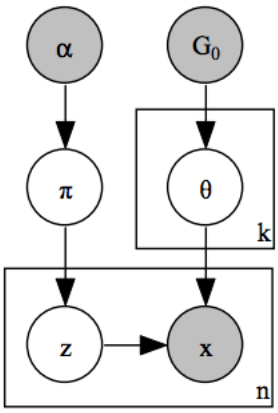
10-708: Probabilistic Graphical Models 10-708, Spring 2014

20: Hierarchical Dirichlet Processes

Lecturer: Eric P. Xing *Scribes: Lavanya Viswanathan, Manaal Faruqui*

1 Stick Breaking Construction



$$\pi \sim Dir\,(\alpha/k)$$
$$\text{For } c = 1, ..., k$$
$$\theta_c \sim G_0$$
$$\text{For } i = 1, ..., n$$
$$z_i \sim Category\,(\pi)$$
$$x_i \sim p\,(x_i|\theta_{z_i})$$

Figure 1: Bayesian Mixture Model

For a Bayesian Mixture Model as shown in figure 1, as $k \to \infty$, we shall have $G = \sum_{c=1}^{\infty} \pi_c \delta_{\phi_c}$, where all the ϕ_c are i.i.d. samples from G_0, while the random sequence $\{\pi_c\}_{c=1}^{\infty}$ sum up to one, shall be constructed by the "Stick Breaking" process [3].

Suppose there is a stick with length 1. Let $\beta_c \sim Beta(1, \alpha)$ for $c = 1, 2, 3, \ldots$, and regard them as fractions we take away from the remainder of the stick every time. Then π_c can be calculated by the length we take away each time.

$$\pi_1 = \beta_1, \pi_2 = (1 - \beta_1)\beta_2, \ldots, \pi_c = \beta_c \prod_{j=1}^{c-1}(1 - \beta_j), \ldots \tag{1}$$

By this construction, we will have $\sum_{c=1}^{\infty} \pi_c = 1$. In this way, we get an explicit construction of G. $\beta_c \sim$

$Beta(1, \alpha)$, $\phi_c \sim G_0$:

$$G = \sum_{c=1}^{\infty} \pi_c \delta_{\phi_c} \tag{2}$$

2 Chinese Restaurant Process

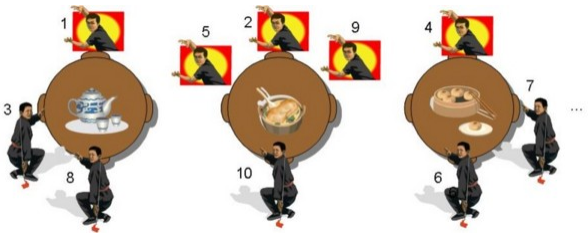


Figure 2: Chinese Restaurant Process

The Chinese restaurant process (CRP) is a distribution of partitions of integers, which was introduced by Pitman and Dubins. Suppose N customers arrive at a restaurant with infinite capacity sequentially. Denote n_i as the number of customers already sitting at table i. Each incoming customer chooses a table at random, with probability that:

$$\begin{aligned} p(i|previousCustomers) &\propto n_i \\ p(nextEmptyTable|previousCustomers) &\propto \alpha \end{aligned} \tag{3}$$

where α is a parameter that is invariable during the process. Note that a new customer can either sit at an existing table, or can start a new table. Figure 2[1] illustrates an example of the Chinese restaurant process, in which each customer is numbered by his arrival sequence. CRP is a representation of partition of integers $1, 2, \ldots N$. For example, in figure 2 the numbers are partitioned into 3 different groups.

In CRP mixture model, each data point is generated as follows: Each table corresponds to a cluster, which is associated with a parameter drawn from a prior $p(\eta^*|\lambda)$. For each customer, we first choose a table $Z \sim CRP(\alpha)$, then we draw a value from $p(x|\eta^*_x)$, e.g, a prior over Gaussian locations. The generative process is similar to mixture models, but with unbounded number of mixture components. Note than an outlier in the data can be thought of as a new cluster. So the emphasis of CRP is not on the actual number of clusters. Given data $\{x_1, x_2, \ldots x_N\}$, the posterior is a distribution on:

1. Number of clusters (number of occupied tables)
2. Data points assigned to each cluster
3. parameter η^*_x of each cluster

Generally the number of clusters is random and unknown and new data can be assigned to a new cluster. Our goal is to estimate this posterior distribution.

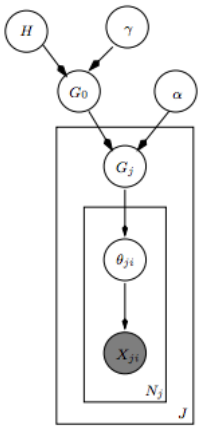


Figure 3: Hierarchical Dirichlet Process

3 Hierarchical Dirichlet Process

A hierarchical Dirichlet process (HDP) is built on multiple Dirichlet processes as shown in figure 3. By adding one more level of DP over G_0, HDP enables data in groups to share countable infinite cluster identities and to exhibit unique cluster propositions. By simply adding a second level of DP over G_0 with concentration parameter γ and base measure H, HDP guaranties the discreteness of G_0. Therefore, HDP mixture models yield exactly the grouped data characteristic [4].

3.1 Chinese Restaurant Franchise

An analog of the Chinese restaurant process for hierarchical Dirichlet processes is the Chinese restaurant franchise. Imagine a franchise of restaurants, serving an infinitely large, global menu. Each table in each restaurant orders a single dish. Let n_{rt} be the number of customers in restaurant r sitting at table t. Let m_{rd} be the number of tables in restaurant r serving dish d. Let m_d be the number of tables, across all restaurants, serving dish d. Customers enter the restaurants, and sit at tables according to the CRP. The first customer enters a restaurant, and picks a table. The n^{th} customer enters the restaurant. He sits at an existing table with probability $\frac{m_k}{n-1+\alpha}$, where m_k is the number of people sitting at table k. He sits at a new table with probability $\frac{\alpha}{n-1+\alpha}$. In addition, each table in each restaurant picks a dish, with probability proportional to the number of times it has been served across all restaurants.

p(table t choose dich d — previous tables) = $\frac{m_d}{T+\gamma}$ for an existing table
p(table t choose dich d — previous tables) = $\frac{\gamma}{T+\gamma}$ for a new table

3.2 Infinite topic Models

A motivation from a different angle that brought about the need for HDP relates to topic modeling. Consider the case where we want to model Latent Dirichlet Allocation (LDA) as a model with infinite topics. This would definitely be advantageous, at the same time it is slightly trickier to deal with infinite topics than dealing with infinite clusters for population inference. This is because, in LDA, when we say we need infinite

[1] `http://cdn-ak.f.st-hatena.com/`

topics, what we really mean is that we want the number of topics to be unbounded. However, we must note that the way topics are used in an LDA context is different from how cluster centroids are used in a clustering scenario. In LDA, the connection is more indirect and we are in fact estimating the per document topic breakdown. Since we deal with "infinite" topics here, we need to have ways of dealing with the scenario wherein, we represent the topics in two documents in the same infinite dimensional space, that is the m^{th} topic in the vector for document A should be the same as the m^{th} topic for document B, even though we are now dealing with infinite dimensional space.

In LDA, each distribution is associated with a distribution over K topics, and the issue here is as to how to choose the number of topics. Choosing infinitely many topics serves as a solution to this issue. And for this, we replace the Dirichlet distribution over topics, by a Dirichlet process.

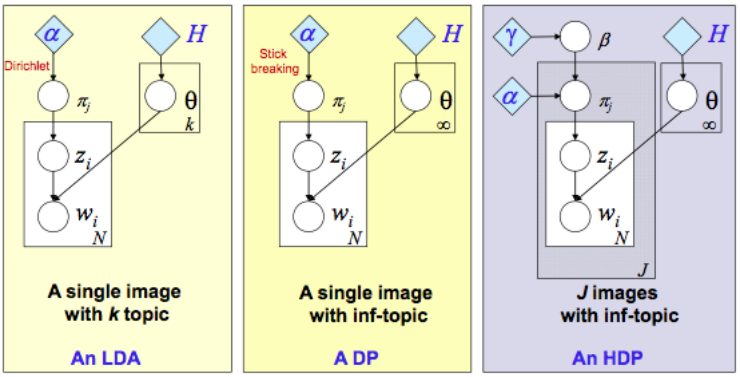


Figure 4: Infinite Topic Models

The typical LDA model (on the left), is modeled as containing N words per document, each word w_i having a topic indicator z_i, and every document having a topic weight vector π_j and containing K topics. When modeled as a DP (center), it is similar to the earlier case; we are going to group words into clusters and each cluster is defined by a word frequency vector, and now we let the number of clusters be infinite. However, one apparent issue here is that after we have dealt with an article with vector π_j, when we sample another $\pi_{j'}$, there is no direct way to get the same ordering of topics as we did for the previous article j. Now, moving to HDP (right), we use the upper level DP to create an infinite pool of topics. Then we can treat every article J by itself as a clustering problem, and every article in this clustering problem is going to draw from the infinite discrete pool the set of topics that is useful to it. This way different documents will have a mechanism of sharing topics with one another, even though the weights corresponding to that topic may vary, the centroid (ordering of topics) would be same. And hence graphically, now the π doesn't just follow a DP, but now follows a HDP so that the same set of infinite topics can be chosen across different topics (same set of topics, but with different weights across different topics). Then we can do appropriate inference on this model. For example, in the Chinese restaurant franchise, we can consider restaurants and dishes as being equivalent to documents and topics respectively. Let H be a V-dimensional Dirichlet distribution, so a sample from H is a distribution over a vocabulary of V words. Then we can sample a global distribution over topics as,

$$G_0 = \sum_{k=1}^{\infty} \pi_k \delta_{\beta_k} \; DP(\alpha, H) \tag{4}$$

And then for each document $m = 1, ..., M$,

- Sample a distribution over topics $G_m \sim DP(\gamma, G_0)$

- For each word $n = 1, ..., N_m$
  - Sample a topic $\phi_{mn} \sim Discrete(G_0)$
  - Sample a word $w_{mk} \sim Discrete(\phi_{mn})$

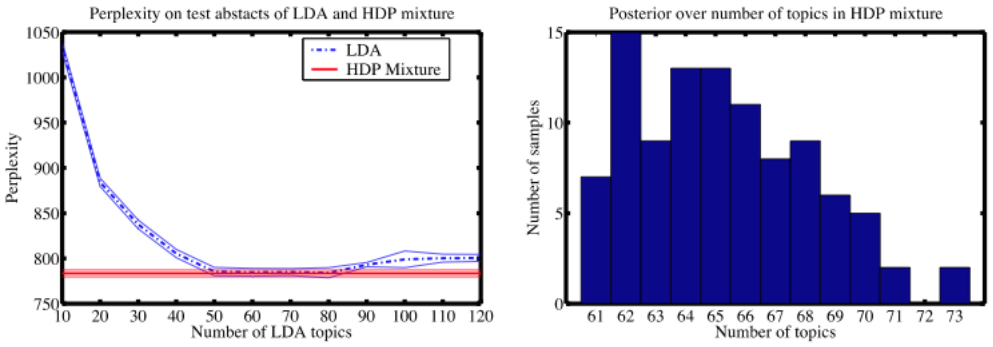


Figure 5: The "right" number of topics

We can then justify the usefulness of this proposed new model by using perplexity as a metric. This is a lenient justification, as we know that there no exact "right" number of topics. We see here (figure 5) that HDP allows us to explore a whole range of the number of topics and determine a best case setup for model complexity.

4 Genetic Inference of the World Population

4.1 Terminology

- Single Nucleotide Polymorphism (SNP): DNA sequence variation due to difference of a single nucleotide.
  - Each DNA site is called a "locus"
  - Each variant is called an allele
- Happlotype: A consecutive sequence of SNP alleles on the same chromosome.
- Genotype: Unordered pair of alleles at a locus present in an individual

4.2 Haplotype Inference

Genotype can be obtained easily but an individual's genotype might not define its haplotype uniquely. For example, a heterozygous diploid organism with genotype T/C T/G A/A could have haplotypes TGA and CTA or TTA and CGA. Given the genotypes for many individuals, haplotype inference can be done by making use of the observation that haplotypes are shared in a population. Thus, given sets of possible haplotypes, we choose the one that uses fewer haplotypes overall.

A Finite Mixture of Allele Model The probability of a genotype g is:

$$p(g) = \sum_{h_1,h_2 \in \mathcal{H}} p(h_1, h_2)p(g|h_1, h_2) \tag{5}$$

where, $\mathcal{H}$ is the population haplotype pool, $p(h_1, h_2)$ is the haplotype model and $p(g|h_1, h_2)$ is the genotyping model. Generally, $|\mathcal{H}| << K = 2^J$ and $p(h_1, h_2) = p(h_1)p(h_2)$.

An Infinite Mixture of Allele Model Since its difficult to infer the parameter values $\mathcal{H}, K$, a statistical model for haplotype inference based on a dirichlet process prior and a likelihood has been proposed called the DP-Haplotyper [5]. In this model each individuals genotype is formed by drawing two random templates from the ancestral pool, and subjected to random perturbations. Posterior inference can be done by an MCMC procedure, that used both Gibbs and Metropolis-hasting updates. This process is analogous to the CRP as follows:

- Associate a population haplotype with a table
- sample α, θ at each table from a base measure G_0 to obtain the population haplotype and nucleotide substitution frequency for that component
- The posterior distribution on the number of population haplotypes can be obtained from $p(h|A, \theta)$ and $p(g|h_1, h_2)$

Inheritance and Observation Models

1. Noisy Observation Model: $H_{i_1}, H_{i_2} \rightarrow G_i$, $P_G(g|h_1, h_2) : g_t = h_{1,t} \oplus h_{2,t}$ with probability λ
2. Single-locus mutation model: $A_{C_{i_e}} \rightarrow H_{i_e}$

$$P_H(h_t|a_t, \theta) = \begin{cases} \theta & if h_t = a_t \\ \frac{1-\theta}{|B|-1} & otherwise \end{cases}$$

MCMC for Haplotype Inference Using Gibbs sampling we can perform inference by drawing samples of each random variable to be sampled given values of all the remaining variables. Integrate out the parameters such as θ or λ and sample C_{i_e}, a_k, and H_{i_e}.

$$p(c_{i_e} = k|\mathbf{c}_{[-i_e]}, \mathbf{h}, \mathbf{a}) \propto p(c_{i_e} = k|c_{[-i_e]})p(h_{i_{e_k}}|a_k, \mathbf{h}_{[-i_e]}, \mathbf{c}) \tag{6}$$

4.3 Multi-population Haplotype Inference using Hierarchical DP Mixture

Now we see how to perform haplotype inference while dealing with multiple populations, such as geographic regions or different ethnicities. We could use the earlier model on each population separately or on the union of all populations. This would however result in our losing valuable information about say population structures implicitly present in the data or the data becoming too fragmented. Thus, we choose to couple the haplotype inference problems using a hierarchical DP mixture model and choose to solve the different clustering problems jointly. To do this, we consider the ancestors of the populations to be shared, but weighted differently. We can consider the ancestors to be drawn for a DP at the top level, providing atoms that then constitute each of the population's DP mixtures.

We refer to the HDP mixture model from figure 3. To describe the process in some detail, in each bottom level urn j (each population), we can choose θ_k with a probability of $\frac{m_{jk}}{\sum_{k'} m_{jk'}+\alpha_0}$ or we may return to the top level urn with a probability $\frac{\alpha_0}{\sum_{k'} m_{jk'}+\alpha_0}$ as in a typical DP. In the top level urn, we can now select θ_k with probability $\frac{n_k}{\sum_{k'} n_{k'}+\gamma}$ or draw a new sample with probability $\frac{\gamma}{\sum_{k'} n_{k'}+\gamma}$. Drawing the θ_k for the top level urn also selects the parameter for the bottom level urn. This two level scheme allows the ancestors to be shared across populations, but with different weights within each population. This approach has been detailed in [6].

Let us now see how we can obtain the general samples for the urns in the two levels. The top level urn follows the Dirichlet process $DP(\gamma, H)$:

$$\theta_i|\theta_{-i} \sim \sum_{k=1}^{K} \frac{n_k}{i+\gamma}\delta_{\phi_k^*}(\theta_i) + \frac{\gamma}{i+\gamma}H(\theta_i) \tag{7}$$

In the bottom level urn we condition on $DP(\gamma, H)$ for the m_jth draw from the mth bottom level urn:

$$\theta_{m_j}|\theta_{-m_j} \sim \sum_{k=1}^{K} \frac{m_{j,k}+\alpha\frac{n_k}{n+\gamma}}{m_j+\alpha}\delta_{\phi_k^*}(\theta_{m_j}) + \frac{\alpha}{m_j+\alpha}\frac{\gamma}{i+\gamma}H(\theta_{m_j}) \tag{8}$$

$$= \sum_{k=1}^{K} p_k^*\delta_{\theta_k^*}(\theta_{m_j}) + p_{K+1}^*H(\theta_{m_j}) \tag{9}$$

5 Dynamic Dirichlet Process

Another dimension of DP that is interesting to explore is modeling time series data. Time series data brings with it some associated challenges: the clustering problem is essentially changing over time and the association of data to cluster centroids would also change over time. The two main ideas that we explore here are:

- Infinite Hidden Markov Model (HMM): a hidden markov DP
- Dependent DP/HDP: directly evolving a DP/HDP

Here we will see a short introduction of the first and a more detailed exploration of the second.

5.1 Infinite HMM

If suppose a HMM has k states, then the transition matrix would have $k * k$ dimensions. If we extend the number of states to infinity, then we would have a matrix whose rows and columns are infinite in number. Treat the matrix as a collection of multinomial distribution. Now each row in the matrix is a multinomial distribution that tells us where we would go at the next time point. And now, when we are not aware as to how many possible states exist in the next time step, we can make that a DP. The issue here is that no matter where we come from at time t, we will move into a set of states that is common (across the source states from time t) at time t+1. That is, the set of target states remains the same irrespective of where we originate from. This sharing of target set across different rows might not be desirable. Now if we just design different independent DPs for each row, then these infinite states will not be shared. That is, the set

of states when moving out of state 1 will not be the same as the set of states while moving out of state 2. We want different DPs for the different transition events to be shared. And we thus need a HDP, wherein all the DPs will be drawn from a base measure which is itself a DP.

Thus infinite HMM is also a HDP, with infinite number of clusters, since we have infinite states to move to out of one single state at time t.

5.2 Dependent DP/HDP

In Dependent DP, we are going to create DPs over time but not in a totally independent way. This problem is sometimes also referred to as Evolutionary Clustering. Here, the cluster components keep changing, even in a fixed dimensionality scenario. This is different from other static clustering techniques in that, here mixture components can die out, new ones can be born and retained mixture component parameters evolve according to Markovian dynamics. This type of approach can be used in Text processing, say for historical data: topics can come into existence or die and topics can change over time.

We now see a simple construction of such an evolving process, once again employing the Chinese Restaurant Process metaphor.

Layout:

- Customers correspond to data points
- Tables correspond to clusters or mixture components
- Dishes correspond to parameter of the mixtures

Taking the usual CRP, we want to be able to use this over time in a recurrent fashion, so that the restaurant that we build at time point t, would depend on the restaurant at time point $t-1$. The restaurant thus operates in epochs: it is taken to be closed at the end of each epoch and that the state of the restaurant at time epoch t depends on that at epoch $t-1$. This is the first order markov dependence, and can be extended to higher-order dependencies.

Lets look at this example, at time point $T = 1$.
Generative Process: Customers at time $T = 1$ are seated as usual:

- Choose table $j \propto N_{j,1}$ and sample $x_i \sim f(\phi_{j,1})$
- Choose a new table $K + 1 \propto \alpha$
  - Sample $\phi_{K+1,1} \sim G_0$ and sample $x_i \sim f(\phi_{K+1,1})$

So we have the customers belonging to different clusters grouped here, and we have the cluster centroids from which the dishes are being sampled.

Now at time point $T = 2$, we consider it to be dependent on the system state at $T = 1$. We can first copy the active (occupied) tables from the previous day (or time point) since they must have been popular to some degree. Each table is annotated with the number of customers it attracted at the previous time point and thus has an associated count of imaginary customers. Thus, when a new customer comes in on day two ($T = 2$), he is not going to make his decision as he would have at $T = 1$ and he would be able to look at how popular a table was the previous day and choose accordingly. He could now choose any of the new existing tables or a new table based on the counts, even though he is the very first customer for day 2. This

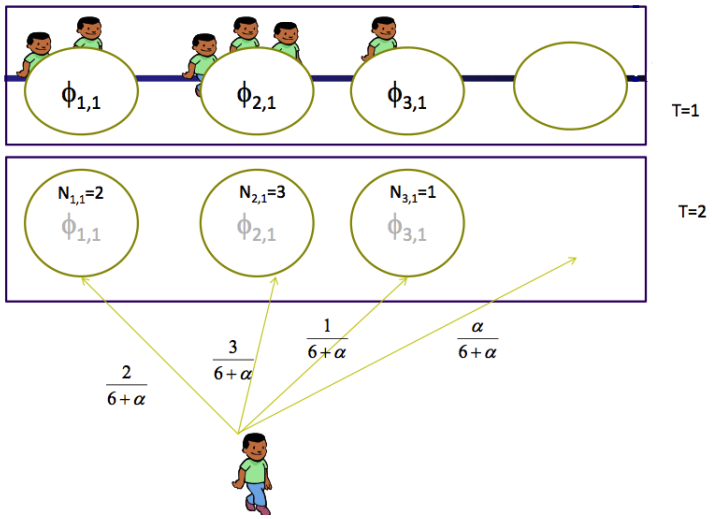


Figure 6: Recurrent Chinese Restaurant Process

will continue for the rest of the customers, and at the end of the day, a sweep to check for active and new clusters will take place. Some existing clusters may die out, and new clusters may spring up. Now counts from day 2 will be carried over as imaginary counts to day 3 and the process continues.

As mentioned earlier, the model can be extended to higher-order dependencies and the dependencies can decay over time. Then the pseudo-counts for table k at time t: $\sum_{w=1}^{W} e^{\frac{-w}{\lambda}} N_{k,t-w}$, where W is the history size, first term is the decay factor and the second term represents the number of customers sitting at table K at time epoch t-w. It can also be that the dishes are not exactly the same across time points, for instance we may use a linear Gaussian model, say Kalman filtering, to model the transformation between $\phi_{1,1}$ and the slightly different $\phi_{1,2}$. This is useful, since the customers that we might get on the second day might vary from the first day and/or the chef may have slight variety in the menu. Another instance, say in the case of topic modeling, though we may talk about the same topic on two consecutive days, say sports, the actual content may vary, for example depending on that day's game.

6 The Big Picture

We earlier reviewed finite mixture models, for instance could be k-means or the EM algorithm. We can extend in the time space, still sticking to finite dimensions, but modeling time series data, say by employing HMM or Kalman filtering type of design. Thus, we will have an evolving topic model or evolving clusters in this setup. We can also grow in the dimensionality space, by replacing the Dirichlet prior with a Dirichlet Process prior. And then we can combine these two to lead to dynamic and infinite dimensional models.

References

[1] Lecture Slides. `http://www.cs.cmu.edu/~epxing/Class/10708/lectures/lecture20-HDP.pdf`.

[2] Scribe Notes from Spring 13. `http://www.cs.cmu.edu/~epxing/Class/10708-13/lecture/scribe24.pdf`.

[3] Jayaram Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[4] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2004.

[5] Eric Xing, Roded Sharan, and Michael I. Jordan. Bayesian haplotype inference via the dirichlet process. In *In Proceedings of the 21st International Conference on Machine Learning*, pages 879–886. ACM Press, 2004.

[6] Eric P. Xing, Kyung-Ah Sohn, Michael I. Jordan, and Yee-Whye Teh. Bayesian multi-population haplotype inference via a hierarchical dirichlet process mixture. In *In Proceedings of the 23st International Conference on Machine Learning*, pages 1049–1056. ACM Press, 2006.