PACHINKO ALLOCATION: DAG-STRUCTURED MIXTURE MODELS OF TOPIC CORRELATIONS

A Dissertation Presented

by

WEI LI

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

April 2007

Department of Computer Science

PACHINKO ALLOCATION: DAG-STRUCTURED MIXTURE MODELS OF TOPIC CORRELATIONS

A Dissertation Presented

by

WEI LI

Approved as to style and content by:

Andrew McCallum, Chair

W. Bruce Croft, Member

Sridhar Mahadevan, Member

John Staudenmayer, Member

David Blei, Member

W. Bruce Croft, Department Chair
Department of Computer Science

ABSTRACT

PACHINKO ALLOCATION: DAG-STRUCTURED MIXTURE MODELS OF TOPIC CORRELATIONS

APRIL 2007

WEI LI

B.Sc., PEKING UNIVERSITY

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Andrew McCallum

Statistical topic models are increasingly popular tools for summarization and manifold discovery in discrete data. However, the majority of existing approaches capture no or limited correlations between topics. I propose the *pachinko allocation* model (PAM), which captures arbitrary, nested, and possibly sparse correlations between topics using a directed acyclic graph (DAG). I present various structures within this framework, different parameterizations of topic distributions, and an extension to capture dynamic patterns of topic correlations. In proposed new work, I will explore two approaches to structure learning: a non-parametric Bayes method based on Dirichlet processes, and a highly-scalable heuristic alternative. The model will be evaluated on document classification, likelihood of held-out data, the ability to

support fine-grained topics, and topical keyword coherence. Proposed applications include information retrieval, hierarchical topic discovery in research papers, social network analysis and semi-supervised learning.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

Statistical topic models have been successfully used to analyze large amounts of textual information in many tasks, including language modeling, document classification, information retrieval, document summarization and data mining. Given a collection of textual documents, parameter estimation in these models discovers a low-dimensional set of multinomial word distributions called "topics". Mixtures of these topics give high likelihood to the training data, and the highest probability words in each topic provide keywords that briefly summarize the themes in the text collection. In addition to textual data (including news articles, research papers and email), topic models have also been applied to images, biological findings and other non-textual multi-dimensional discrete data.

While topic models capture correlation patterns in words, the majority of existing approaches capture none or limited correlations among topics themselves. However, in real-world data, topics are generally not independent from each other. Ignoring these correlations makes an unrealistic assumption, and will limit the ability to discover a large number of fine-grained, tightly-coherent topics. Motivated by the desire to build more accurate models that can extract highly specific topics, I am interested in topic models that explicitly capture topic correlations.

Now I propose the *pachinko allocation* model (PAM), which uses a directed acyclic graph (DAG) structure to represent and learn arbitrary-arity, nested, and possibly sparse topic correlations. In PAM, the concept of topics are extended to be distribu-

tions not only over words, but also over other topics. The model structure consists of an arbitrary DAG, in which each leaf node is associated with a word in the vocabulary, and each non-leaf "interior" node corresponds to a topic, having a distribution over its children. An interior node whose children are all leaves would correspond to a traditional topic. But some interior nodes may also have children that are other topics, thus representing a mixture over topics. With many such nodes, PAM therefore captures not only correlations among words, but also correlations among topics themselves.

Note that the DAG structure in PAM is extremely flexible. It could be a simple tree (hierarchy), or an arbitrary DAG, with cross-connected edges, and edges skipping levels. The nodes can be fully or sparsely connected. The structure could be fixed beforehand or learned from the data. PAM provides a general framework for which several existing models can be viewed as special cases. I present a variety of model structures in Figure 1.1.

In PAM each interior node's distribution over its children could be parameterized arbitrarily. I will investigate various options including multinomial distribution and Dirichlet compound multinomial (DCM). Given a DAG and a parameterization, the generative process samples a topic path for each word. It begins at the root of the DAG, sampling one of its children according to the corresponding distribution, and so on sampling children down the DAG until it reaches a leaf, which yields a word. The model is named for pachinko machines—a game popular in Japan, in which metal balls bounce down around a complex collection of pins until they land in various bins at the bottom.

Similar to many other topic models, PAM extracts and analyzes co-occurrence dependencies in the data from a stationary point of view. Noticeably, many of the large datasets to which topic models are applied to are often collected over time. Topics rise and fall in prominence; they split apart; they merge to form new topics;

words change their correlations. More importantly, topic co-occurrences also change significantly over time, and time-sensitive patterns can be learned from them as well. In order to discover topics that are more localized in time and the evolution patterns in topics and their correlations, I propose an extension of PAM that uses temporal information as observed continuous variables.

While PAM provides a powerful means to describe inter-topic correlations and extract large numbers of fine-grained topics, it also presents a challenge to determine the appropriate DAG size and structure for a particular dataset. Previous work has used cross-validation to estimate the number of topics, but this method is not efficient especially for PAM, which allows arbitrary topic structures. In proposed new work, I will investigate two approaches to automatically learning the structure: a non-parametric Bayes method based on Dirichlet processes, and a highly-scalable heuristic alternative.

In this proposal, I will discuss inference algorithms and parameter estimation for PAM and its variants. I also present experimental results demonstrating PAM's improved performance in three different tasks, including topical word coherence assessed by human judges, likelihood on held-out test data, and document classification accuracy. Future applications include information retrieval, hierarchical topic discovery in research papers, social network analysis, language modeling for speech and semi-supervised learning.

1.2 Contributions

The major contributions of the proposed model are:

1. **General framework to capture topic correlations.** By using a directed acyclic graph, the pachinko allocation model is able to explicitly represent arbitrary correlations between topics. Several existing models can be viewed as its special cases.

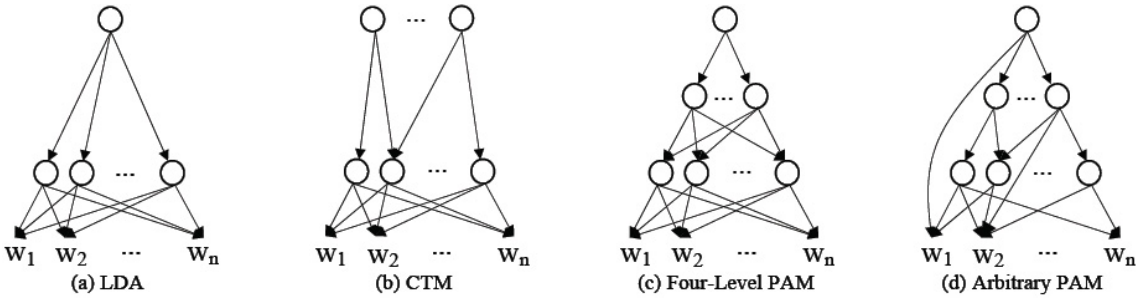


Figure 1.1. Model structures for four topic models (a) LDA: This model samples a multinomial over topics for each document, and then generates words from the topics. (b) CTM: Each topic at the lower level is a multinomial distribution over words and for each pair of them, there is one additional topic that has a distribution over them. (c) Four-Level PAM: A four-level hierarchy consisting of a root, a set of super-topics, a set of sub-topics and a word vocabulary. (d) PAM: An arbitrary DAG structure to encode the topic correlations. Each interior node is considered a topic and has a distribution over its children.

2. **Modeling dynamic properties of topics.** The combined approach of PAM and temporal information can discover not only how topics are correlated, but also when such correlations occur or disappear. Unlike post-hoc analysis that discovers topics without the use of timestamps and then projects their occurrence counts into discretized time, the proposed approach uses temporal information to improve topic discovery.

3. **Automatic structure learning.** While the DAG structure in PAM could be fixed beforehand, it can also be automatically learned from data. Two different approaches are proposed, using Dirichlet processes and a heuristic search algorithm respectively.

4. **Robust likelihood estimation.** For a non-trivial DAG structure in PAM, there is no closed-form solution to evaluate data likelihood. Unlike previous work that has used a harmonic mean method, I propose a more robust estimation technique in the family of empirical likelihood.

5. **Improved topic modeling.** I present empirical results in three different tasks, including topic coherence by human evaluation, likelihood on held-out test data, and document classification. PAM shows improved performance over a variety of topic models.

1.3 Proposal Layout

The remainder of this proposal is organized as follows. First in Chapter 2 is a brief review of related models. Then I describe the pachinko allocation model in Chapter 3, discussing both the general framework and one particular structure and parameterization. In Chapter 4, I present an extension to PAM that incorporates temporal information to study dynamical behaviors of topics. Experimental results are presented in Chapter 5. In Chapter 6, I propose more advanced DAG structures and an alternative parameterization. I will also discuss two approaches to automatic structure learning. Chapter 7 concludes with a timeline to finish the proposed work.

CHAPTER 2

RELATED MODELS

2.1 Statistical Topic Models

2.1.1 Latent Dirichlet allocation

Latent Dirichlet allocation (LDA) [4] is a widely-used topic model, often applied to textual data, and the basis for many variants. LDA represents each document as a mixture of topics, where each topic is a multinomial distribution over words in a vocabulary. To generate a document, LDA first samples a per-document multinomial distribution over topics from a Dirichlet distribution. Then it repeatedly samples a topic from this multinomial and samples a word from the topic. The corresponding graphical model is shown in Figure 3.1(a).

The topics discovered by LDA capture correlations among words, but LDA does not explicitly model correlations among topics. This limitation arises because the topic proportions in each document are sampled from a single Dirichlet distribution. As a result, LDA has difficulty modeling data in which some topics co-occur more frequently than others. However, topic correlations are common in real-world text data, and ignoring these correlations limits LDA's ability to predict new data with high likelihood. Ignoring topic correlations also hampers LDA's ability to discover a large number of fine-grained, tightly-coherent topics. Because LDA can combine arbitrary sets of topics, LDA is reluctant to form highly specific topics, for which some combinations would be "nonsensical".

It is easy to see that LDA can be viewed as a special case of PAM: the DAG corresponding to LDA is a three-level hierarchy consisting of one root at the top, a

set of topics in the middle and a word vocabulary at the bottom. The root is fully connected to all the topics, and each topic is fully connected to all the words. The model structure is shown in Figure 1.1(a). Each topic in LDA has a multinomial distribution over words, and the root has a Dirichlet compound multinomial distribution over topics.

2.1.2 Hierarchical LDA

Hierarchical LDA (HLDA) [1] is a variation of LDA that assumes a hierarchical structure among topics. Each topic has a distribution over words. Topics at higher levels are more general, such as stopwords, while the more specific words are organized into topics at lower levels. To generate a document, it first samples a leaf in the hierarchy. Then for each word in the document, it samples a node on the path from the leaf to the root, and this node generates the word. Thus HLDA can well explain a document that discusses a mixture of *computer science*, *artificial intelligence* and *robotics*. However, for example, the document cannot cover both *robotics* and *natural language processing* under the more general topic *artificial intelligence*. This is because a document is sampled from only one topic path in the hierarchy. Compared to HLDA, PAM provides more flexibility because it samples a topic path for each word instead of each document. Note that it is possible to create a DAG structure in PAM that would capture hierarchically nested word distributions, and obtain the advantages of both models. More details will be discussed in Chapter 6.

2.1.3 Hierarchical Dirichlet Processes

Teh et al. propose hierarchical Dirichlet processes (HDP) to model groups of data that have a pre-defined hierarchical structure [15]. Each pre-defined group is associated with a Dirichlet process, whose base measure is sampled from a higher-level Dirichlet process. HDP can capture topic correlations defined by this nested data structure, however, it does not automatically discover such correlations from

unstructured data. A simple version of HDP does not use a hierarchy over pre-defined groups of data, but can be viewed as an extension to LDA that integrates over (or alternatively selects) the appropriate number of topics. I will present an HDP-based approach to automatically learning the number of topics in PAM in Chapter 6.

2.1.4 Correlated Topic Model

An alternative model that not only represents topic correlations, but also learns them, is the correlated topic model (CTM) [2]. It is similar to LDA, except that rather than drawing topic mixture proportions from a Dirichlet, it does so from a logistic normal distribution, whose parameters include a covariance matrix in which each entry specifies the correlation between a pair of topics. Thus in CTM topics are not independent. Positive result has been reported that CTM performs better than LDA on log-likelihood of held-out test data, and also supports larger numbers of topics.

Pairwise covariance matrix is one way to represent topic correlations. Another possibility is to use mixture models. The model structure of CTM can be described by a special case of PAM, as shown in Figure 1.1(b). The nodes at the lowest level are CTM topics, and for each pair of them, there is one additional node that captures their correlation. One advantage of a mixture model is that it can have fewer parameters. Consider a simple example shown in Figure 2.1. In this example, we need 21 parameters in the covariance matrix while we only need 14 parameters for the mixture model. The advantage is especially obvious when we use a large number of topics because the number of parameters in the covariance matrix grows as the square of the number of topics.

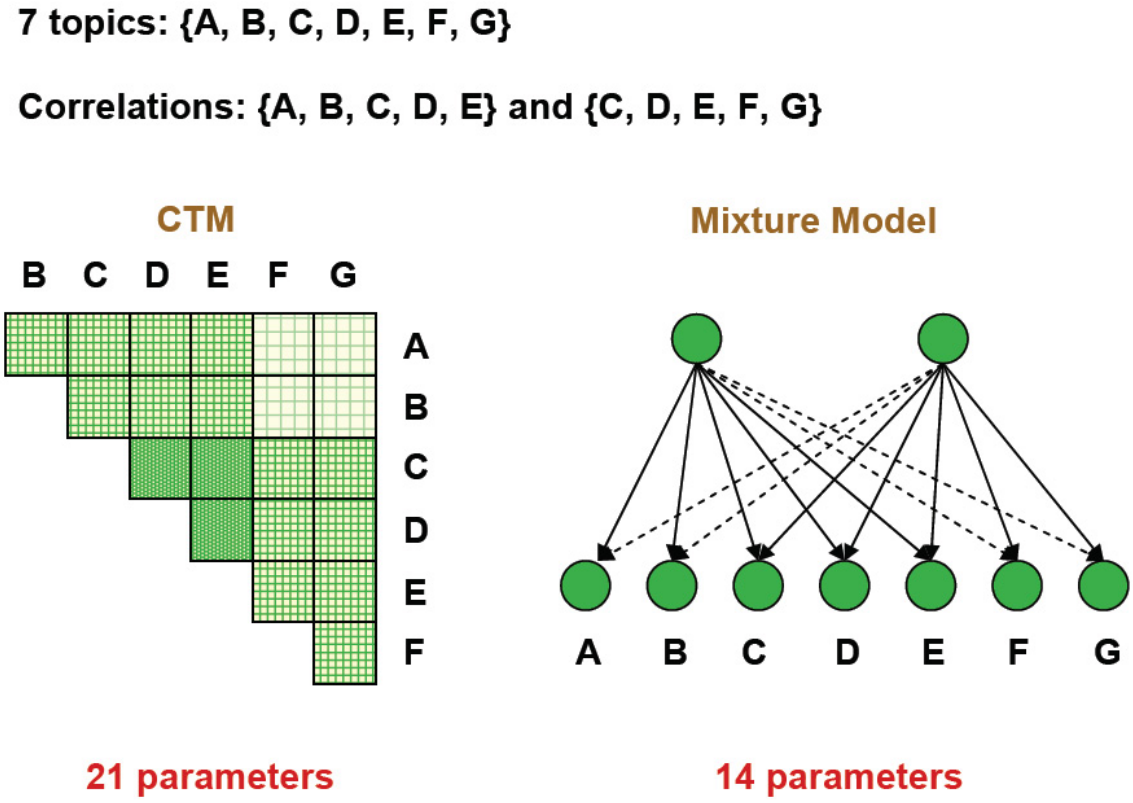


Figure 2.1. An example of topic correlations, which can be represented by both a symmetric covariance matrix (on the left) and a mixture model (on the right). One of the advantages of a mixture model is that it may include fewer parameters.

2.2 Topics Over Time

Several studies have examined topics and their changes across time. Rather than jointly modeling word co-occurrence and time, many of these methods simply use post-hoc or pre-discretized analysis [8, 17, 14].

More recently, time series analysis rooted models have become popular, many of which are based on dynamic models, with a Markov assumption that the state at time $t + 1$ or $t + \Delta t$ is independent of all other history given the state at time t. Hidden Markov models and Kalman filters are two such examples. For instance, Blei and Lafferty present a version of their work of CTM in which the alignment among topics across time steps is modeled by a Kalman filter on the Gaussian distribution

in the logistic normal distribution [3]. This approach employs a Markov assumption over time and it is based on the view that the "meaning" (or word associations) of a topic changes over time.

Another Markov model that aims to find word patterns in time is Kleinberg's "burst of activity model" [10]. This approach uses an infinite-state automaton with a particular state structure in which high activity states are reachable only by passing through lower activity states. Rather than leveraging time stamps, it operates on a stream of data, using data ordering as a proxy for time. Its infinite-state probabilistic automaton has a continuous transition scheme similar to Continuous Time Bayesian Networks (CTBNs) [13]. However, it operates only on one word at a time.

CHAPTER 3

PROPOSED MODEL

In this chapter, I detail the pachinko allocation model (PAM), and describe its generative process, inference algorithm and parameter estimation method.

3.1 General Framework

The notation for the pachinko allocation model is summarized in Table 3.1. PAM connects words in V and topics in T with a DAG structure, where topic nodes occupy the interior levels and the leaves are words. Several possible model structures are shown in Figure 1.1. Each topic t_i is associated with a distribution g_i over its children. In general, g_i could be any distribution over discrete variables, such as logistic normal.

First I describe the generative process for PAM with an arbitrary DAG, assuming that the distributions associated with topics are Dirichlet compound multinomials (DCM). Each distribution g_i is parameterized with a vector α_i, which has the same dimension as the number of children in t_i.

To generate a document d, I use the following two-step process:

1. Sample $\theta_{t_1}^{(d)}$, $\theta_{t_2}^{(d)}$, ..., $\theta_{t_s}^{(d)}$ from $g_1(\alpha_1)$, $g_2(\alpha_2)$, ..., $g_s(\alpha_s)$, where $\theta_{t_i}^{(d)}$ is a multinomial distribution of topic t_i over its children.

2. For each word w in the document,

   - Sample a topic path $\mathbf{z}_w$ of length L_w: $< z_{w1}, z_{w2}, ..., z_{wL_w} >$. z_{w1} is always the root and z_{w2} through z_{wL_w} are topic nodes in T. z_{wi} is a child of $z_{w(i-1)}$ and it is sampled according to the multinomial distribution $\theta_{z_{w(i-1)}}^{(d)}$.

Table 3.1. Notation in PAM.

V	word vocabulary $\{w_1, w_2, ..., w_n\}$
T	a set of topics $\{t_1, t_2, ..., t_s\}$
r	the root
$g_i(\alpha_i)$	Dirichlet distribution associated with topic t_i
d	a document
$\theta_{t_i}^{(d)}$	multinomial distribution sampled from topic t_i for document d
z_{wi}	the ith topic sampled for word w

- Sample word w from $\theta_{z_{wL_w}}^{(d)}$.

Following this process, the joint probability of generating a document d, the topic assignments $\mathbf{z}^{(d)}$ and the multinomial distributions $\theta^{(d)}$ is

$$P(d, \mathbf{z}^{(d)}, \theta^{(d)}|\alpha) = \prod_{i=1}^{s} P(\theta_{t_i}^{(d)}|\alpha_i) \times \prod_{w} (\prod_{i=2}^{L_w} P(z_{wi}|\theta_{z_{w(i-1)}}^{(d)}) P(w|\theta_{z_{wL_w}}^{(d)}))$$

Integrating out $\theta^{(d)}$ and summing over $\mathbf{z}^{(d)}$, we calculate the marginal probability of a document as:

$$P(d|\alpha) = \int \prod_{i=1}^{s} P(\theta_{t_i}^{(d)}|\alpha_i) \times \prod_{w} \sum_{\mathbf{z}_w} (\prod_{i=2}^{L_w} P(z_{wi}|\theta_{z_{w(i-1)}}^{(d)}) P(w|\theta_{z_{wL_w}}^{(d)})) \mathrm{d}\theta^{(d)}$$

Finally, the probability of generating a whole corpus is the product of the probability for every document:

$$P(\mathbf{D}|\alpha) = \prod_{d} P(d|\alpha)$$

3.2 Four-Level PAM

In this section, I will focus on one special case of PAM and describe its inference algorithm and parameter estimation method. It is a four-level hierarchy consisting of one root topic r, s topics at the second level $T = \{t_1, t_2, ..., t_s\}$, s' topics at the third level $T' = \{t'_1, t'_2, ..., t'_{s'}\}$ and words at the bottom. I call the topics at the second level

super-topics and the ones at the third level sub-topics. The root is connected to all super-topics, super-topics are fully connected to sub-topics and sub-topics are fully connected to words (Figure 1.1(c)).

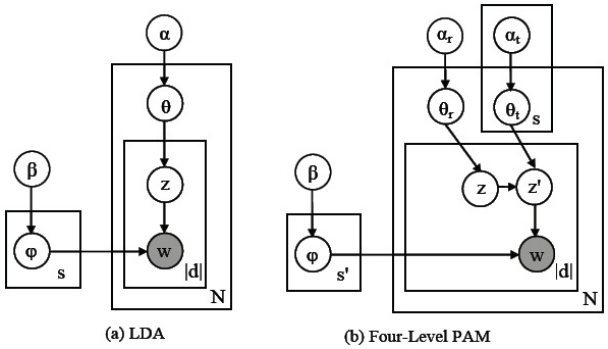


Figure 3.1. Graphical models for (a) LDA and (b) four-level PAM

I use two different distributions for the topics in this structure. In addition to a set of Dirichlet compound multinomials associated with the root $g_r(\alpha_r)$ and super-topics $g_i(\alpha_i)$, the sub-topics are modeled with fixed multinomial distributions $\phi_{t'_j}$, sampled once for the whole corpus from a single Dirichlet distribution $g(\beta)$. The corresponding graphical model is shown in Figure 3.1(b). The generative process for a document d is as follows:

1. Sample $\theta_r^{(d)}$ from the root $g_r(\alpha_r)$, where $\theta_r^{(d)}$ is a multinomial distribution over super-topics.

2. For each super-topic t_i, sample $\theta_{t_i}^{(d)}$ from $g_i(\alpha_i)$, where $\theta_{t_i}^{(d)}$ is a multinomial distribution over sub-topics.

3. For each word w in the document,

   - Sample a super-topic z_w from $\theta_r^{(d)}$.
   - Sample a sub-topic z'_w from $\theta_{z_w}^{(d)}$.
   - Sample word w from $\phi_{z'_w}$.

As we can see, both the model structure and generative process for this special setting are similar to LDA. The major difference is that it has one additional layer of super-topics modeled with Dirichlet compound multinomials, which is the key component capturing topic correlations here. Another way to interpret this structure is that given the sub-topics, each super-topic is essentially an individual LDA. Therefore, it can be viewed as a mixture over a set of LDAs.

Following this process, the joint probability of generating a document d, the topic assignments $\mathbf{z}^{(d)}$ and the multinomial distributions $\theta^{(d)}$ is

$$P(d, \mathbf{z}^{(d)}, \theta^{(d)} | \alpha, \Phi) = P(\theta_r^{(d)} | \alpha_r) \prod_{i=1}^{s} P(\theta_{t_i}^{(d)} | \alpha_i) \times \prod_w (P(z_w | \theta_r^{(d)}) P(z'_w | \theta_{z_w}^{(d)}) P(w | \phi_{z'_w}))$$

Integrating out $\theta^{(d)}$ and summing over $\mathbf{z}^{(d)}$, we calculate the marginal probability of a document as:

$$P(d | \alpha, \Phi) = \int P(\theta_r^{(d)} | \alpha_r) \prod_{i=1}^{s} P(\theta_{t_i}^{(d)} | \alpha_i) \times \prod_w \sum_{\mathbf{z}_w} (P(z_w | \theta_r^{(d)}) P(z'_w | \theta_{z_w}^{(d)}) P(w | \phi_{z'_w})) \mathrm{d}\theta^{(d)}$$

The probability of generating a whole corpus is the product of the probability for every document, integrating out the multinomial distributions for sub-topics Φ:

$$P(\mathbf{D} | \alpha, \beta) = \int \prod_j P(\phi_{t'_j} | \beta) \prod_d P(d | \alpha, \Phi) \mathrm{d}\Phi$$

The hidden variables in the four-level PAM include the sampled multinomial distributions Θ, Φ and topic assignments $\mathbf{z}$. Furthermore, I need to learn the parameters in the Dirichlet compound multinomials $\alpha = \{\alpha_r, \alpha_1, \alpha_2, ..., \alpha_s\}$. I could apply the Expectation-Maximization (EM) algorithm for inference, which is often used to estimate parameters for models involving hidden variables. However, EM has been shown to perform poorly for topic models due to many local maxima.

Instead, I apply Gibbs Sampling to perform inference and parameter learning. For an arbitrary DAG, I need to sample a topic path for each word given other variable assignments enumerating all possible paths and calculating their conditional probabilities. In the special four-level PAM structure, each path contains the root, a super-topic and a sub-topic. Since the root is fixed, I only need to jointly sample the super-topic and sub-topic assignments for each word, based on their conditional probability given observations and other assignments, integrating out the multinomial distributions Θ; (thus the time for each sample is in the number of possible pairs of a super-topic and a sub-topic). The following equation shows corresponding probability. For word w in document d:

$$P(z_w = t_i, z'_w = t'_j | \mathbf{D}, \mathbf{z}_{-w}, \alpha, \beta) \propto \frac{n_{ri}^{(d)} + \alpha_{ri}}{n_r^{(d)} + \sum_{i'} \alpha_{ri'}} \times \frac{n_{ij}^{(d)} + \alpha_{ij}}{n_i^{(d)} + \sum_{j'} \alpha_{ij'}} \times \frac{n_{jw} + \beta_w}{n_j + \sum_m \beta_m}.$$

Here z_w and z'_w correspond to super-topic and sub-topic assignments respectively. $\mathbf{z}_{-w}$ is the topic assignments for all other words. Excluding the current token, $n_r^{(d)}$ and $n_i^{(d)}$ are the numbers of occurrences of the root r and super-topic t_i in document d; $n_{ri}^{(d)}$ is the number of times super-topic t_i is sampled from the root r in document d, and α_{ri} is the corresponding Dirichlet parameter; $n_{ij}^{(d)}$ is the number of times sub-topic t'_j is sampled from the super-topic t_i in document d, and α_{ij} is the corresponding Dirichlet parameter; n_j is the number of occurrences of sub-topic t'_j in the whole corpus and n_{jw} is the number of occurrences of word w in sub-topic t'_j; β_w is the Dirichlet parameter for word w in β.

Note that in the Gibbs sampling equation, I assume that the Dirichlet parameters α are given. While LDA can produce reasonable results with a simple uniform Dirichlet, I have to learn these parameters for the super-topics in PAM since they capture different correlations among sub-topics. As for the root, I assume a fixed Dirichlet parameter. To learn α, I could use maximum likelihood or maximum a posteriori estimation. However, since there are no closed-form solutions for these methods and

I wish to avoid iterative methods for the sake of simplicity and speed, I approximate it by moment matching. In each iteration of Gibbs sampling, I update

$$\begin{aligned} mean_{ij} &= \frac{1}{N} \times \sum_d \frac{n_{ij}^{(d)}}{n_i^{(d)}}; \\ var_{ij} &= \frac{1}{N} \times \sum_d (\frac{n_{ij}^{(d)}}{n_i^{(d)}} - mean_{ij})^2; \\ m_{ij} &= \frac{mean_{ij} \times (1 - mean_{ij})}{var_{ij}} - 1; \\ \alpha_{ij} &\propto mean_{ij}; \\ \sum_j \alpha_{ij} &= \exp(\frac{\sum_j log(m_{ij})}{s' - 1}). \end{aligned}$$

For each super-topic t_i and sub-topic t'_j, I first calculate the sample mean $mean_{ij}$ and sample variance var_{ij}. $n_{ij}^{(d)}$ and $n_i^{(d)}$ are the same as defined above. Then I estimate α_{ij}, the jth component in α_i from sample mean and variance. N is the number of documents and s' is the number of sub-topics.

Smoothing is important when estimating the Dirichlet parameters with moment matching. From the equations above, we can see that when one sub-topic t'_j does not get sampled from super-topic t_i in one iteration, α_{ij} will become 0. Furthermore from the Gibbs sampling equation, we know that this sub-topic will never have the chance to be sampled again by this super-topic. I introduce a prior in the calculation of sample means so that $mean_{ij}$ will not be 0 even if $n_{ij}^{(d)}$ is 0 for every document d.

CHAPTER 4

TOPIC CORRELATIONS OVER TIME

In this chapter, I propose an extension to PAM that captures not only topic correlations but also their changes over time. Some previous work has performed post-hoc analysis to study the dynamical behaviors of topics—discovering topics without the use of timestamps and then projecting their occurrence counts into discretized time [8]— but this misses the opportunity for time to improve topic discovery. A more systematic approach is the state transition based methods [3] using the Markov assumption. Recently, a simple new topics over time (TOT) model is proposed to use temporal information in topic models [16] which represents timestamps of documents as observed continuous variables. A significant difference between TOT and previous work with similar goals is that TOT does not discretize time and does not make Markov assumptions over state transitions in time. Each topic is associated with a continuous distribution over time, and topics are responsible for generating both observed timestamps as well as words.

To generate the words in a document, TOT follows the same procedure as LDA. Each document is represented as a mixture of topics and each topic is a multinomial distribution over a word vocabulary. The mixture components in the documents are sampled from a single Dirichlet distribution. Therefore, TOT focuses on modeling individual topics and their changes over time.

In order to capture the phenomena that topics are correlated and the correlation evolves over time, I introduce a combined approach of PAM and TOT. Each document is associated with one timestamp, but for the convenience of inference [16], I

consider it to be shared by all the words in the document. In order to generate both words and their timestamps, I modify the generative process in the four-level PAM as follows: For each word w in a document d, I still sample a super-topic z_w and a sub-topic z'_w based on multinomial distributions $\theta_r^{(d)}$ and $\theta_{z_w}^{(d)}$. Simultaneously, I also sample timestamps x_w and x'_w from the two topics based on the corresponding Beta distributions Beta(ψ_{z_w}) and Beta($\psi_{z'_w}$).

Now the joint probability of generating a document d, the topic assignments $\mathbf{z}^{(d)}$, the timestamps $\mathbf{x}^{(d)}$ and the multinomial distributions $\theta^{(d)}$ is

$$P(d, \mathbf{z}^{(d)}, \mathbf{x}^{(d)}, \theta^{(d)} | \alpha, \Phi, \Psi) = P(\theta_r^{(d)} | \alpha_r) \prod_{i=1}^{s} P(\theta_{t_i}^{(d)} | \alpha_i) \times$$
$$\prod_w (P(z_w | \theta_r^{(d)}) P(z'_w | \theta_{z_w}^{(d)}) P(w | \phi_{z'_w}) P(x_w | \psi_{z_w}) P(x'_w | \psi_{z'_w}))$$

Integrating out $\theta^{(d)}$ and summing over $\mathbf{z}^{(d)}$, I calculate the marginal probability of a document and its timestamps as:

$$P(d, \mathbf{x}^{(d)} | \alpha, \Phi, \Psi) = \int P(\theta_r^{(d)} | \alpha_r) \prod_{i=1}^{s} P(\theta_{t_i}^{(d)} | \alpha_i) \times$$
$$\prod_w \sum_{\mathbf{z}_w} (P(z_w | \theta_r^{(d)}) P(z'_w | \theta_{z_w}^{(d)}) P(w | \phi_{z'_w}) P(x_w | \psi_{z_w}) P(x'_w | \psi_{z'_w})) \mathrm{d}\theta^{(d)}$$

Finally, the probability of generating a whole corpus with timestamps is the product of the probability for every document, integrating out the multinomial distributions for sub-topics Φ:

$$P(\mathbf{D}, \mathbf{X} | \alpha, \beta, \Psi) = \int \prod_j P(\phi_{t'_j} | \beta) \prod_d P(d, \mathbf{x}^{(d)} | \alpha, \Phi, \Psi) \mathrm{d}\Phi$$

Based on the above generative process, each word is associated with multiple timestamps sampled from different topics. However, there is typically only one timestamp associated with each document. When fitting our model, each training document's timestamp is shared by all the words in the document. But after fitting, if it

actually runs as a generative model, this process would generate different timestamps for every word.

Similar to PAM, I perform Gibbs sampling for inference and the following equation shows the joint probability of a super-topic and a sub-topic. For word w in document d:

$$P(z_w = t_i, z'_w = t'_j | \mathbf{D}, \mathbf{X}, \mathbf{z}_{-w}, \alpha, \beta, \Psi) \propto \frac{n_{ri}^{(d)} + \alpha_{ri}}{n_r^{(d)} + \sum_{i'} \alpha_{ri'}} \times \frac{n_{ij}^{(d)} + \alpha_{ij}}{n_i^{(d)} + \sum_{j'} \alpha_{ij'}} \times \frac{n_{jw} + \beta_w}{n_j + \sum_m \beta_m} \times \frac{(1 - x_w)^{\psi_{i1}-1} {x_w}^{\psi_{i2}-1}}{B(\psi_{i1}, \psi_{i2})} \times \frac{(1 - x'_w)^{\psi_{j1}-1} {x'_w}^{\psi_{j2}-1}}{B(\psi_{j1}, \psi_{j2})}$$

The Beta parameters Ψ are also estimated by the method of moments.

CHAPTER 5

EXPERIMENTAL RESULTS

Experimental results for the four-level PAM and its combination with the topics over time model are discussed in this chapter.

5.1 Four-Level PAM

In this section, I present example topics that PAM discovers from real-world text data and evaluate against LDA using three measures: topic clarity by human judgement, likelihood of held-out test data, and document classification accuracy. I also compare held-out data likelihood with CTM and HDP.

In the experiments I discuss below, I use a fixed four-level hierarchical structure for PAM, which includes a root, a set of super-topics, a set of sub-topics and a word vocabulary. For the root, I always assume a fixed Dirichlet distribution with parameter 0.01. This parameter can be changed to adjust the variance in the sampled multinomial distributions. I choose a small value so that the variance is high and each document contains only a small number of super-topics, which tends to make the super-topics more interpretable. The sub-topics are multinomial distributions sampled once for the whole corpus from a given Dirichlet with parameter 0.01. So the only parameters I need to learn are the Dirichlet parameters for the super-topics, and multinomial parameters for the sub-topics.

In Gibbs sampling for both PAM and LDA I use 2000 burn-in iterations, and then draw a total of 10 samples in the following 1000 iterations. The total training time

for the NIPS dataset (as described in Section 5.1.2) is approximately 20 hours on a 2.4 GHz Opteron machine with 2GB memory.

5.1.1 Topic Examples

The first test dataset comes from Rexa, a search engine over research papers (http://Rexa.info). I randomly choose a subset of abstracts from its large collection. In this dataset, there are 4000 documents, 278438 word tokens and 25597 unique words. Figure 5.1 shows a subset of super-topics in the data, and how they capture correlations among sub-topics. For each super-topic t_i, I rank the sub-topics $\{t'_j\}$ based on the learned Dirichlet parameter α_{ij}. In this graph, each circle corresponds to one super-topic and links to a set of sub-topics as shown in the boxes, which are selected from its top 10 list. The numbers on the edges are the corresponding α values. As we can see, all the super-topics share the same sub-topic in the middle, which is a subset of stopwords in this corpus. Some super-topics also share the same content sub-topics. For example, the topics about *scheduling* and *tasks* co-occur with the topic about *agents* and also the topic about *distributed systems*. Another example is *information retrieval*. It is discussed along with both the *data mining* topic and the *web, network* topic.

5.1.2 Human Judgement

To formally compare the topics discovered by PAM and LDA, I conducted blind topic evaluation. Each of five human evaluators was provided with a set of topic pairs, one each from PAM and LDA, anonymized and in random order. Evaluators were asked to choose which one has stronger sense of semantic coherence and specificity.

These topics were generated using the NIPS abstract dataset (NIPS00-12), which includes 1647 documents, a vocabulary of 11708 words and 114142 word tokens. I use 100 topics for LDA, and 50 super-topics and 100 sub-topics for PAM. The topic pairs are created based on similarity. For each sub-topic in PAM, I find its most similar

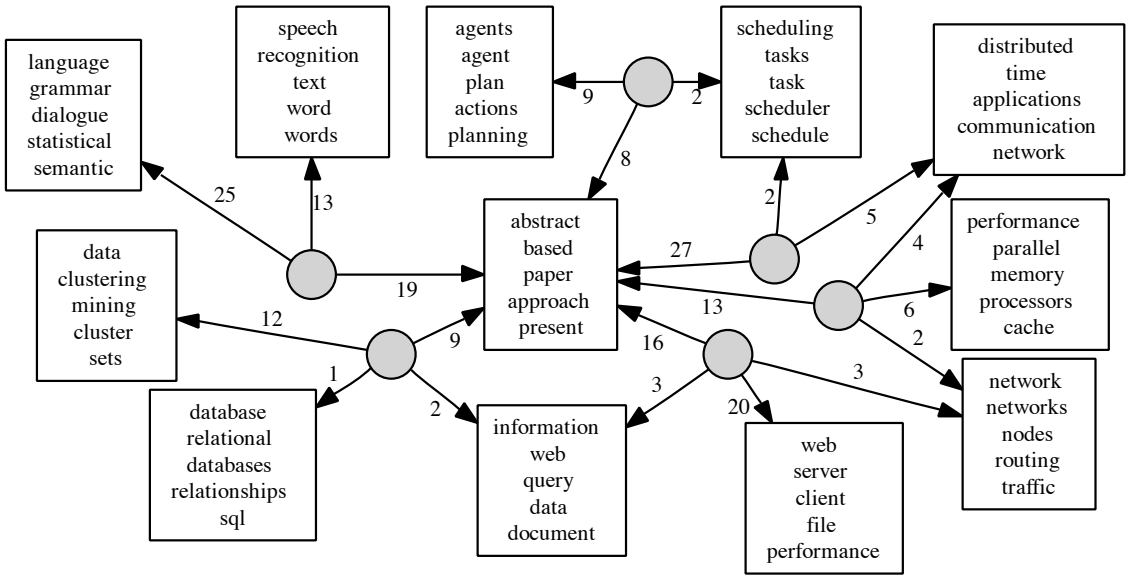


Figure 5.1. Topic correlation in PAM. Each circle corresponds to a super-topic each box corresponds to a sub-topic. One super-topic can connect to several sub-topics and capture their correlation. The numbers on the edges are the corresponding α values for the (super-topic, sub-topic) pair.

topic in LDA and present them as a pair. I also find the most similar sub-topic in PAM for each LDA topic. Similarity is measured by the KL-divergence between topic distributions over words. After removing redundant pairs and dissimilar pairs that share less than 5 out of their top 20 words, I provide the evaluators with a total of 25 pairs. I present four example topic pairs in Table 5.1.2. There are 5 PAM topics that every evaluator agrees to be the better ones in their pairs, while LDA has none. And out of 25 pairs, 19 topics from PAM are chosen by the majority ($\geq$ 3 votes). I show the full evaluation results in Table 5.1.2.

5.1.3 Likelihood Comparison

In addition to human evaluation of topics, I also provide quantitative measurements to compare PAM with LDA, CTM and HDP. In this experiment, I use the same NIPS dataset and split it into two subsets with 75% and 25% of the data respectively.

Table 5.1. Example topic pairs in human judgement.

PAM	LDA	PAM	LDA
control	control	motion	image
systems	systems	image	motion
robot	based	detection	images
adaptive	adaptive	images	multiple
environment	direct	scene	local
goal	con	vision	generated
state	controller	texture	noisy
controller	change	segmentation	optical
5 votes	0 vote	4 votes	1 vote
PAM	LDA	PAM	LDA
signals	signal	algorithm	algorithm
source	signals	learning	algorithms
separation	single	algorithms	gradient
eeg	time	gradient	convergence
sources	low	convergence	stochastic
blind	source	function	line
single	temporal	stochastic	descent
event	processing	weight	converge
4 votes	1 vote	1 vote	4 votes

Table 5.2. Human judgement results. For all the categories, 5 votes, $\geq$ 4 votes and $\geq$ 3 votes, PAM has more topics judged better than LDA.

	LDA	PAM
5 votes	0	5
$\geq$ 4 votes	3	8
$\geq$ 3 votes	9	16

Then I learn the models from the larger set and calculate likelihood for the smaller set. I use 50 super-topics for PAM, and the number of sub-topics varies from 20 to 180.

To calculate the likelihood of held-out data, I must integrate out the sampled multinomials and sum over all possible topic assignments. This problem has no closed-form solution. Previous work that uses Gibbs sampling for inference approximates the likelihood of a document d by the harmonic mean of a set of conditional probabilities $P(d|\mathbf{z}^{(d)})$, where the samples are generated using Gibbs sampling [9]. However, this approach has been shown to be unstable because the inverse likelihood does not have finite variance [5] and has been widely criticized (e.g. [12] discussion).

In our experiments, I employ a more robust alternative in the family of non-parametric likelihood estimates—specifically an approach based on empirical likelihood (EL), *e.g.* [7]. In these methods one samples data from the model, and calculates the empirical distribution from the samples. In cases where the samples are sparse, a kernel may be employed. I first randomly generate 1000 documents from the trained model, based on its own generative process. Then from each sample I estimate a multinomial distribution (directly from the sub-topic mixture). The probability of a test document is then calculated as its average probability from each multinomial, just as in a simple mixture model. Unlike in Gibbs sampling, the samples are unconditionally generated; therefore, they are not restricted to the topic co-occurrences observed in the held-out data, as they are in the harmonic mean method.

I show the log-likelihood on the test data in Figure 5.2, averaging over all the samples in 10 different Gibbs sampling. Compared to LDA, PAM always produces higher likelihood for different numbers of sub-topics. The advantage is especially obvious for large numbers of topics. LDA performance peaks at 40 topics and decreases as the number of topics increases. On the other hand, PAM supports larger numbers of topics and has its best performance at 160 sub-topics. When the number of topics is small, CTM exhibits better performance than both LDA and PAM. However, as I use more and more topics, its likelihood starts to decrease. The peak value for CTM is at 60 topics and it is slightly worse than the best performance of PAM. I also apply HDP to this dataset. Since there is no pre-defined data structure, HDP does not model any topic correlations but automatically learns the number of topics. Therefore, the result of HDP does not change with the number of topics and it is similar to the best result of LDA.

I also present the likelihood for different numbers of training documents in Figure 5.3. The results are all based on 160 topics except for HDP. As we can see, the performance of CTM is noticeably worse than the other three when there is limited

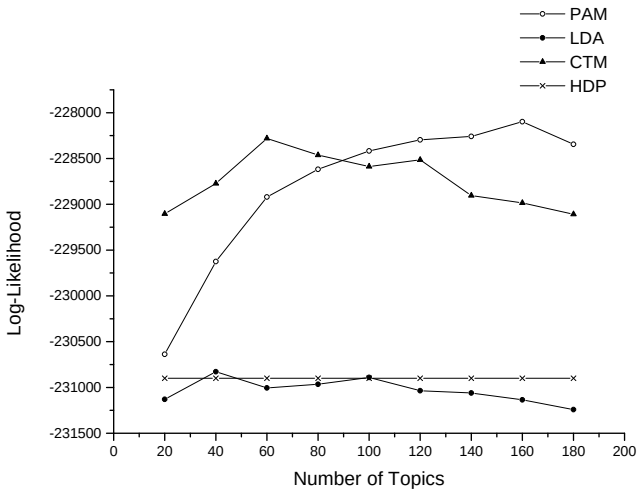


Figure 5.2. Likelihood comparison with different numbers of topics: the results are averages over all samples in 10 different Gibbs sampling and the maximum standard error is 113.75.

amount of training data. One possible reason is that CTM has a large number of parameters to learn especially when the number of topics is large.

5.1.4 Document Classification

Another evaluation comparing PAM with LDA is document classification. I conduct a 5-way classification on the comp subset of the 20 newsgroups dataset. This contains 4836 documents with a vocabulary size of 35567 words. Each class of documents is divided into 75% training and 25% test data. I train a model for each class and calculate the likelihood for the test data. A test document is considered correctly classified if its corresponding model produces the highest likelihood. I present the classification accuracy for both PAM and LDA in Table 5.1.4. According to the sign test, the improvement of PAM over LDA is statistically significant with a p-value < 0.05.

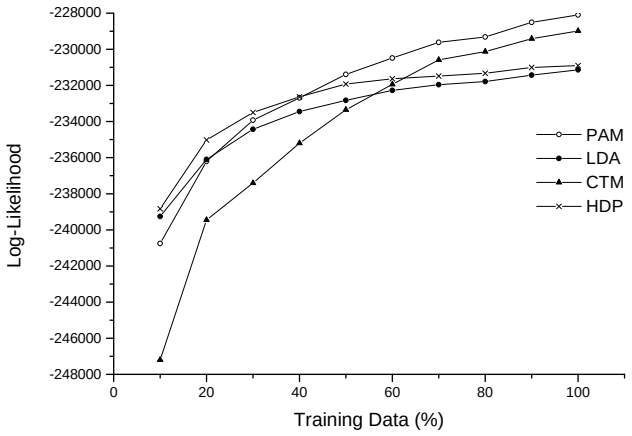


Figure 5.3. Likelihood comparison with different amounts of training data: the results are averages over all samples in 10 different Gibbs sampling and the maximum standard error is 171.72.

Table 5.3. Document classification accuracies (%).

class	# docs	LDA	PAM
graphics	243	83.95	86.83
os	239	81.59	84.10
pc	245	83.67	88.16
mac	239	86.61	89.54
windows.x	243	88.07	92.20
total	1209	84.70	87.34

5.2 PAMTOT

In this section, I present example topics discovered by the PAMTOT model, focusing on the interesting patterns in the evolution of topics and their correlations.

The dataset I use in this experiment is also from Rexa. I choose a subset of paper titles and abstracts that are mostly about machine learning and natural language processing. Then from this subset, I randomly draw 4454 documents spanning from the years 1991 to 2005. For each of the 15 years, there are exactly 300 documents except 1991, for which there were only 254 machine learning documents in the corpus. The overall distribution is therefore close to uniform. After down-casing and removing

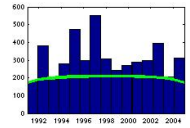
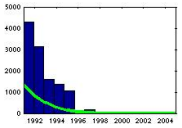
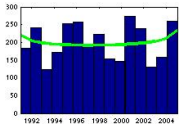
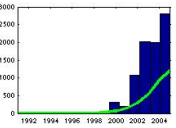
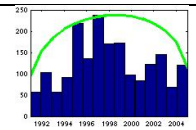
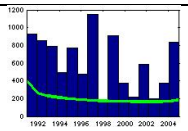
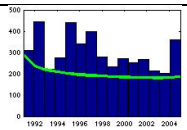
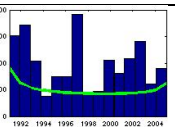
Speech Recognition	SR and Knowledge Rep.		Machine Translation	MT and Ontology	
					
speech	speech	knowledge	language	language	semantic
word	word	process	translation	translation	ontology
words	words	domain	natural	natural	description
corpus	corpus	case	english	english	xml
statistical	statistical	conceptual	machine	machine	ontologies
part	part	processes	languages	languages	rdf
semantic	semantic	concepts	sentence	sentence	documents
context	context	base	sentences	sentences	schema
recognition	recognition	construction	target	target	annotation
automatic	automatic	categories	source	source	concepts
					
speech	speech	knowledge	language	language	semantic
spoken	spoken	database	natural	natural	ontology
recognition	recognition	databases	translation	translation	description
speaker	speaker	domain	processing	processing	annotation
adaptation	adaptation	sources	sentence	sentence	concepts
vocabulary	vocabulary	discovery	linguistic	linguistic	ontologies
dependent	dependent	base	generation	generation	rdf
acoustic	acoustic	conceptual	machine	machine	terms
items	items	relational	syntactic	syntactic	descriptions
short	short	large	languages	languages	semantics

Figure 5.4. Two examples discovered by PAMTOT (above) and PAM (bottom) from the Rexa dataset. Each example consists of a sub-topic and a super-topic. The titles are our own interpretation of the topics. Histograms show how the topics are distributed over time; the fitted Beta PDFs is shown also. (For PAM, Beta distributions are fit in a post-hoc fashion). For sub-topics, I list the top words below the histograms. For super-topics, I list the top words for their child topics.

stopwords, I obtain a total set of 372936 word tokens and 21748 unique words. The DAG structure is the same as I described in Section 1.

I show two example trends in Figure 5.4. Each column corresponds to one sub-topic or super-topic. The titles are my own interpretation of the topics. The Beta distributions over time and their actual histograms over time are displayed in the graphs. I also list the 10 most likely words for each sub-topic and the highly correlated

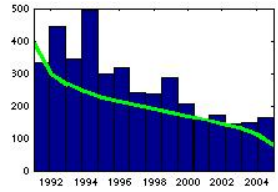
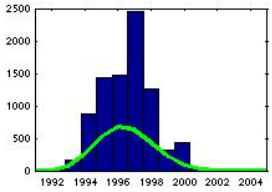
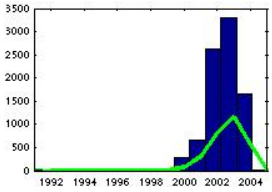
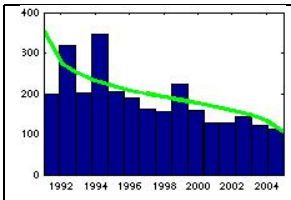
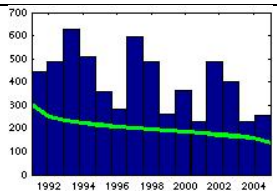
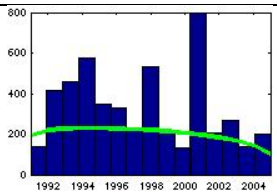
Genetic Algorithms	GA and Neural Networks		GA and SVM	
				
genetic	genetic	network	genetic	kernel
optimization	optimization	networks	optimization	vector
evolutionary	evolutionary	neural	evolutionary	support
problems	problems	input	problems	online
algorithms	algorithms	output	algorithms	svm
population	population	hidden	population	music
evolution	evolution	units	evolution	principal
fitness	fitness	inputs	fitness	pca
ga	ga	layer	ga	semi
global	global	training	global	profiles
				
genetic	genetic	network	genetic	classification
evolutionary	evolutionary	networks	evolutionary	training
evolution	evolution	neural	evolution	classifier
population	population	input	population	classifiers
ga	ga	training	ga	vector
fitness	fitness	abstract	fitness	set
selection	selection	layer	selection	multiple
mutation	mutation	architecture	mutation	svm
crossover	crossover	units	crossover	support
operators	operators	inputs	operators	binary

Figure 5.5. Another example showing a pattern discovered by PAMTOT. The first column is a sub-topic and the other two columns correspond to two parent super-topics that capture its correlations with other topics.

children of each super-topic. As a comparison, I also show their most similar PAM topics at the bottom, decided by KL-divergence. The time analysis for PAM topics is done post-hoc.

The first column in Figure 5.4 demonstrates how the sub-topic "Speech Recognition" changes over time. As we can see, this topic has a relatively smooth distribution between year 1991 and 2005. More interestingly, as shown in the second column, a super-topic that captures the correlation between "Speech Recognition" and another topic "Knowledge Representation" has more dramatic changes in the frequency. These two topics are well connected before 1994 and the correlation gradually ceases after 1998. My understanding is that other techniques for speech recognition have become more popular than knowledge-based approaches. At the bottom of these two columns, I show the corresponding sub-topic and super-topic discovered by PAM without time information. While the time distribution of "Speech Recognition" remains almost the same, PAM alone cannot discover the pattern of the correlation between these two topics.

The two columns on the right show another example. "Machine Translation" has been a popular topic over the entire time period. On the other hand, "Ontologies" is a relatively new subject. We see increasing correlation between them from year 2000. Again, without time information, PAM does not pick up this trend clearly.

Figure 5.5 shows another pattern that becomes clearer when we analyze two super-topics at the same time. The first column corresponds to the sub-topic "Genetic Algorithms". Its frequency has been slowly decreasing from year 1991 to 2005. Its connection with other topics are more localized in time. As shown by the second and third columns, it co-occurs more often with "Neural Networks" around 1996, and from 2000, the focus has shifted to other topics like "Support Vector Machines". This pattern reflects the popularities of these techniques in different years. I cannot capture this trend by PAM and post-hoc analysis of time. As the graphs at the bottom show, we can only see slight decrease of the correlation between "Genetic Algorithms" and "Neural Networks" over the years, and also the connection with

Table 5.4. Errors and accuracies of time (publishing year) predictions for PAMTOT

	L1 Error	E(L1)	Accuracy
PAMTOT	1.56	1.57	0.29
PAM	5.34	5.30	0.10

"Support Vector Machines" has too much noise to exhibit any interesting pattern over time.

One interesting feature of this approach (and one not shared by state-transition-based Markov models of topical shifts) is the capability of predicting the timestamp given the words in a document. This task also provides another opportunity to quantitatively compare PAMTOT against PAM.

On the Rexa dataset, I present the ability to predict the publishing year given the text of the abstract of a paper, as measured in accuracy, L1 error (the difference between predicted and true years) and expected L1 distance to the correct year (average differences between all years and true year). As shown in Table 5.4, PAMTOT achieves almost triple the accuracy of PAM, and provides an L1 relative error reduction of 70%.

CHAPTER 6

PROPOSED WORK

In this chapter, I describe future work including using alternative DAG structures and parameterizations, structure learning approaches, and possible applications.

6.1 Alternative Structures and Parameterizations

While PAM allows arbitrary DAGs to capture topic correlations, I have been focusing on a special four-level hierarchical structure in the experiments. Now I will propose a more advanced structure that combines the advantages of PAM and HLDA.

In the four-level PAM, only sub-topics are responsible for generating words, while the root and super-topics do not have distributions over words. One disadvantage of this setting is that generic words are also mixed into sub-topics. On the other hand, hierarchical LDA provides a better solution. In HLDA, each topic in the hierarchy has a distribution over words. Topics at higher levels are more general, such as stopwords, while the more specific words are organized into topics at lower levels. However, the generative process of HLDA is restricted to sampling a document from one particular path in the hierarchy. Therefore, it is unable to generate documents that discuss several topics from different paths. The model structure of HLDA can be represented as a DAG shown in Figure 6.1(a). The lower part in the DAG corresponds to the topic hierarchy in HLDA. For each leaf, there is one additional node that has a distribution over nodes on the path from the leaf to the root. Furthermore, the Dirichlet distribution associated with the node at the top needs to have a high

variance so that in each document, only one of its children will be sampled for the entire document, which specifies one particular path in the topic hierarchy of HLDA.

Figure 6.1(b) shows an extension to HLDA, where there are additional layers of nodes that represent mixtures over topic leaves. Under this structure, it is possible to generate documents that cover a set of topics from multiple paths while still maintaining the advantage of HLDA to discover topics with different levels of granularity. One of the proposed work is to implement the inference algorithm and parameter estimation method under this model structure, and apply it to real-world text data.

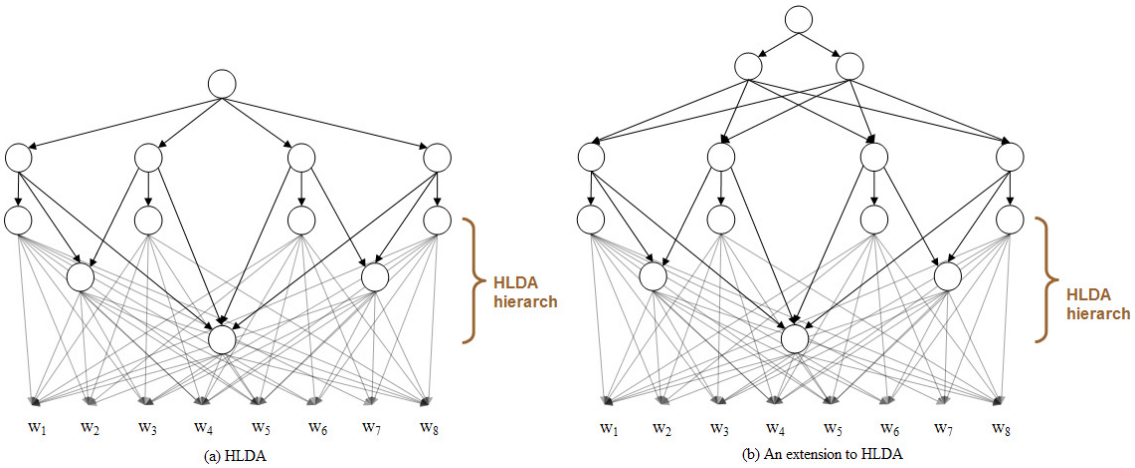


Figure 6.1. Model structures for hierarchical LDA and its combination with PAM. (a) HLDA: The lower part corresponds to the topic hierarchy in HLDA and for each topic leaf, there is one additional node that has a distribution over the nodes on the path from the leaf to the root. (b) An extension to HLDA: Additional layers of topics are used to model mixtures over topic leaves in HLDA, therefore, it is no longer restricted to sampling a document from one particular path in the hierarchy.

In addition to using more advanced DAG structures, another direction of future work is to experiment with different parameterizations. For example, in the four-level PAM, the topic distributions over words are fixed multinomials sampled once for the whole corpus from a single Dirichlet. An alternative distribution is the Dirichlet compound multinomial, which has recently been proposed by [11] to model topic

distributions. Since the root and super-topics are already modeled with DCMs, the distinction between these topics and sub-topics will be decreased in this way.

6.2 Non-parametric Bayes Approach to Structure Learning

6.2.1 The Model

While PAM provides a powerful means to describe inter-topic correlations and extract large numbers of fine-grained topics, it has the same practical difficulty as many other topic models, i.e., how to determine the number of topics. It could be estimated using cross-validation, but this method is not efficient even for simple topic models like LDA. Since PAM has a more complex topic structure, it is more difficult to evaluate all possibilities and select the best one.

Another approach to this problem is to automatically learn the number of topics with hierarchical Dirichlet processes (HDP) [15]. HDP is intended to model groups of data that have a pre-defined hierarchical structure. Others have used it as an extension of LDA where it integrates over (or alternatively selects) the appropriate number of topics.

Here I propose a combined approach of pachinko allocation model and hierarchical Dirichlet processes to automatically learn topic structures from data. I assume a hierarchical DAG structure for PAM and model each topic with a Dirichlet process. Then the Dirichlet processes at the same level are further organized into one individual HDP, which is used to estimate the number of topics at this level. Similar to the generative process of PAM, each word is associated with a topic path. The topic assignments essentially provide a hierarchical grouping of data. Therefore, the topic sampled from a upper-level HDP is used by lower-level HDPs to sample more topics. In other words, the overall structure of this model is an example of nested hierarchical Dirichlet processes.

First consider the four-level PAM where the numbers of super-topics and sub-topics need to be specified beforehand. Now I combine it with hierarchical Dirichlet processes to automatically learn the numbers. I describe the generative process in terms of Chinese restaurant process.

The Chinese restaurant process (CRP) is one way to describe the sampling procedure from a Dirichlet process. It assumes a Chinese restaurant with an infinite number of tables. When a customer comes, he sits at a table with the following probabilities:

$P(\text{an occupied table } t) = \frac{C(t)}{\sum_{t'} C(t')+\alpha}$,

$P(\text{an unoccupied table}) = \frac{\alpha}{\sum_{t'} C(t')+\alpha}$,

where $C(t)$ is the number of customers sitting at table t, $\sum_{t'} C(t')$ is the total number of customers in the restaurant and α is a parameter.

After all customers sit down, I obtain a distribution over tables. This distribution can be viewed as a sample from a Dirichlet process. In HDP, the Chinese restaurant process is extended to the Chinese restaurant franchise (CRF), where each table is associated with a dish. When a customer sits at a previously unoccupied table, a dish will be assigned to the table with the following probabilities:

$P(\text{an existing dish } d) = \frac{C(d)}{\sum_{d'} C(d')+\gamma}$,

$P(\text{a new dish}) = \frac{\gamma}{\sum_{d'} C(d')+\gamma}$,

where $C(d)$ is the number of tables serving dish d, $\sum_{d'} C(d')$ is the total number of tables and γ is a parameter.

Now I describe the generative process of PAM-HDP as a nested Chinese restaurant franchise. The upper-level is a two-layer CRF. I use the new notation of *entry* and *category* . The lower-level is a three-layer CRF. In addition to *table* and *dish* , I introduce a new layer of *menu* between them.

Each restaurant has an infinite number of entries, each of which has a category associated with it. When a customer enters an entry, he implicitly selects the category

Table 6.1. Notation in PAM-HDP.

r_j	the jth restaurant
e_{jk}	the kth entry in the jth restaurant
c_l	the lth category. Each entry is associated with a category.
t_{jln}	the nth table in the jth restaurant that has category c_l.
m_{lp}	the pth menu in category c_l. Each table is associated with a menu from the corresponding category.
d_m	the mth dish in the global set of dishes. Each menu is associated with a dish.

he wants too, which will affect his choice of table and menu later. In every restaurant, there are an infinite number of tables associated with every category. A customer can only sit at a table with the same category as the entry he chooses. Furthermore, every category has its own set of menus, and they are shared among different restaurants. There is a menu for each table, selected from the corresponding category. And for each menu, there is a dish associated with it. All the dishes are globally shared among different menus, no matter what categories they belong to. If a customer sits at a table that already has other customers, he shares the same menu and thus dish with these customers. When he wants to sit at a new table, a menu is assigned to the table. If the menu is new too, a dish will be assigned to it.

Now I briefly explain the problem again using the notation in PAM. I assume that there are infinite numbers of super-topics and sub-topics, and model them with nested hierarchical Dirichlet processes. The upper-level HDP determines how to sample the super-topic (i.e., the "category"). Then the super-topic points to another three-layer HDP that samples the sub-topic (the "dish").

I summarize the notation in Table 6.1.

The generative process is described as follows.

A customer x arrives at restaurant r_j.

He chooses the kth entry e_{jk} in the restaurant with probabilities

(1) e_{jk} is an existing entry: $\frac{C(j,k)}{\sum_{k'} C(j,k')+\alpha_0}$

(2) e_{jk} is new: $\frac{\alpha_0}{\sum_{k'} C(j,k')+\alpha_0}$

$C(j,k)$ is the number of customers that entered the kth entry before in this restaurant. $\sum_{k'} C(j,k')$ is the total number of customers in the restaurant.

If the customer enters a new entry, we assign category c_l for it with probabilities

(1) c_l is an existing category: $\frac{\sum_{j'} C(l,j')}{\sum_{j'}\sum_{l'} C(l',j')+\gamma_0}$

(2) c_l is new: $\frac{\gamma_0}{\sum_{j'}\sum_{l'} C(l',j')+\gamma_0}$

$C(l,j)$ is the number of entries that have category c_l in restaurant r_j.$\sum_{j'} C(l,j')$ is the total number of entries in all restaurants that have category c_l. $\sum_{j'}\sum_{l'} C(l',j')$ is the total number of entries in all restaurants.

After choosing the category, the customer makes the decision for which table he will sit at. The probability that he chooses t_{jln} is

(1) t_{jln} is an existing table: $\frac{C(j,l,n)}{\sum_{n'} C(j,l,n')+\alpha_1}$

(2) t_{jln} is new: $\frac{\alpha_1}{\sum_{n'} C(j,l,n')+\alpha_1}$

$C(j,l,n)$ is the number of customers sitting at table t_{jln}. $\sum_{n'} C(j,l,n')$ is the number of customers in restaurant r_j that choose category c_l.

If the customer sits at an existing table, he will share the menu and dish with other customers at the same table. Otherwise, he will choose a menu for the new table from $\{m_{l1}, m_{l2}, ...\}$. The probability to sample m_{lp} is proportional to

(1) m_{lp} is an existing menu: $\frac{\sum_{j'} C(j',l,p)}{\sum_{j'}\sum_{p'} C(j',l,p')+\gamma_1}$

(2) m_{lp} is new: $\frac{\gamma_1}{\sum_{j'}\sum_{p'} C(j',l,p')+\gamma_1}$

$C(j,l,p)$ is the number of tables in restaurant j that have menu m_{lp}. $\sum_{j'} C(j',l,p)$ is the number of tables in all restaurants that have menu m_{lp}. $\sum_{j'}\sum_{p'} C(j',l,p')$ is the total number of tables associated with category c_l in all restaurants.

If the customer gets an existing menu, he will eat the dish on the menu. Otherwise, he needs to sample a dish for the new menu. The probability that dish d_m is picked is proportional to

(1) d_m is an existing dish: $\frac{\sum_{l'} C(l',m)}{\sum_{m}\sum_{l'} C(l',m)+\sigma_1}$

(2) d_m is new: $\frac{\sigma_1}{\sum_{m}\sum_{l'} C(l',m)+\sigma_1}$

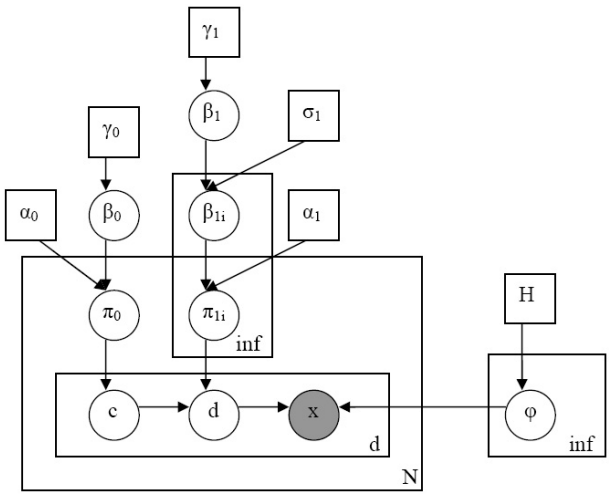


Figure 6.2. Graphical model for PAM-HDP

$C(l, m)$ is the number of menus in category c_l that serve dish d_m. $\sum_{l'} C(l', m)$ is the number of menus that serve dish d_m. $\sum_m \sum_{l'} C(l', m)$ is the total number of menus in all categories.

α_0, α_1, γ_0, γ_1 and σ_1 are all parameters in Dirichlet processes.

The corresponding graphical model is shown in Figure 6.2.

Similar to PAM and many other topic models, I use Gibbs sampling to perform inference. For each customer x, I want to jointly sample the 5 variables associated with it. Assume x wants to sit at restaurant r_j, then the conditional probability that he chooses entry e_{jk}, category c_l, table t_{jln}, menu m_{lp} and dish d_m given observations X and other variable assignments (denoted by Π_{-x}) is:

$$P(e_{jk}, c_l, t_{jln}, m_{lp}, d_m | x, X_{-x}, \Pi_{-x}) \propto$$
$$P(e_{jk}, c_l | \Pi_{-x}) P(t_{jln}, m_{lp}, d_m | \Pi_{-x}, c_l) P(x | X_{-x}, \Pi_{-x}, d_m)$$

Now I discuss the three terms in the above equation individually.

$P(e_{jk}, c_l | \Pi_{-x}) \propto$

(1) e_{jk} is an existing entry and c_l is the category associated with it:

$\frac{C(j,k)}{\sum_{k'} C(j,k')+\alpha_0}$

(2) e_{jk} is new and c_l is an existing category:

$\frac{\alpha_0}{\sum_{k'} C(j,k')+\alpha_0} \frac{\sum_{j'} C(l,j')}{\sum_{j'} \sum_{l'} C(l',j')+\gamma_0}$

(3) e_{jk} and c_l are both new:

$\frac{\alpha_0}{\sum_{k'} C(j,k')+\alpha_0} \frac{\gamma_0}{\sum_{j'} \sum_{l'} C(l',j')+\gamma_0}$

(4) other cases: 0

Note that the number of non-zero probabilities here is only the sum of the numbers of existing entries and categories instead of their product. Similarly,

$P(t_{jln}, m_{lp}, d_m | \Pi_{-x}, c_l) \propto$

(1) t_{jln} is an existing table, m_{lp} is the menu associated with it and d_m is the dish associated with m_{lp}:

$\frac{C(j,l,n)}{\sum_{n'} C(j,l,n')+\alpha_1}$

(2) t_{jln} is new, but m_{lp} is an existing menu and d_m is the dish associated with it:

$\frac{\alpha_1}{\sum_{n'} C(j,l,n')+\alpha_1} \frac{\sum_{j'} C(j',l,p)}{\sum_{j'} \sum_{p'} C(j',l,p')+\gamma_1}$

(3) both t_{jln} and m_{lp} are new and d_m is an existing dish:

$\frac{\alpha_1}{\sum_{n'} C(j,l,n')+\alpha_1} \frac{\gamma_1}{\sum_{j'} \sum_{p'} C(j',l,p')+\gamma_1} \frac{\sum_{l'} C(l',m)}{\sum_{m} \sum_{l'} C(l',m)+\sigma_1}$

(4) all three variables are new:

$\frac{\alpha_1}{\sum_{n'} C(j,l,n')+\alpha_1} \frac{\gamma_1}{\sum_{j'} \sum_{p'} C(j',l,p')+\gamma_1} \frac{\sigma_1}{\sum_{m} \sum_{l'} C(l',m)+\sigma_1}$

(5) other cases: 0

Again, the number of non-zero probabilities is not the product of the numbers of possible values for the three variables. Lastly I estimate the probability to sample an observation x. Assume that the base distribution H is a symmetric Dirichlet distribution with parameter α_2, then,

$P(x|X_{-x}, \Pi_{-x}, d_m) \propto \frac{C(m,x)+\alpha_2}{\sum_{x'}(C(m,x')+\alpha_2)}$

$C(m, x)$ is the number of times that menu d_m is assigned to customer x.

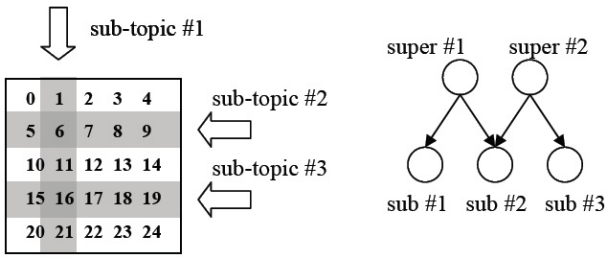


Figure 6.3. An example of synthetic data

6.2.2 Preliminary Results

Preliminary experiments are described below, and a thorough evaluation will be performed in the future.

6.2.2.1 Synthetic dataset

I first apply this model to synthetic datasets, evaluating the discovered topic structures against the true structures. When I generate the training data, the word vocabulary is organized into a v-by-v grid. Each sub-topic is a uniform distribution over either a column or a row of words. A super-topic is an arbitrary combination of sub-topics. An example is shown in Figure 6.3. Then I follow the same generative process described above to randomly sample documents. By changing the values of parameters α_0, γ_0, α_1, γ_1 and σ_1, I obtain different datasets where each of them consists of 100 documents and each document contains 200 tokens. I compare the discovered structures with true structures in Table 6.2.2.1. Each sub-topic lists the words that occur at least 10 times and each super-topic lists all the children and the number of their occurrences. For example, in the true structure of the first dataset, sub-topic #0 contains words 1, 6, 11, 16, and 21. Super-topic #0 has two children, 6820 times choosing sub-topic #0 and 3209 times choosing #1.

The structure in the first dataset is simple. I have a 5-by-5 vocabulary and our model clearly identifies all the sub-topics and super-topics. The second dataset is more complicated. The grid size is increased to 10 and the numbers of super-topics

Table 6.2. Synthetic experiment results

	true	learned
sub-topics	0: 1, 6, 11, 16, 21 1: 15, 16, 17, 18, 19 2: 0, 1, 2, 3, 4 3: 2, 7, 12, 17, 22	0: 1, 6, 11, 16, 21 1: 15, 16, 17, 18, 19 2: 0, 1, 2, 3, 4 3: 2, 7, 12, 17, 22
super-topics	0–0: 6820 0–1: 3209 1–2: 5802 1–3: 4169	0–0: 6819 0–1: 3208 1–0: 16 1–2: 5757 1–3: 4200
sub-topics	0: 9, 19, ..., 99 1: 20, 21, ..., 29 2: 90, 91, ..., 99 3: 6, 16, ..., 96 4: 50, 51, ..., 59 5: 0, 1, ..., 9 6: 40, 41, ..., 49	0: 9, 19, ..., 99 1: 20, 21, ..., 29, 56 2: 90, 91, ..., 99 3: 6, 16, ..., 96 4: 50, 51, ..., 59 5: 0, 1, ..., 9 6: 40, 41, ..., 49
super-topics	0–2: 3495 0–4: 1208 0–5: 1837 1–0: 156 1–1: 2092 1–3: 4837 2–0: 2537 2–5: 1420 2–6: 2418	0–0: 1250 0–1: 2095 0–2: 3485 0–3: 4845 0–4: 1211 0–5: 2225 0–6: 1307 1–0: 1487 1–5: 1024 1–6: 1071

and sub-topics are 3 and 7 respectively. As we can see, our model still identifies all the sub-topics. However, the super-topics are more difficult to discover. The learned super-topic #1 corresponds to the true super-topic #2, although the counts for sub-topics #0 and #6 are decreased. The other two super-topics in the true structure are merged into one big super-topic.

6.2.2.2 20 newsgroups dataset

I also evaluate our model on real-world text data. In this experiment, I choose the 20 newsgroups dataset because it has a pre-defined hierarchical structure. Therefore I can compare our model with both PAM and HDP. I use the same comp5 subset

Table 6.3. Example topics discovered by PAM-HDP from 20 newsgroups dataset comp5 subset

drive	0.0973	mac	0.0673	mb	0.0542	jpeg	0.0715	power	0.0355
disk	0.0502	system	0.0367	simms	0.0497	image	0.0302	cpu	0.0302
drives	0.0388	comp	0.0342	ram	0.0423	gif	0.0283	fan	0.0238
hard	0.0377	disk	0.0263	memory	0.0316	color	0.0190	heat	0.0219
controller	0.0284	sys	0.0242	bit	0.0282	images	0.0167	supply	0.0171
bios	0.0219	ftp	0.0192	vram	0.0247	format	0.0149	motherboard	0.0163
floppy	0.0219	macintosh	0.0178	simm	0.0232	quality	0.0142	sink	0.0152
system	0.0196	apple	0.0171	board	0.0227	file	0.0116	case	0.0141
ide	0.0189	faq	0.0153	meg	0.0220	bit	0.0111	switch	0.0141
scsi	0.0154	software	0.0128	chip	0.0200	version	0.0103	chip	0.0120

as in the document classification experiment. I present example topics discovered by PAM-HDP in Table 6.3.

In the training procedure, only HDP uses the categorization information in the data. PAM and PAM-HDP do not rely on the data structure. PAM has different performance for different numbers of super-topics and sub-topics, while HDP and PAM-HDP automatically learn the numbers of topics. The parameters for PAM-HDP are:

$\alpha_0 = 1.0$, $\gamma_0 = 0.2$,

$\alpha_1 = 2.0$, $\gamma_1 = 2.0$, $\sigma_1 = 5.0$,

$\alpha_2 = 0.01$.

I use the same values for HDP's corresponding parameters. The performance is evaluated based on likelihood of held-out test data and the result is presented in Figure 6.4.

The left 6 bars show the results of PAM with different numbers of super-topics and sub-topics. The 7th bar is HDP, where 98 sub-topics are discovered from data. When I use 5 super-topics, which is the number of categories, PAM does not perform as well as HDP. This is easy to understand because HDP has the "correct" information about the 5 super-topics. However, when I increase the number of super-topics, PAM benefits from the flexibility to capture more topic correlations and outperforms HDP.

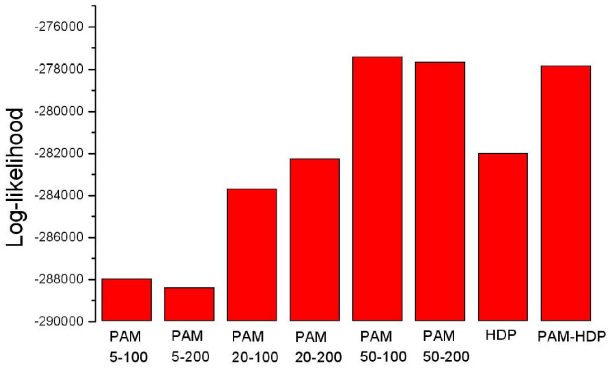


Figure 6.4. Likelihood comparison among PAM, HDP and PAM-HDP

With PAM-HDP, I automatically discover 76 super-topics and 112 sub-topics, and the likelihood is only slightly worse than the best performance of PAM.

6.3 A Heuristic Algorithm to Structure Learning

In this section, I propose another approach to structure learning. It is inspired by a search algorithm proposed by Chow and Liu to estimate dependence trees from data [6]. Compared to the non-parametric Bayes method, this algorithm is more scalable.

First consider a simple DAG structure as shown in Figure 6.5. Given a set of documents and the topics discussed in every document, the probability of this DAG D is defined as:

$$P(D) = P(r)P(t_1|r)P(t_2|r)P(w_1|t_1)P(w_2|t_1,t_2)P(w_3|t_2),$$

where $P(r)$ equals 1, $P(t_i|r)$ is the number of documents that topic t_i occurs divided by the total number of documents, $P(w_i|t_j)$ is the number of documents that w_i and t_j both occur divided by the number of documents that contain t_j, and $P(w_i|t_j,t_k)$ is the number of documents that w_i, t_j, t_k co-occur divided by the number of documents that discusses both topic t_i and t_j.

Given this definition, I need to address several issues to automatically discover a DAG D^* that has the maximum probability among all possible structures. For example, the number of topics is unknown, and the probabilities involving topics

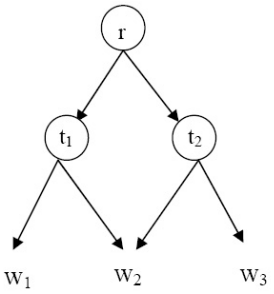


Figure 6.5. A simple DAG structure

cannot be directly evaluated without knowing the topics discussed in every document. I will first discuss our solution to a simplified problem where the structure space only consists of trees, and then it will be extended to arbitrary DAG structures.

The simplified problem can be related to a more general task, i.e., to approximate an n-dimensional discrete probability distribution by the product of $n-1$ second-order distributions. Chow and Liu have investigated this problem and presented a solution that discovers the maximum-likelihood estimate of the distribution. Let $P(X)$ denote the joint probability of n variables $x_1, x_2, ..., x_n$. An approximation that only uses second-order distributions can be written as

$P_T(X) = \prod_{i=1}^{n} P(x_{mi}|x_{mj})$,

where $(m_1, m_2, ..., m_n)$ is a permutation of $1, 2, ..., n$ and each variable x_i can be conditioned on only one of the other variables. It is easy to see that every approximation $P_T(X)$ corresponds to one tree structure T, where each node is associated with one variable, and there is an edge connecting two variables x_i and x_j if they are in the same term of the above expansion. This tree T is called the dependence tree for the approximation $P_T(X)$. Each edge in a dependence tree is assigned with an edge weight, which is the mutual information between the two nodes connected by the edge:

$I(x_i, x_j) = \sum_{x_i, x_j} P(x_i, x_j) \log \frac{P(x_i, x_j)}{P(x_i)P(x_j)}$

Furthermore, the weight of a dependence tree is defined as the sum of all edge weights. It has been shown that the problem of finding the optimal approximation $P_{T^*}(X)$ is the same as the problem of finding a tree T^* that has the maximum weight. A simple search algorithm has been proposed to solve this problem. It starts with no edges at all and iteratively connects two variables if they have the highest mutual information compared to other pairs and adding this edge does not cause any cycle. They have shown that this algorithm discovers a tree structure that corresponds to the maximum-likelihood estimate of the distribution.

It is easy to see that finding a tree with the maximum probability for PAM is a special case of this approximation problem. The variables include both topics and words, and there is one additional restriction that a word variable can only be conditioned on a topic variable. However, the solution proposed by Chow and Liu cannot be directly applied to our problem because the number of topics is unknown, and more importantly, the topics discussed in every document are unknown. Therefore I propose an alternative approach based on a similar heuristic. Instead of maximizing the overall mutual information from every edge of the tree, which cannot be calculated from the data, I try to maximize the mutual information between words that are assigned to the same topic. The outline of the search algorithm is shown below:

1. Let C be the set of words.
2. Let E be empty.
3. Repeat until C is empty
    - Choose the pair $< x_i, x_j >$ from C that has the highest mutual information.
    - Remove x_i and x_j from C, and add their average x_{ij} to C
    - Add the edge $< x_i, x_j >$ to E
4. Output E

Note that this is a greedy agglomerative algorithm that does not guarantee to find the optimal solution. Additional operations such as removing edges or topic nodes might be introduced to improve the performance. To extend this algorithm to discover DAG structures instead of trees, I need to allow the nodes that are already merged to be connected with other nodes in the future.

6.4 Applications

In this section, I describe a variety of applications that I propose to perform in the future.

1. **Information retrieval.** Topic models such as LDA have been used in information retrieval to evaluate the probability of a query given a document, which is then used to generate a ranked list of relevant documents. The ability of supporting a large number of fine-grained topics in PAM can provide a better solution to IR systems with large collections of documents.

2. **Hierarchical topic discovery in research papers.** This application can be used to evaluate the extended HLDA structure, where the goal is to discover topics with different levels of granularity. I will use the Rexa dataset for this experiment.

3. **Social network analysis.** Similar to words and topics, people are also correlated. Social network analysis is an increasingly interesting area where variants of LDA have been successfully used to discover correlation patterns. Here I propose extensions to PAM that incorporate information about people. The goal is to extract people specific topics, which may be used to analyze their social roles and relations.

4. **Semi-supervised learning.** Topic models can be viewed as an example of soft clustering, where each word can belong to multiple clusters. The clustering

result from unlabeled data can be used in many supervised tasks such as document classification. PAM generates a hierarchical structure among clusters, which has been shown to be especially useful in semi-supervised learning.

CHAPTER 7

SUMMARY

In this proposal, I have presented pachinko allocation, a mixture model that uses a DAG structure to capture arbitrary topic correlations. Each leaf in the DAG is associated with a word in the vocabulary, and each interior node corresponds to a topic that models the correlation among its children, where topics can be not only parents of words, but also other topics. The DAG structure is completely general, and some topic models like LDA can be represented as special cases of PAM. Compared to other approaches that capture topic correlations such as hierarchical LDA and correlated topic model, PAM provides more expressive power to support complicated topic structures and adopts more realistic assumptions for generating documents.

I have also presented an approach combining the pachinko allocation model and topics over time that jointly captures topic correlations and identifies their localization in time. I have applied this model to a large corpus of research papers and discovered interesting patterns in the evolution of topics and the connections among them. Unlike some related work with similar motivations, PAMTOT does not require discretization in time or Markov assumptions on state dynamics. The relative simplicity provides advantages for future extensions to model more complex structures among not only topics, but other related information as well.

I propose the following work that needs to be completed:

1. **A non-parametric Bayes approach to structure learning.** More evaluation for the proposed model. This work is planned to be finished in August 2006.

2. **An extension to HLDA.** Derive the inference algorithm and parameter estimation method under this structure. Apply it to the Rexa dataset to discover a topic hierarchy. This work is planned to be finished in November 2006.

3. **Application to information retrieval.** Make the training procedure more scalable in order to handle the large amount of data in IR systems. This work will be finished in January 2007, with a SIGIR submission.

4. **The heuristic algorithm for structure learning.** Improve the search procedure and justify it theoretically. This work is planned to be finished in late February or early March of 2007.

5. **Other applications.** Apply PAM to social network analysis and semi-supervised learning. This work will finish in early April 2007.

BIBLIOGRAPHY

[1] Blei, D., Griffiths, T., Jordan, M., and Tenenbaum, J. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems 16*. 2004.

[2] Blei, D., and Lafferty, J. Correlated topic models. In *Advances in Neural Information Processing Systems 18*. 2006.

[3] Blei, David, and Lafferty, John. Dynamic topic models. In *International Conference on Machine Learning (ICML)* (2006).

[4] Blei, David, Ng, Andrew, and Jordan, Micheal. Latent dirichlet allocation. *Journal of Machine Learning Research 3* (2003), 993–1022.

[5] Chib, S. Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association* (1995).

[6] Chow, C., and Liu, C. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory* (1968), 462–467.

[7] Diggle, P., and Gratton, R. Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society* (1984).

[8] Griffiths, T., and Steyvers, M. Finding scientific topics. *Proceedings of the National Academy of Sciences 101(suppl. 1)* (2004), 5228–5235.

[9] Griffiths, T., and Steyvers, M. Finding scientific topics. In *Proceedings of the National Academy of Sciences* (2004), pp. 5228–5235.

[10] Kleinberg, J. Bursty and hierarchical structure in streams. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2002).

[11] Madsen, Rasmus E., Kauchak, David, and Elkan, Charles. Modeling word burstiness using the dirichlet distribution. In *International Conference on Machine Learning (ICML)* (2006).

[12] Newton, M., and Raftery, A. Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society* (1994).

[13] Nodelman, U., Shelton, C.R., and Koller, D. Continuous time bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI)* (2002), pp. 378–387.

[14] Song, X., Lin, C.-Y., Tseng, B. L., and Sun, M.-T. Modeling and predicting personal information dissemination behavior. In *The Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2005).

[15] Teh, Y., Jordan, M., Beal, M., and Blei, D. Hierarchical Dirichlet processes. *Journal of the American Statistical Association* (2005).

[16] Wang, Xuerui, and McCallum, Andrew. Topics over time: A non-markov continuous-time model of topical trends. In *Submitted to the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2006).

[17] Wang, Xuerui, Mohanty, Natasha, and McCallum, Andrew. Group and topic discovery from relations and text. In *SIGKDD Workshop on Link Discovery: Issues, Approaches and Applications (LinkKDD-05)* (2005), pp. 28–35.