_{t,i}$ denotes the chain index associated with this datum, then we have $\theta_{t,i} = \phi_{c_{t,i},t}$ — in other words, the set of ϕ's define the unique mixtures/clusters, or put it equivalently, two data items might have equal θ values if they belong to the same cluster. Moreover, we might use the following abbreviations for notational simplicity (z denotes a generic variable):

- $\{z_{t,.}\}$ to denote $\{z_{t,1}, z_{t,2}, \cdots\}$
- $z_{t,1:i}$ to denote $\{z_{t,1}, z_{t,2}, \cdots, z_{t,i}\}$

3.2 The Recurrent Chinese Restaurant Process

The RCRP, showin in Figure 1-a, is a generalization of the CRP introduced in section 3. The RCRP operates in epochs, say, days. Customers entered the restaurant in a given day are not allowed to stay beyond the end of this day. At the end of each day, the consumptions of dishes are analyzed by the owner of the restaurant who assumes that popular dishes will remain popular in the next day, and uses this fact to plan the ingredients to be bought, and the seating plan for the next day. To encourage customers in the next day to try out those pre-planed dishes, he records on each table the dish which was served there, as well as the number of customers who shared it. As another incentive, he allows the first customer to set on such a table to order a (flavored) variation of the dish recorded there. In this metaphor, dishes correspond to chains, and the variation correspond to the dynamic evolution of the chain. The generative process proceeds as follows. At day t, customer i can pick an empty table, k, that was used to serve dish $\phi_{k,t-1}$, with probability equals to $\frac{n_{k,t-1}}{N_{t-1}+i+\alpha-1}$, he then chooses the current flavor of the dish,$\phi_{k,t}$, distributed according to $\phi_{k,t} \sim P(.|\phi_{k,t-1})$. If this retained table k has already $n_{k,t}^{(i)}$customers, then he joins them with probability $\frac{n_{k,t-1}+n_{k,t}^{(i)}}{N_{t-1}+i+\alpha-1}$ and shares the current flavor of the dish there. Alternatively, he can pick a *new empty* table that was not used in the previous day,$t-1$, i.e., not available in I_{t-1}, with probability $\frac{\alpha}{N_{t-1}+i+\alpha-1}$,lets call it K^+, and orders a dish $\phi_{K^+,t} \sim G_0$ — this is the mechanism by which a new chain/cluster emerges. Finally, he can share a *new* table k, with $n_{k,t}^{(i)}$ customers, with probability $\frac{n_{k,t}^{(i)}}{N_{t-1}+i+\alpha-1}$ and shares the newly ordered dish with them. Putting everything together, we have:

$$\theta_{t,i}|\{\theta_{t-1,.}\}, \theta_{t,1:i-1}, G_0, \alpha \sim \frac{1}{N_{t-1}+i+\alpha-1} \times \Big[\sum_{k \in I_{t-1}} \Big(n_{k,t-1} + n_{k,t}^{(i)}\Big)\delta(\phi_{k,t}) + \sum_{k \in I_t^{(i)} - I_{t-1}} n_{k,t}^{(i)}\delta(\phi_{k,t}) + \alpha G_0 \Big], \quad (3)$$

where in the first summation $\phi_{k,t} \sim P(.|\phi_{k,t-1})$ (i.e. retained from the previous day), and in the second one $\phi_{k,t} \sim G_0$ which is drawn by the j^{th} customer at time t for some $j < i$ (i.e. new chains born at epoch t). If we conveniently define $n_{k,t}$ to be 0 for $k \in I_{t-1} - I_t^{(i)}$ (chains which died out) and similarly $n_{k,t-1}$ be 0 for $k \in I_t^{(i)} - I_{t-1}$ (i.e. newly born chains at time t), then we can compactly write Equation 3 as:

$$\theta_{t,i}|\{\theta_{t-1,.}\}, \theta_{t,i:i-1}, G_0, \alpha \sim \frac{1}{N_{t-1}+i+\alpha-1} \times \Big[\sum_{k \in I_{t-1} \cup I_t^{(i)}} \Big(n_{k,t-1} + n_{k,t}^{(i)}\Big)\delta(\phi_{k,t}) + \alpha G_0 \Big]. \quad (4)$$

3.3 The infinite Limit of a finite Dynamic Mixture Model

In this section we show that the same sampling scheme in Equation (4) can be obtained as the infinite limit of the finite mixture model in Figure 1-b. We consider the following generative process for a finite dynamic mixture model with K mixtures. For each t do:

1. $\forall k$: Draw $\phi_{k,t} \sim P(.|\phi_{k,t-1})$
2. Draw $\pi_t \sim Dir(n_{1,t-1} + \alpha/K, \cdots, n_{K,t-1} + \alpha/K)$
3. $\forall i \in N_t$ Draw $c_{t,i} \sim Multi(\pi_t)$, $x_{t,i} \sim F(.|\phi_{c_{t,i},t})$

By integrating over the mixing proportion π_t, It is quite easy to write the prior for $c_{t,i}$ as conditional probability of the following form:

$$P(c_{t,i} = k|c_{t-1,1:N_{t-1}}, c_{t,1:i-1}) = \frac{n_{k,t-1} + n_{k,t}^{(i)} + \alpha/K}{N_{t-1}+i+\alpha-1}. \quad (5)$$

If we let $K \to \infty$, we find that the conditional probabilities defining the $c_{t,i}$ reachs the following limit:

$$\begin{aligned} P(c_{t,i} = k|c_{t-1,1:N_{t-1}}, c_{t,1:i-1}) &= \frac{n_{k,t-1} + n_{k,t}^{(i)}}{N_{t-1}+i+\alpha-1} \\ P(c_{t,i} = \texttt{a new cluter}) &= \frac{\alpha}{N_{t-1}+i+\alpha-1} \end{aligned} \quad (6)$$

Putting Equations (5) and (6) together, we can arrive at Equation (4).

3.4 The Temporarily Dependent Random Measures view of the TDPM

Here we show that the same process in section 4 can be arrived at if we model each epoch using a DPM and connect the random measures G_t as shown in Figure 5. This appendix is rather technical and is provided only for completeness, however it can be skipped without any loss of continuity.

The derivation here depends on the well known fact that the posterior of a DP is a also a DP [4]. That is, $G|\phi_1, \cdots, \phi_k, G_0, \alpha \sim DP\Big(\alpha + n, \sum_k \frac{n_k}{n+\alpha}\delta(\phi_k) + \frac{\alpha}{n+\alpha}G_0\Big)$, where $\{\phi_k\}$ are the collection of unique values of $\theta_{1:n}$ sampled from G. Now, we consider the following generative process. For each t, do:

1. $\forall k \in I_{t-1}$ draw $\phi_{k,t} \sim P(.|\phi_{k,t-1})$
2. Draw $G_t|\{\phi_{k,t}\}\forall k \in I_{t-1}, G_0, \alpha \sim DP(\alpha + N_{t-1}, G_0^t)$
3. $\forall i \in N_t$, Draw $\theta_{t,i}|G_t \sim G_t \qquad x_{t,n}|\theta_{t,n} \sim F(.|\theta_{t,n})$

where $G_0^t = \sum_{k \in I_{t-1}} \frac{n_{k,t-1}}{N_{t-1}+\alpha}\delta(\phi_{k,t}) + \frac{\alpha}{N_{t-1}+\alpha}G_0$. Now by integrating $G_t \sim DP(N_{t-1} + \alpha, G_0^t)$. We can easily show that:

$$\theta_{t,i}|\{\theta_{t-1,.}\}, \theta_{t,1:i-1}, G_0, \alpha \sim \frac{1}{i + (\alpha + N_{t-1}) - 1}\Big[\sum_{k \in I_t^{(i)}} n_{k,t}^{(i)}\delta(\phi_{k,t}) + (\alpha + N_{t-1})G_0^t\Big]. \quad (7)$$

Now substituting G_0^t into the above equation plus some straightforward algebra, we arrive at:

$$\theta_{t,i}|\{\theta_{t-1,.}\}, \theta_{t,1:i-1}, G_0, \alpha \sim \frac{1}{N_{t-1} + i + \alpha - 1}\Big[\sum_{k \in I_t^{(i)}} n_{k,t}^{(i)}\delta(\phi_{k,t}) + \sum_{k \in I_{t-1}} n_{k,t-1}\delta(\phi_{k,t}) + \alpha G_0\Big], \quad (8)$$

which when rearranged is equivalent to Equation 4

4 Gibbs Sampling Algorithms

Given the previous constructions for the TDPM model, we are ready to derive a Gibbs sampling scheme equivalent to algorithm 2 in [1]. The state of the sampler contains both the chain indicator for every data item, $\{c_{t,i}\}$, as well as the value of all the available chains at all time epochs, $\{\phi_{k,t}\}$. We iterate between two steps: given the current state of the chains, we sample a class indicator for every data item, and then given the class indicators for all data item, we update the current state of the chains. We begin by the second step, let $\phi_k^{(x)}$ denote the collection of data points associated with chain k at all time steps, that is $\phi_k^{(x)} = \{\forall t(\forall i \in N_t)\, x_{t,i}|c_{t,i} = k\}$. Note also that conditioning on the class indicators, each chain is conditionally independent from the other chains. Therefore, $P(\phi_k|\{c_{t,i}\}) = P(\{\phi_{k,t}\}|\phi_k^{(x)})$. This calculation depends on both the chain dynamic evolution model $P(.|)$ and the data likelihood $F(.|.)$, therefore, this posterior should be handled in a case by case fashion, for instance, when the dynamic evolution model is a linear state-space model with Gaussian emission (likelihood), this posterior can be calculated exactly via the RTS smoother [5,6]. Once this posterior is calculated, we can update the current state of the chains by sampling each chain over time as a block from this posterior. Now, we proceed to the first step, for a given data point, $x_{t,i}$, conditioning on the state of the chains and other indicator variables (i.e. how data points other than $x_{t,i}$ are assigned to chains), we sample $c_{t,i}$ as follows:

$$\begin{aligned} &P(c_{t,i}|c_{t-1}, c_{t,-i}, c_{t+1}, x_{t,i}, \{\phi_k\}_{t,t-1}, G_0, \alpha) \propto \\ &P(c_{t,i}|c_{t-1}, c_{t,-i}, x_{t,i}, \{\phi_k\}_{t,t-1}, G_0, \alpha)P(c_{t+1}|c_t), \end{aligned} \quad (9)$$

where we introduce the following abbreviations: c_{t-1}, c_{t+1} denotes all indicators at time $t-1$ and t respectively. $c_{t,-i}$ denotes the chain indicators at time t without $c_{t,i}$, and $\{\phi_k\}_{t,t-1}$

denotes all chains alive at either time epoch t or $t-1$, i.e., $\phi_k \forall k \in I_{t-1} \cup I_t$. We also let $n_{k,t}^{(-i)}$ denote $n_{k,t}$ without the contribution of data point $x_{t,i}$. The first factor in Equation (9) can be computed using Eq. (4) as follows:

$$
\begin{aligned}
P(c_{t,i} = k \in I_{t-1} \cup I_t | ...) &\propto \frac{n_{k,t-1} + n_{k,t}^{(-i)}}{N_{t-1} + N_t + \alpha - 1} F(x_{t,i}|\phi_{k,t}) \\
P(c_{t,i} = K^+ | ...) &\propto \frac{\alpha}{N_{t-1} + N_t + \alpha - 1} \int F(x_{t,i}|\theta) dG_0(\theta),
\end{aligned} \tag{10}
$$

where K^+ denotes a globally new chain index (i.e., a new chain is born). It should be noted here that in the first part of Equation (10), there is a subtlety that we glossed over. When we consider a chain from time $t-1$ that has not been inherited yet at time t (that is $k \in I_{t-1}, n_{k,t}^{(-i)} = 0$), we must treat it exactly as we treat sampling a new chain from G_0 with G_0 replaced by $P(\phi_{k,t}|\phi_{k,t-1})$.

The second factor in Equation (9) can be computed with reference to the construction in section 4.3 as follows (note that c_t here includes the current value of $c_{t,i}$ under consideration in Equation 9). First, note that computing this part is equivalent to integrating over the mixture weights π_{t+1} which depend on the counts of the chain indicators at time t. The subtlety here is that in section 4.3 we let $K \to \infty$; however, here we only need to let the two count vectors be of equal size, which is $K_{t,t+1} = |I_{t,t+1}|$, where as defined before $I_{t,t+1} = I_t \cup I_{t+1}$, by padding the counts of the corresponding missing chains with zeros. It is straightforward to show that:

$$
\begin{aligned}
P(c_{t+1}|c_t) = &\frac{\Gamma\Big(\sum_{k \in I_{t,t+1}} n_{k,t} + \alpha/K_{t,t+1}\Big)}{\prod_{k \in I_{t,t+1}} \Gamma(n_{k,t} + \alpha/K_{t,t+1})} \times \\
&\frac{\prod_{k \in I_{t,t+1}} \Gamma(n_{k,t} + n_{k,t+1} + \alpha/K_{t,t+1})}{\Gamma\Big(\sum_{k \in I_{t,t+1}} n_{k,t} + n_{k,t+1} + \alpha/K_{t,t+1}\Big)}
\end{aligned} \tag{11}
$$

It should be noted that the **cost** of running a full Gibs iteration is $O(n)$ where n is the total number of data points.

5 Modeling Higher-Order Dependencies

One problem with the above construction of the TDPM is that it forgets too quickly especially when it comes to its ability to model cluster popularity at time $t+1$ based on its usage pattern at time t, while ignoring all previous information before time epoch t. Moreover, once a cluster is dead, i.e. its usage pattern at time t is 0, it can no longer be revived again. Clearly, in some applications one might want to give a slack for a cluster before declaring it dead. For example, when the TDPM is used to model news stories on a daily basis, if a theme that was active in time epoch $t-1$ had no documents associated with it at time t, then the TDPM will consider it dead, however, in practice, this might not be the case.

By analogy to the RCRP equivalent construction, the owner who plans the restaurant ingredients based on a daily usage is less prudent than an owner who considers a larger time frame, perhaps a week. However, one should not treat the usage pattern of cluster k at time t and at

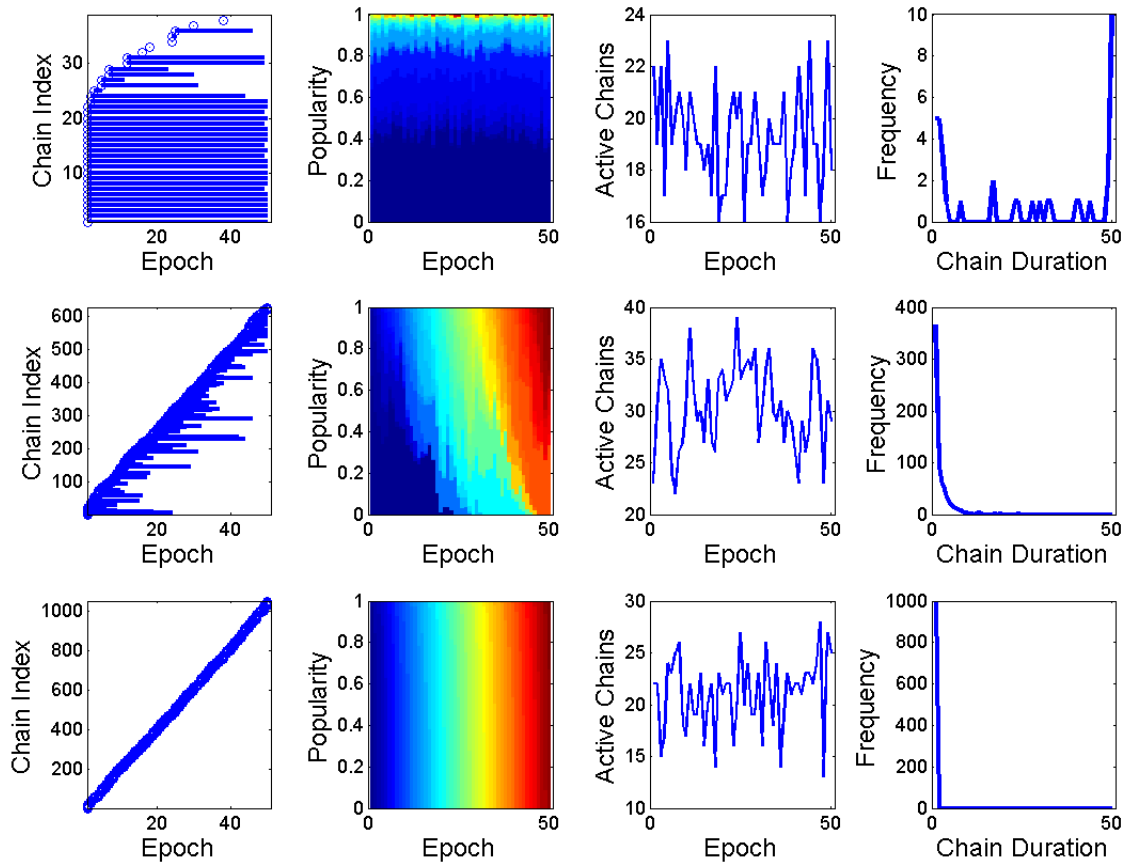

Figure 2: Simulating various clustering patterns from a TDPM(α,λ,W). **Top**: DPM, **middle**: a TDPM and **bottom**: a set of independent DPM at each epoch. See section 6 for more details

time, say, $t-h$, as contributing equally to our prediction of this cluster's popularity at time $t+1$. A possible solution here is to incorporate historic usage patterns by *decaying* their contribution exponentially over time epochs. A similar idea has been proposed in [14], however in [14], each epoch has exactly one data point, and the width of the history window used in [14] is rather infinity — or more precisely at most n. This in fact makes the cost of running a single Gibbs iteration, i.e. sampling all data items once, $O(n^2)$. In the solution we propose here, we define two new hyperparmaters, kernel width, λ, and history size, W. We will describe our approach only using the RCRP for simplicity since as shown before, it is equivalent to the other constructions. To model higher order dependencies, the only difference is that the owner of the restaurant records on each table, not only its usage pattern on day $t-1$, but its weighted cumulative usage pattern over the last W days. Where the weight associated with the count from day $t-h$ is given by $\exp^{\frac{-h}{\lambda}}$, and as such the contribution from epoch $t-h$ decays exponentially over time. A customer $x_{t,n}$ entering the restaurant at time t will behave exactly in the same way as before using the new numbers recorded on the table.

There are two implications to this addition. First, the cost of running one Gibbs iteration is $O(n \times W)$, which is still manageable as W must be smaller than T, the number of epochs, which is in turn much smaller than the total number of data points, n, thus we still maintain a linear time complexity. Second, an active cluster is considered dead if and only if, it is not used for exactly W contiguous echoes, which creates the necessary slack we were looking for. Changing the Gibbs sampling equations in section 5 to accommodate this new addition is very

straightforward and removed for the light of space.

It is interesting to note that these two new hyper-parameters allow the TDPM to degenerate to either a set of independent DPMs at each epoch when W=0, and to a global DPM, i.e ignoring time, when $W = T$ and $\lambda = \infty$. In between, the values of these two parameters affect the expected life span of a given cluster/chain. The larger the value of W and λ, the longer the expected life span of chains, and vice versa.

To illustrate this phenomenon, we sampled different cluster configurations from the TDPM model by running the RCRP metaphor for T = 50 epochs and seating 300 customers at each epoch. We simulated three hyper-parameter configurations (α, λ, W) as follows. The configuration used at the top of Figure 2 is (5,∞,50=T) which reduces the TDPM to a DPM. The configuration at the middle is a TDPM with hyperparmaters (5,.4,4), while the bottom TDPM degenerates to a set of independent DPMs at each epoch by setting the hyper-parameters to (5,.5,0) — in fact the value of λ here is irrelevant. For each row, the first panel depicts the duration of each chain/cluster,the second panel shows the popularity index at each epoch, i.e. each epoch is represented by a bar of length one, and each *active* chain is represented by a color whose length is proportional to its popularity at this epoch. The third panel gives the number of active chains at each epoch and the fourth panel shows the number of chains with a given life-span (duration). This fourth panel is a frequency curve and in general all TDPMs exhibit a power-law (Zipf's) distribution as the one in the middle, but with different tail lengths, while a DPM and independent DPMs show no such power-law curves. Another interesting observation can be spotted in the second column: note how cluster intensities change smoothly over time in the TDPM case, while it is abrupt in independent DPMs or rarely changing in a global DPM. This shows that TDPM with three tunable variables can capture a wide range of clustering behaviors.

6 Infinite Dynamic Mixture of Gaussian Factors

In this section we show how to use the TDPM model to implement an infinite dynamic mixture of Gaussian factors. We let each chain represent the evolution of the mean parameter of a Gaussian distribution with a fixed covariance Σ. The chain dynamics is taken to be a linear state-space model, and for simplicity, we reduce it to a random walk. More precisely, for a given chain ϕ_k: $\phi_{k,t}|\phi_{k,t-1} \sim N(\phi_{k,t-1}, \rho I)$ and $x_{t,i}|c_{t,i} = k \sim N(\phi_{k,t}, \Sigma)$. The base measure $G_0 = N(0, \sigma I)$. Using the Gibbs sampling algorithm in section 5, computing the chain posterior given its associated data points, $\phi_k^{(x)}$, can be done exactly using the RTS smoother algorithm [6]. We simulated 30 epochs, each of which has 100 points from the TDMP with the above specification, and with hyperparmaters as follows: $\alpha = 2.5, \mathrm{W} = 1, \lambda = .8, \sigma = 10, \rho = 0.1 \mathrm{and}\, \Sigma = 0.1\mathrm{I}$. We ran Gibbs sampling for 1000 iterations and then took 10 samples every 100 iterations for evaluations. The results shown in Figure 3 are averaged over these 10 samples. To measure the success of the TDPM, we compared the clustering produced by the TDPM to the ground truth, and to that produced from a fixed dynamic mixture of Kalman Filters [6] with various number of chains, K =(5,10,15,20,25,30). For each K, we ran 10 trials with different random initializations and averaged the results.

We compared the clustering produced by the two methods, TDPM, and the one with fixed number of *evolving* chains over time, to the ground truth using the variation of information measure in [21]. This measure uses the mutual information between the two clustering under

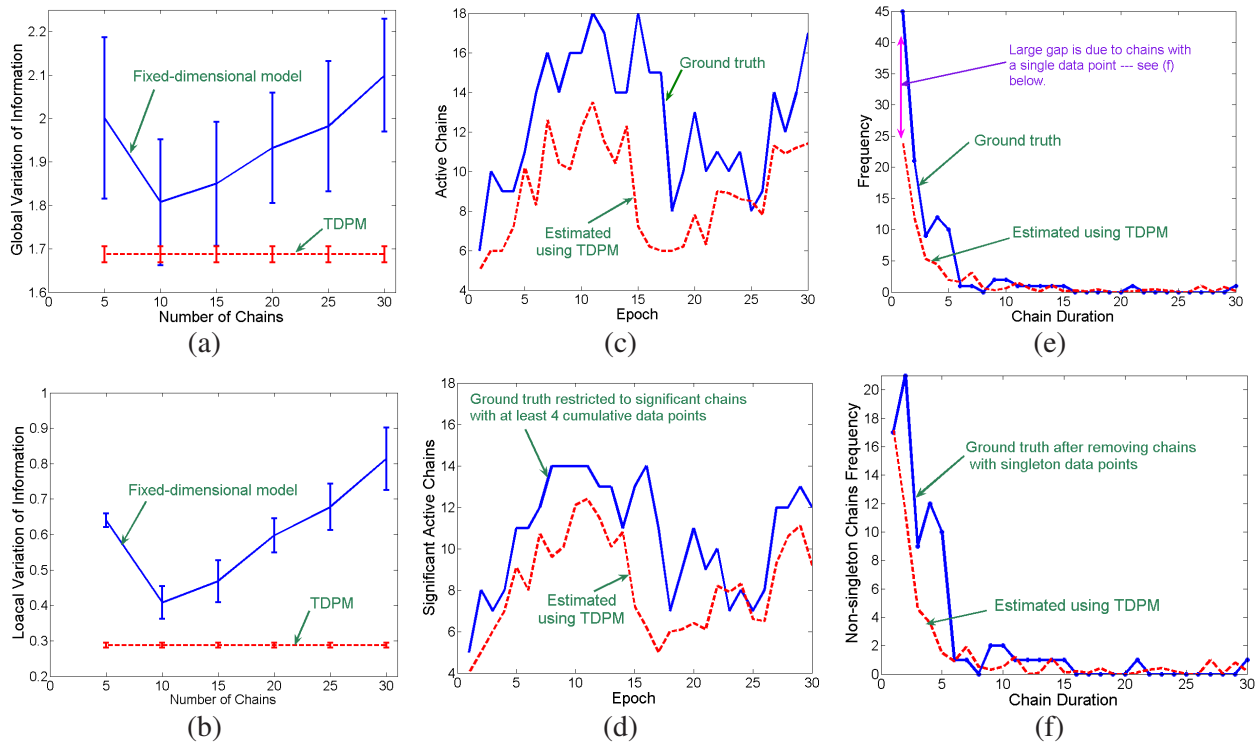

Figure 3: Illustrating results on simulated data. Panels (a,b) contrast the accuracy of the recovered clustering, using global and local consistency measures, against that estimated using fixed dimensional models (see text for details). Panels (c-f) illustrate the TDPM ability to vary the number of clusters/chains over time, results from fixed-dimensional models, which is fixed over time, are not shown to avoid cluttering the display. Panel (d) and (f) illustrate that most omissions (errors) are due to insignificant chains. All results are averaged over 10 samples taken 100 iterations apart for the TDPM, and over 10 random initializations for the fixed-dimensional models. Error bars are not shown in panels (c-f) for clarity, however,the maximum standard error is 1.4

.

consideration, and their entropy to approximate the distance between them across the lattice of all possible clustering (see [21] for more details). We explored two ways of applying this measure to dynamic clustering, the global variation of information, GVI, and the local variation of information, LVI. In GVI, we ignored time, and considered two data points to belong to the same cluster if they were generated from the same chain at any time point. In LVI, we applied the VI measure at each time epoch separately and averaged the results over epochs. GVI captures global consistency of the produced clustering, while LVI captures local consistency (adaptability to changes in the number of clusters). The results are shown in Figure 3-a, 3-b (lower values are better)and show that the TDPM is superior to a model in which the number of clusters are fixed over time, moreover, setting K to the maximum number of chains over all time epochs does not help. In addition to these measures, we also examined the ability of the TDPM to track the evolution of the number of chains (Figure 3-c)and their duration over time (Figure 3-e). These two figures show that in general, the TDPM tracks the correct ground-truth behavior, and in fact most of the errors are due to insignificant chains, i.e. chains/clusters which contain a very small (1-3) data points as shown in Figure 3-d and Figure 3-f. It is worth

mentioning that the fixed dimension models produce the same number of chains over time, which we omit from Figure 3-(c-f) for clarity.

7 A Simple Non-Parametric Dynamic Topic Model

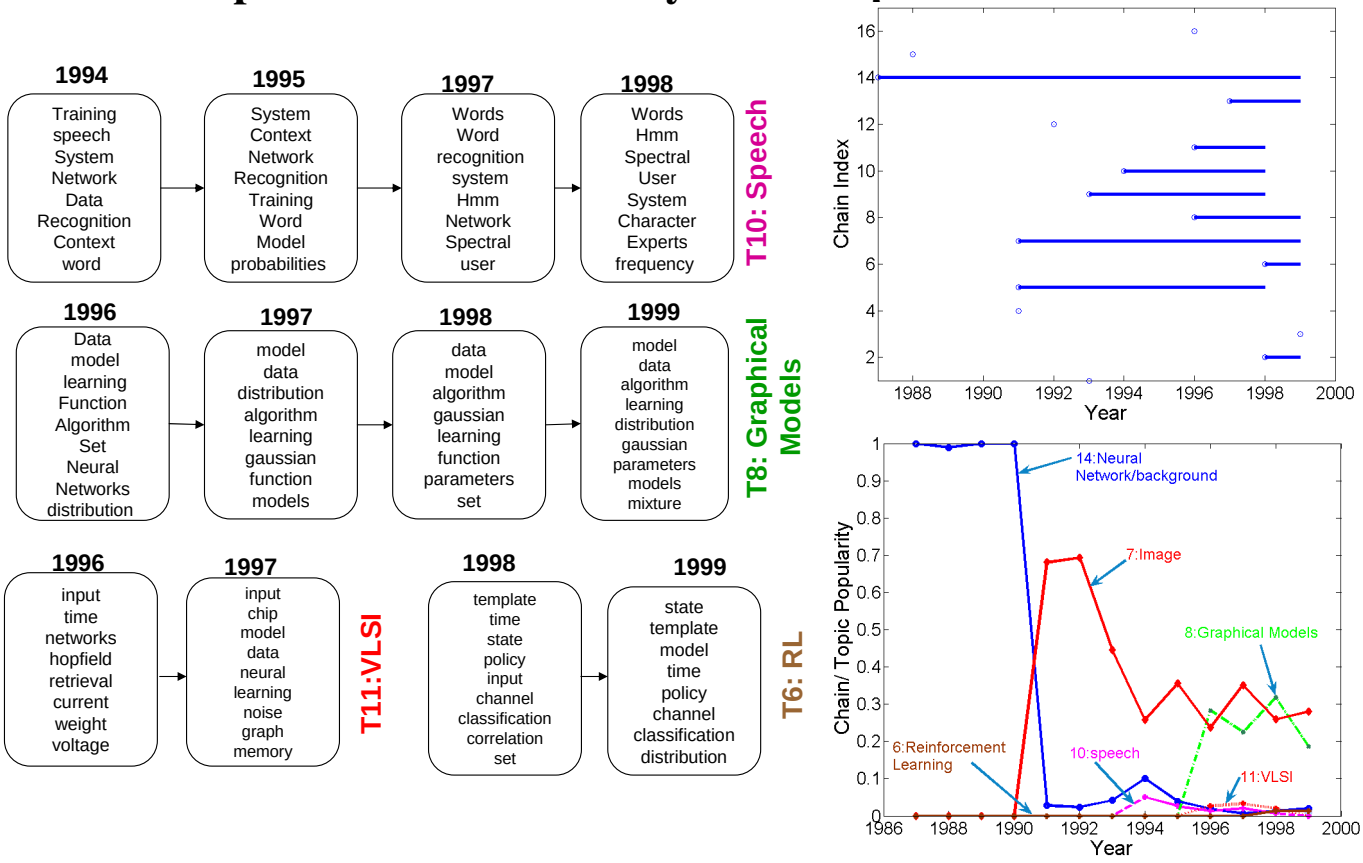

Figure 4: Illustrating results on the NIPS12 dataset. **Right-top**: chains (topics) death-birth over time. **Right-bottom**: the popularity of some topics over the years, where topics names are hand labeled. **Left**: keywords over time in some topics.

Statistical admixture topic models have recently gained much popularity in managing large document collections. In these models, each document is sampled from a mixture model according to a document's specific mixing vector over the mixture components (*topics*), which are often represented as a multinomial distribution over a given vocabulary. An example of such models is the well-known latent Dirichlet allocation (LDA)[7]. Recent approaches advocate the importance of modeling the dynamics of different aspects of topic models: topic trends [10], topic word distributions [8] and topic correlations [9]. In this section we show how to implement a simple non-parametric dynamic topic model. The model presented here is simpler than mainstream topic models in that each document is generated from a *single* topic rather than from a mixture of topics as in LDA. However, this is not a restriction of our framework, as we will mention in the future work section how this simple model can be extend to a full-fledged one. The model we present here is only meant as another illustration of the generality of our framework.

To implement this simple non-parametric dynamic topic model, SNDTM for short, let $x_{t,i}$ represent a document composed of word frequency counts. Each chain represents the natural

parameter of the multinomial distribution associated with a given topic, similar to the the dynamic LDA model in [8]. Each topic's natural parameter chain, ϕ_k, evolves using a random walk model [8]. To generate a document, first map the natural parameter of its topic $\phi_{k,t}$ to the simplex via the logistic transformation in Equation (8-10), and then generate the document, i.e. $x_{t,i}|c_{t,i} = k \sim \mathrm{Multinomial}(\mathrm{x_{t,i}}|\mathrm{Logistic}(\phi_{\mathrm{k,t}}))$.

In Equations (8-10), $C(\phi_{k,t})$ is a normalization constant (i.e., the log partition function). We denote this logistic transformation with the function $Logisitc(.)$. Furthermore, due to the normalizability constrain of the multinomial parameters, $\vec{\beta}_{k,t}$ only has $M-1$ degree of freedom, where M is the vocabulary length. Thus we only need to represent and evolve the first $M-1$ components of $\phi_{k,t}$ and leave $\phi_{k,t} = 0$. For simplicity, we omit this technicality from further consideration.

$$\begin{aligned} \beta_{k,t,m} &= \exp\{\phi_{k,t,m} - C(\phi_{k,t})\}, \quad \forall m = 1, \ldots, M \\ \text{where} \qquad & C(\phi_{k,t}) = \log\Big(\sum_{m=1}^{M} \exp\{\phi_{k,t,m}\}\Big). \end{aligned} \tag{12}$$

One problem with the above construction is the non-conjugacy between the multinomial distribution and the logistic normal distribution. In essence, we can no longer use vanilla RTS smoother to compute the posterior over each chain as required by the Gibbs sampling algorithm in section 5. In [8], numerical techniques were proposed to solve this problem; here, for simplicity, we use a deterministic Laplace approximation to overcome this non-conjugacy problem. We first put the emission of chain ϕ_k at time t in the exponential family representation. It is quite straightforward to show that:

$$\prod_{x \in \phi^x_{k,t}} \prod_{m=1}^{M} p(x_{t,i,m}|\phi_{k,t}) = \exp\{v_{k,t}\phi_{k,t} - |v_{k,t}| \times C(\phi_{k,t})\} \tag{13}$$

where $v_{k,t}$ is an M-dimensional (row) vector that represents the histogram of word occurrences from topic k at time step t . And $|.|$ is the L1 norm of a given vector. Equation (13) still does not represent a Gaussian emission due to the problematic $C(\phi_{k,t})$. Therefore, we approximate it with a second-order quadratic Taylor approximation around $\hat{\phi}_{k,t}$ — to be specified shortly . This results in a linear and quadratic term of $\phi_{k,t}$. If we let H and g to be the hessian and gradient of such expansion, we can re-arrange equation (13) into a gaussian emission with mean $\chi_{\hat{\phi}_{k,t}}$ and covariance $\varphi_{\hat{\eta}_{k,t}}$ given by:

$$\varphi_{\hat{\phi}_{k,t}} = \mathrm{inv}\big(|v_{k,t}|H(\hat{\phi}_{k,t})\big), \tag{14}$$

$$\chi_{\hat{\phi}_{k,t}} = \hat{\phi}_{k,t} + \varphi_{\hat{\eta}^t_k}\big(v_{k,t} - |v_{k,t}|g(\hat{\phi}_{k,t})\big). \tag{15}$$

Using this Gaussian approximation to the non-gaussian emission, we can compute the posterior over $\phi_{k,t}|\phi^{(x)}_{k,t}$ using the RTS smoother with observations, and observation noises as given by Equations (15) and (14) respectively. Due to the high-dimensionality of the associated vectors in this linear state-space model, we approximate the Hessian in the above calculations with its diagonal, which results in an M-independent linear state-space models, one for each word.

Moreover, $\hat{\phi}_{k,t}$ is set to $\text{inverseLogisitic}\left(\frac{\text{v}_{\text{k,t}}}{|\text{v}_{\text{k,t}}|}\right)$, which is the inverse logistic transformation of the MLE (maximum likelihood estimation) of the topic's k multinomial distribution at time t.

We used this simple model to analyze the NIPS12 collection that contains the proceedings of the Neural Information Processing Conference from 1987-1999[2]. Stop words were removed from this collection, we also removed infrequent words and kept only the top most frequent 2000 words. We divided the collection into 13 epochs based on the publication year of the paper. We set the hyperparmaters of the TDPM as in Section 7 with $\alpha = .1$, and we ran Gibbs sampling for 1000 iterations. To speed up convergence, we initialized the sampler from the result of a global non-parametric clustering using the method in [13] which resulted in around 7 clusters, each of which spans the whole 13 years. In figure 4, we display topic durations, which shows that the model indeed captures the death and birth of various topics. In the same figure, we also show the top keywords in some topics (chains) as they evolve over time. As shown in this figure, regardless of the simplicity of the model, it captured meaningful topic evolutions.

8 Relation to Other DPM Approaches:

We have purposefully delayed discussing the relationship between the TDPM and other *dependent* DPM models until we lay down the foundation of our model in order to place the discussion in context. In fact, several approaches have been recently proposed to solve the same fundamental problem addressed in this paper: how to add the notion of time into the DPM. With the exception of [14], most of these approaches use the stick-breaking construction of the DPM [16][17]. In this construction, the DPM is modeled as an infinite mixture model, where each mixture component has a weight associated with it. Coupling the weights and/or (component parameters) of nearby DPMs results in a form of dependency between them. This new process is called ordered-based (Dependent) DPMs. However, we believe that utilizing the CRP directly is easier as we have explained in section 4.2, and more importantly, this approach enables us to model the rich-gets-richer phenomenon, which we believe captures, in a wide range of applications, how a cluster popularity evolves over time. As for the work in [14], we have already explained one difference in section 6. Another difference is that in [14] cluster parameters are fixed and do not evolve over time. Recently, a simple clustering-topic model built on top of [16] was proposed in [18]. This is similar to the experiments we carried in section 8, however, in [18] the cluster (topic) parameters were fixed over time.

9 Conclusions and Future Work

In this paper we presented the temporal Dirichlet process mixture model as a framework for modeling complex longitudinal data. In the TDPM, data is divided into epochs, where the data items within each epoch are partially exchangeable. Moreover, The number of mixture components used to explain the dependency structure in the data is unbounded. Components can retain, die out or emerge over time, and the actual parameterization of each component can also evolve over time in a Markovian fashion. We gave various constructions of the TDPM as well as a Gibbs sampling algorithm for posterior inference. We also showed how to use the

[2] Available from http://www.cs.toronto.edu/ roweis/data.html

TDPM to implement an infinite mixture of Kalman filters as well as a simple non-parametric dynamic topic model.

In the future we plan to explore other techniques for posterior inference like variational inference as in [11,12] and search based techniques [13] that showed promising results in the DPM case and achieved up to 200-300 speedup over using Gibbs sampling. At the application level, we plan to extend the simple non-parametric dynamic topic model into a full-fledged topic model by replacing the Logistic Normal likelihood with another DPM for each document as in [15].

References

[1] Neal, Radford M. 1998. Markov chain sampling methods for Dirichlet process mixture models. Technical Report 9815, University of Toronto, Department of Statistics and Department of Computer Science, September.

[2] Ferguson, Thomas S. 1973. A Bayesian analysis of some nonparametric problems. The Annals of Statistics, 1(2):209-230, March.

[3] Blackwell, David and James B. MacQueen. 1973. Ferguson distributions via Polya urn schemes. The Annals of Statistics, 1(2):353-355, March.

[4] Antoniak, Charles E. 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. The Annals of Statistics, 2(6):1152-1174, November.

[5] Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems. University of Toronto Technical Report CRG-TR-96-2, 1996.

[6] R. Kalman. (1960). A new approach to linear filtering and prediction problems. Transaction of the AMSE: Journal of Basic Engineering, 82:3545.

[7] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. Journal of Machine Learning Research, 3:9931022, January 2003.

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

[9] X. Wang, W. Li, and A. McCallum. A Continuous-Time Model of Topic Co-occurrence Trends. AAAI Workshop on Event Detection, 2006.

[10] X. Wang and A. McCallum. Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends. Conference on Knowledge Discovery and Data Mining (KDD),2006.

[11] Kurihara, Kenichi, Max Welling, and Yee Whye Teh. 2007. Collapsed variational dirichlet process mixture models. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).

[12] Blei, David and Michael I. Jordan. 2005. Variational inference for Dirichlet process mixtures. Bayesian Analysis, 1(1):121-144, August.

[13] Hal Daume, Fast search for Dirichlet process mixture models, Conference on AI and Statistics (2007)

[14] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Time-sensitive Dirichlet process mixture models. Technical Report CMU-CALD-05-104, Carnegie Mellon University, 2005.

[15] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical Dirichlet processes. Journal of the American Statistical Association, 2006. 101(476):1566-1581

[16] J.E. Griffn and M.F.J. Steel. Order-based dependent Dirichlet processes. Journal of the American Statistical Association,101(473).

[17] S. MacEachern. Dependent Dirichlet processes. Technical report, Dept. of Statistics, Ohio State university, 2000.

[18] N. Srebro and S. Roweis. Time-varying topic models using dependent Dirichlet processes. Technical report, Department of Computer Science, University of Toronto, 2005.

[19] E.P. Xing, Dynamic Nonparametric Bayesian Models and the Birth-Death Process. CMU-CALD Technical Report 05-114.

[20] J. McAuliffe, D. Blei, and M. Jordan. Nonparametric empirical Bayes for the Dirichlet process mixture model. Statistics and Computing, 16(1):514, 2006.

[21] Marina Meila. Comparing Clusterings: An Axiomatic View. In Proceedings of the 22nd International Conference on Machine Learning , 2005.

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Carnegie Mellon