PROBABILISTIC LATENT VARIABLE MODELS FOR KNOWLEDGE DISCOVERY AND OPTIMIZATION

BY

XIAOLONG WANG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Doctoral Committee:

Professor Chengxiang Zhai, Chair
Professor Jiawei Han
Professor David Forsyth
Professor Angelia Nedich, Arizona State University
Professor Joy Ying Zhang, Carnegie Mellon University

Abstract

I conduct a systematic study of probabilistic latent variable models (PLVMs) with applications to knowledge discovery and optimization. Probabilistic modeling is a principled means to gain insight of data. By assuming that the observed data are generated from a distribution, we can estimate its density, or the statistics of our interest, by either Maximum Likelihood Estimation or Bayesian inference, depending on whether there is a prior distribution for the parameters of the assumed data distribution.

One of the primary goals of various machine learning/data mining models is to reveal the underlying knowledge of observed data. A common practice is to introduce latent variables, which are modeled together with the observations. Such latent variables compute, for example, the class assignments (labels), the cluster membership, as well as other unobserved measurements of the data. Besides, proper exploitation of latent variables facilities the optimization itself, which leads to computationally efficient inference algorithms.

In this thesis, I describe a range of applications where latent variables can be leveraged for knowledge discovery and efficient optimization. Works in this thesis demonstrate that PLVMs are a powerful tool for modeling incomplete observations. Through incorporating latent variables and assuming that the observations such as citations, pairwise preferences as well as text are generated following tractable distributions parametrized by the latent variables, PLVMs are flexible and effective to discover knowledge in data mining problems, where the knowledge is mathematically modelled as continuous or discrete values, distributions or uncertainty. In addition, I also explore PLVMs for deriving efficient algorithms. It has been shown that latent variables can be employed as a means for model reduction and facilitates the computation/sampling of intractable distributions.

Our results lead to algorithms which take advantage of latent variables in probabilistic models. We conduct experiments against state-of-the-art models and empirical evaluation shows that our proposed approaches improve both learning performance and computational efficiency.

To My Family.

Acknowledgments

This project would not have been possible without the support of many people. Many thanks to my adviser, Chengxiang Zhai, who read my numerous revisions and helped make some sense of the confusion. Also thanks to my committee members, Jiawei Han, David Forsyth, and Joy Zhang, who offered guidance and support. And finally, thanks to parents, and numerous friends who endured this long process with me, always offering support and love.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The general treatment of data mining and machine learning problems can be categorized into two classes: probabilistic methods and non-probabilistic methods. For classification applications, for example, probabilistic methods include logistic regression, maximum entropy, and conditional random fields, for binary, multi-class, and sequential predictions, respectively. The non-probabilistic counterpart includes the well known support vector machines (or the more general max-margin methods), which is also investigated for binary, multi-class and structure predictions. In clustering problems, one of the most widely used probabilistic methods is the family of mixture models while matrix factorizations are usually adopted in non-probabilistic settings. The focus of this thesis is on the probabilistic methods, which have several important advantages: (1) Probabilistic models assign probabilities instead of real-value scores to outcomes (cluster id, class label), which convey statistical uncertainty. Also, the measurement of probability is intuitive and statistically meaningful. (2) In contrast to the optimization within the non-probabilistic framework, where expert knowledge is required to determine the form objective function, probabilistic methods naturally yield a principled and generic optimization paradigm: Maximum likelihood estimation (MLE), or equivalently, Kullback-Leibler (KL) divergence minimization. (3) In Bayesian settings, model regularization can be further achieved by specifying a prior distribution of the model parameters. The optimization problem is then solved by either Maximum A Posterior (MAP) or posterior expectation, which extends MLE. These advantages are appealing both theoretically and practically, which motivates the studies in this thesis.

Probabilistic latent variable models (PLVMs) have provided a mathematical-based approach to the statistical modeling of a wide variety of random phenomena which cannot be explained well by simple distributions, such as binomial, multinomial, Poisson for discrete distributions, and Gaussian, Dirichlet for continuous distributions, respectively. PLVMs assume that the observed data are accompanied by a group of "unobserved" latent variables. And the distribution of the observed data is conditioned on the latent variables. PLVMs are able to model complex distributions through an appropriate choice of the latent variables to represent accurately the local areas of support of the true distribution. Computation can therefore be made feasible through incorporating the latent variables, as the latent variables are usually chosen with a tractable form.

An illustrating example, topic modeling, demonstrates how latent variables can be used to model "topics." A topic

is mathematically represented by a multinomial distribution over words in a vocabulary. The unigram distribution of a document is then regarded as a "mixture" of the topics. Though the observation is merely words in the documents, by introducing latent variables, namely the topic assignments of words, the semantic relationship of words can be identified to a great extent, and the prominent subject of a document can be revealed as well. For instance, in topic modeling such as Probabilistic Latent Semantic Indexing (PLSI) and Latent Dirichlet Allocation (LDA), words like "science" and "technology" would both have a large probability in a particular topic of scientific research, while "baseball" and "basketball" would both have a large probability in another topic of sports. In computer vision, topic modeling is also applied to the task of image segmentation where pixels of an image are seen as a mixture of latent objects.

We devote the rest of this section to illustrate how we can leverage probabilistic latent variable models for knowledge discovery and optimization.

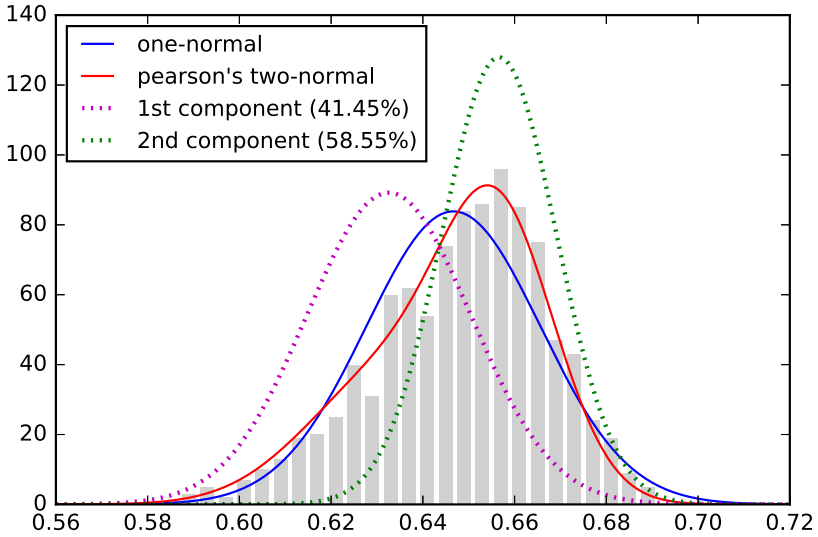
1.1 Latent Variable for Knowledge Discovery

PLVMs as an extremely flexible method of modeling have been extensively studied for knowledge discovery. In recent decades, from probabilistic latent semantic indexing, latent Dirichlet allocation, to Dirichlet process, Indian buffet process, literatures have witnessed numerous PLVMs being proposed and widely applied to varying fields such as natural language processing, speech recognition, and computer vision. In this section, we restrict our analysis to mixture models, also better known as topic modeling in recent literature.

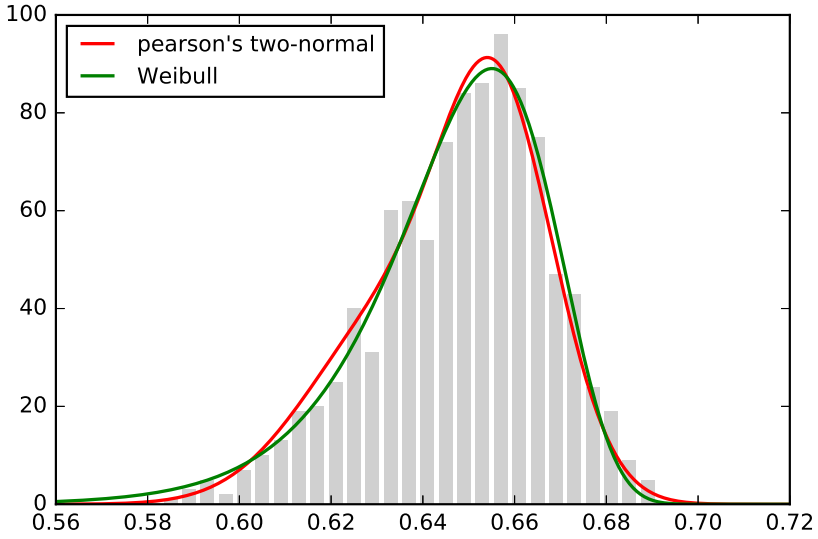
1.1.1 Mixture Models — A Historical Account

The early research efforts on mixture models can be dated back to 1896 when Karl Pearson fitted a mixture of two normal probability density functions (Pearson, 1896) on the problem of *Breadth of "Forehead" of Crabs*. As a pioneering biostatistician, he has been credited for the finite mixture models and method of moments among his other contributions. In hindsight, his work also established the computational (optimization) theory of statistical modeling, a difficult yet interesting research area even today, which inspires my study on this topic composing most of this thesis.

The dataset on which Pearson modeled consisted of measurement on the ratio of forehead width to the body length of 1000 crabs sampled at the Bay of Naples by zoologist W.F.R. Weldon. Weldon analyzed the histogram of the observations, which is plotted in Figure 1.1a, along with a normal distribution fitted using Maximum Likelihood (see the solid blue line). However, Weldon (1893) speculated that the asymmetry in the histogram, "a well-marked deviation from this normal shape," could be resulted from a hypothesis that "the units grouped together in the measured material are not really homogeneous." To validate whether the population of crabs was evolving toward two subspecies, he turned to his colleague Pearson for help on mathematics.



(a) In this plot, the bar chart of the observations from Weldon is shown in grey. The blue solid line shows the single normal distribution fitting the data using Maximum Likelihood; And the solid line in red plots the mixture model of two normals distributions derived by Pearson using moment matching where its two components are also displayed in green and purple dotted lines.



(b) Comparison between the Pearson's mixture of two normals and a single Weibull distribution. Pearson's mixture model provides a tighter fitting at the mode of empirical distribution. Note that the density function of Weibull distribution is much more complicated than that of normal distribution and it requires numeric means to estimate the parameters.

Figure 1.1: Pearson's Mixture of Two Normals on "Breadth of Forehead of Crabs"

Pearson used two normal distributions to fit the observations. He assumed that the observed data are sampled from $\pi_1\mathcal{N}(\mu_1, \sigma_1^2) + \pi_2\mathcal{N}(\mu_2, \sigma_2^2),\ (\pi_1 + \pi_2 = 1)$. To estimate the parameters, namely, the means (μ_1, μ_2) and

standard-variance (σ_1, σ_2) of the two normal distributions as well as the proportions (π_1, π_2) of the two components, Pearson followed the method of moments (which was also introduced by himself in 1894). Though moment matching is superseded by Fisher's method of maximum likelihood (Pfanzagl, 1994) in nowadays classic statistical modelling, it was a relatively numerically simpler approach in most cases. However, the calculation was still formidable and daunting at the time without the aid of computer or other machinery of any kind. Mathematically, the problem involves five parameters $\mu_1, \mu_2, \sigma_1, \sigma_2$ and π_1 (since we can obtain $\pi_2 = 1 - \pi_1$) and to find a solution, the parameters need to ensure that the mixture model matches on the first five moments. Pearson derived a ninth degree polynomial (nonic) and two candidate real roots are found. He finally chose the solution on the basis of agreement with the sixth moment. In Figure 1.1a, the dashed curve in red shows Pearson's mixture and its two components are displayed in purple and green dotted lines. Clearly, the mixture is skewed and better fits the histogram than a single normal distribution. And indeed, two subspecies are identified which verifies the hypothesis of Weldon.

It is quite an advanced idea to leverage latent variables for statistical modeling at that time. Otherwise properly fitting the asymmetric observations would involve a much more complicated distribution. In fact, we can also explain the data with a skewed Weibull distribution, the parameter of which are nevertheless computationally difficult to estimate (The Maximum Likelihood estimator for the shape parameter is the solution to the equation $\frac{1}{k} = \frac{\sum_{i=1}^{N}(x_i^k \log x_i - x_N^k \log x_N)}{\sum_{i=1}^{N}(x_i^k - x_N^k)} - \frac{1}{N}\sum_{i=1}^{N} \log x_i$, and numeric methods, which were very primitive at the time of late 19th century, is required). Therefore Weibull distribution was not a practical option for Pearson to fit the data when the aid of computers was not available. In Figure 1.1b, we compare Peason's mixture of two normals with one single Weibull distribution fitting the data using Maximum Likelihood. The difference between the two curves is not significant. However, Pearson's result seems to fit better at the mode around 0.66.

1.1.2 Mixture Models — Development of the EM Algorithm

Although solving the mixture model with the method of moments is a very laborious task and performing the necessary calculation is even more heroic (McLachlan and Peel, 2004), it does not always yield the optimal solution in the statistical sense. The maximum likelihood approach, however, possesses superior statistical property as it tries to place higher probability close to the observed data and are more often unbiased. With the development of optimization in the modern computer science, statistical modeling is able to utilize numerical algorithms to solve Maximum Likelihood Estimation (MLE). Among the different optimization methods, the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) has greatly stimulated interest in the use of mixture models as well as other PLVMs. Several reasons can be accounted for the popularisation of the EM algorithm: (1) It is generally easy to implement the algorithm and it has virtually no parameters to tune, as compared to, for example, gradient descent, where a carefully selected learning step is required to ensure fast training; (2) It usually does not need any special treatment to handle the constraints of the

model. For example, in the normal mixture problem, the standard-variance of a component normal is always positive. In the EM algorithm, this is naturally satisfied since it is computed as the empirical standard-variance of the complete data generated out of the posterior distribution; (3) EM is a flexible family of approaches where the variational distribution in the expectation step can be simplified (or constrained) for the purpose of computation efficiency (e.g. mean-field EM and convex relaxations, (see Wainwright and Jordan, 2008, Chapter 5, 7)) and the maximization step can also be substituted by an ascend step. We leave the details of EM algorithm in Section 2.2. In this section, we provide a brief comparison between EM algorithm and Pearson's method of moments and show how Pearson's result can be improved by the EM algorithm.

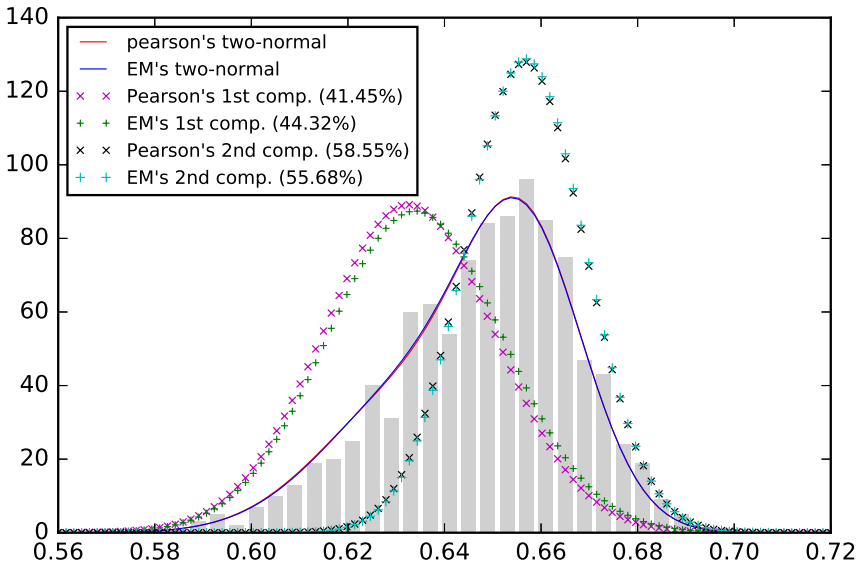


Figure 1.2: Comparison of the mixture model of two normals between Pearson's approach and EM algorithm. The two mixture models are very close to each other showing that the moment-matching method of Pearson obtains a near optimal likelihood.

We plot the curves of the mixture models of the two methods as well as their components in Figure 1.2. The results are almost identical. To assess the quality of the model quantitatively, Pearson used the Chi-square test (Pearson, 1900) which he proposed to examine if the observed data is indeed from the model. We follow his practice and report the result in Table 1.1.

As expected, we see that the EM algorithm results in the smallest Pearson's Chi-square. In less mathematical terms, the observed data is distributed more close to the model given by the EM algorithm. In addition, the p-values in the significant test show that it is more certain that the data is sampled from the mixture normal of EM algorithm. To an extent, the assessment on the Weldon's crab dataset justifies the use of EM algorithm to solve MLE in applications of

Table 1.1: Pearson's Chi-square test and p-Value for a single normal model, a single Weibull model, and the two normal mixture model of Pearson and EM algorithm in the "Breadth of Forehead of Crabs" problem. For the normal models, we also include the model parameters.

Method	μ_1	μ_2	σ_1	σ_2	π_1	π_2	freedom	Chi-square	p value
Single Normal	0.6466	—	0.0190	—	1	—	2	71.6836	2.157×10^{-6}
Single Weibull	—	—	—	—	—	—	2	28.3841	0.2904
Pearson	0.6326	0.6566	0.0179	0.0125	0.4145	0.5855	5	21.0342	0.5186
EM	0.6339	0.6568	0.0182	0.0124	0.4432	0.5568	5	20.8438	0.5304

mixture modeling.

1.1.3 From Mixture Models to Topic Modeling

Since late 1990s, the study on document understanding has witnessed a new approach of PLVMs which is often referred to as topic modeling. The first well recognized topic modeling method, probabilistic latent semantic indexing (PLSI) (Hofmann, 1999), is simple yet effective. Essentially it sees the unigram word (w_d) distribution of a document d as a K-mixture of multinomial distributions $\beta_1, \ldots, \beta_K$ with proportions $\theta_{d,1}, \ldots, \theta_{d,k}$. Those β_K are referred to as "topics" because the words of large probabilities in a component are often semantically related. In addition, the topic weights θ_d of a document provides a succinct summary of the documents. Computationally, θ_d has a much lower dimensionality than w_d and thus can be leveraged as a (part of) feature vector in tasks such as document classification or clustering. Moreover, θ_d is semantically meaningful as the similarity of θ_d's correlates with the similarity of the subject of documents, which can be greatly useful in document understanding, information indexing, etc.

In terms of modeling the latent variables, there are two milestone progresses: the Bayesian inference and nonparametric statistics. The early efforts promoting the Bayesian nonparametrics and advocating the theoretical formalization of topic modeling, specifically, the analysis on random processes of exchangeable partitions (Pitman, 1995), are the lectures taught by Pitman et al. at Berkeley in Spring 2002. Many results obtained in this direction (Blei and Lafferty, 2009; Blei et al., 2003, 2010) are immediate fruit of the course and readers interested in a principle introduction on this topic should refer to the lecture notes (Pitman et al., 2002) and the references therein.

Bayesian inference departs from the traditional MLE framework. It assumes a prior distribution on latent variables parametrized by the *hyperparameters*. The advantages of introducing a prior on latent variables are mainly two folds and we show them using the Latent Dirichlet Allocation (LDA) (Blei et al., 2003) as an example: (1) It enables user to incorporate human knowledge about the latent variables into modeling. In document understanding, the word distribution of a topic as well as the proportion of topics for a document are naturally *sparse*. LDA encourages such behavior by using a Dirichlet prior with a small hyperparameter $\boldsymbol{\alpha}$. (2) By selecting the form of prior distribution carefully, the prior and posterior distributions can be in the *same* family (with different parameters though). Such

conjugate prior-posterior pairs are computationally beneficial in both Gibbs sampling as well as variational inference. LDA chooses Dirichlet as the conjugate prior to the multinomial distribution, and the posterior distribution is also a Dirichlet of parameter $\boldsymbol{\alpha} + \mathbf{n}$, where $\mathbf{n}$ is often referred as the pseudo-count of the latent variables in each topic. Estimation method for Bayesian inference has also been greatly developed beyond MLE. There are two major estimation methods of the latent variables in Bayesian setting which are Bayesian Estimator (Posterior Expectation) and Maximum a Posterior (MAP). The first computes the posterior expectation of the latent variables given the observed data while the second selects the value with the maximal probability in the posterior distribution, which can be viewed as an extension of the MLE method. In the context of topic modeling, it has been noticed that Bayesian estimator is more popular than MAP. The major criticism of MAP is the fact that it is still a point estimation in nature. Specifically in topic modeling, it is not uncommon that the posterior distribution of the latent variables are in fact multi-modal. And therefore it is computationally infeasible (or even intractable) to calculate MAP due to the non-convex nature of the problem.

Nonparametric statistics aims to model the data with possibly infinite number of latent variables. In topic modeling, it implies that one can model a infinite number of topics or words in the vocabulary. Although in practice it does not seem to be immediately useful since there is always a finite upper-bound for these quantities, it is critical to rely on expert knowledge to appropriately select the values. Nonparametric statistics are most powerful to adaptively learn the number of latent variables that are adequately large to explain the data by using random processes. Random processes are extensively studied in recent literature, as surveyed in (Hajek, 2015), including Gaussian process (Rasmussen and Williams, 2006), Dirichlet process (Teh, 2011), Indian buffet process (Ghahramani and Griffiths, 2005), and hierarchical processes (Blei et al., 2010; Griffiths and Tenenbaum, 2004; Teh et al., 2012), just to name a few. Mathematically, to model the latent variables from possibly infinite number of choices, the nonparametric approach assumes a random process as prior. Computationally, there are mainly two strategies, Gibbs sampling and truncated variational inference, to estimate the posterior distribution of the possibly infinite number of latent variables. Gibbs sampling takes advantage of the fact that the prior process usually yields a simple prediction rule of one latent variable given all others. For example, in Dirichlet process, using the notion of Chinese restaurant process (Pitman et al., 2002), the probability of a latent variable choosing an existing value is proportional to the number of other latent variables of the same value, or a new value proportional to the hyperparameter α:

$$\mathrm{P}_{CRP}(z_i = k | z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_N) \propto \begin{cases} \sum_{j=1, j\neq i}^{N} \mathbb{1}(z_j = k) & \text{if } k < K \\ \alpha & \text{if } k = K + 1 \end{cases} \tag{1.1}$$

where it is supposed that the value of $z_j, j \neq i$ is choosing from $1, \ldots, K$ and for any $k < K$ the support is nonempty.

Therefore it is feasible to investigate sampling methods for inference. While alternatively, another strategy for estimation is to approximate the possibly infinite posterior with a finite approximation. For the Dirichlet Process (as well as the generalized Pitman-Yor two-parameter process (Pitman and Yor, 1997)), the truncating approximation is based on a stick-breaking (Ishwaran and James, 2011) interpretation. It views the process as breaking a stick with the proportion as a sample from a Beta distribution and the truncation stops the breaking after there is a predefined number of sticks generated. Both of the above two strategies have advantages: Gibbs sampling does not need to truncate the size of latent variables by a finite number, while the truncated variational inference is generally computational efficient. However, as shown in (Wang and Blei, 2012), it is possible to combine the two ideas together by performing the E-step in variational EM via sampling.

1.2 Latent Variables for Optimization

Previous research such as topic modeling mainly incorporates the latent variables for the purpose of knowledge discovery. Another motivation to use latent variable models is efficient computation. In previous discussion of the "Breadth of Forehead of Crabs" example, we have already seen that by introducing latent variables, the mixture model is much easier to compute than that of the Weibull distribution. However, contemporary efforts in the direction of leveraging PLVMs for efficient computation was less explored. In one of our recent work, Dual-Clustering Maximum Entropy (DCME) (Wang et al., 2016b), it is demonstrated that PLVM is an effective means to improve the optimization efficiency.

We explore PLVM in the context of Maximum Entropy (ME) models. ME is a classic approach in classification as well as word embedding. However, it becomes computationally challenging when the number of classes or the vocabulary size is large. DCME approaches the problem by optimizing ME in its primal-dual form. The key insight is to introduce a latent cluster assignment for each training instance and assume that the dual variables of an instance are determined by the corresponding latent assignment. As an initial investigation, we use the latent variables in a much simpler manner than the mixture models. Specifically, we restrict the latent variable to distribute as a Kronecker delta which has support only on a single value, in contrast to the case of mixture models where the latent variable is subject to a more general multinomial distribution. DCME naturally leads to an approximation of the dual variables which can be computed by a K-means like clustering. More importantly, it enables an efficient online-offline computation scheme whose computational complexity does not depends on the number of classes nor the vocabulary size. Empirical studies demonstrated that DCME significantly outperforms state-of-the-art approaches.

1.3 Contribution of this Thesis

In this thesis, I describe a range of applications where latent variables can be leveraged for knowledge discovery and efficient optimization. Works in this thesis demonstrate that PLVMs are a powerful tool for modelling incomplete observations. Through incorporating latent variables and assuming that the observations such as literature citations, pairwise preferences in crowdsourcing as well as unstructured text are generated following tractable distributions parametrized by the latent variables, PLVMs are flexible and effective to discover knowledge in data mining problems, where the knowledge is mathematically modelled as continuous or discrete values, distributions or uncertainty. For example, when modelling literature citations, latent variables can be inferred to identify research topics and evolution of research themes; While only observing pairwise preferences labelled by non-expert workers in crowdsourcing, PLVM as a generative process is capable to recover the ground truth ranked lists; And finally, by fitting the unstructured text with underlying phrasal structures, it can be shown that both the phrasal allocation and phrase embeddings are effectively computed. In addition, I also explore the PLVMs for deriving efficient algorithms. It has been shown that latent variables can be employed as a means for model reduction or to facilitating computation/sampling of intractable distributions. For instance, PLVM has been shown to improve efficiency of Maximum Entropy which does not scale well as the number of classes by performing model reduction with the latent variables; In addition, in cases where the computation involves a intractable distribution, latent variables are also investigated to facilitate the calculation via Gibbs sampling.

1.4 Overview of this Thesis

In Chapter 2, we briefly discuss a few key mathematical ingredients that can greatly facilitate the understanding of PLVMs. Next, we move on to show two scenarios where PLVMs are applied for knowledge discovery in Chapter 3 and Chapter 4. Leveraging PLVMs for efficient optimization is presented in Chapter 5. The last work we propose in this thesis takes the advantages of PLVMs in both aspects, namely extracting the phrasal structure with an efficient optimization scheme and effectively learning the semantic embeddings of phrases, is discussed in Chapter 6.

The first work analyzes the citations of literatures (Wang et al., 2013). Understanding how research themes evolve over time in a research community is useful in many ways (e.g., revealing important milestones and discovering emerging major research trends). In this study, we propose a novel way of analyzing literature citation to explore the research topics and the theme evolution by modeling article citation relations with a probabilistic generative model. The key idea is to represent a research paper by a "bag of citations" and model such a "citation document" with a probabilistic topic model. We explore the extension of a particular topic model, i.e., Latent Dirichlet Allocation (LDA), for citation analysis, and show that such a Citation-LDA can facilitate discovering of individual research topics as

well as the theme evolution from multiple related topics, both of which in turn lead to the construction of evolution graphs for characterizing research themes. We test the proposed citation-LDA on two datasets: the ACL Anthology Network (AAN) of natural language research literatures and PubMed Central (PMC) archive of biomedical and life sciences literatures, and demonstrate that Citation-LDA can effectively discover the evolution of research themes, with better formed topics than (conventional) Content-LDA.

The second work explores PLVMs in a crowdsourcing setting (Wang et al., 2016a). Crowdsourcing services make it possible to collect huge amount of annotations from less trained crowd workers in an inexpensive and efficient manner. However, unlike making binary or pairwise judgements, labeling complex structures such as ranked lists by crowd workers is subject to large variance and low efficiency, mainly due to the huge labeling space and the annotators' non-expert nature. Yet ranked lists offer the most informative knowledge for training and testing in various data mining and information retrieval tasks such as *learning to rank*. In this paper, we propose a novel generative model called "Thurstonian Pairwise Preference" (TPP) to infer the true ranked list out of a collection of crowdsourced pairwise annotations. The key challenges that TPP addresses are to resolve the inevitable incompleteness and inconsistency of judgements, as well as to model variable query difficulty and different labeling quality resulting from workers' domain expertise and truthfulness. Experimental results on both synthetic and real-world datasets demonstrate that TPP can effectively bind pairwise preferences of the crowd into rankings and substantially outperforms previously published methods.

Another aspect of PLVMs is to improve the efficiency of optimization. To this end, we devote another chapter to discuss the study of Dual-Clustering Maximum Entropy (Wang et al., 2016b). Maximum Entropy (ME), as a general-purpose machine learning model, has been successfully applied to various fields such as text mining and natural language processing. It has been used as a classification technique and recently also applied to learn word embedding. ME establishes a distribution of the exponential form over items (classes/words). When training such a model, learning efficiency is guaranteed by *globally* updating the entire set of model parameters associated with *all* items at *each* training instance. This creates a significant computational challenge when the number of items is large. To achieve learning efficiency with affordable computational cost, we propose an approach named Dual-Clustering Maximum Entropy (DCME). Exploiting the primal-dual form of ME, it conducts clustering in the dual space and approximates each dual distribution by the corresponding cluster center. This naturally enables a hybrid online-offline optimization algorithm whose time complexity per instance only scales as the product of the feature/word vector dimensionality and the cluster number. Experimental studies on text classification and word embedding learning demonstrate that DCME effectively strikes a balance between training speed and model quality, substantially outperforming state-of-the-art methods.

The last work presented in this thesis investigates PLVMs for learning phrasal allocation. Existing word embedding

methods are intrinsically hindered by its unigram (bag-of-words) assumption of language. Although efforts towards resolving the semantics for higher level of language units (e.g.phrase, sentence) have been made, most of them either rely on an external resource or employ a complicated decoding algorithm for identifying the composition structure. In this work, we propose an effective yet simple generic algorithm, Phrasal Latent Allocation with Negative Sampling (PLANS), to compute the phrase embedding. We propose transient Chinese Restaurant Process (tCRP) as a prior for words to allocate the phrases within which they are enclosed. In addition, similar to Skipgram, PLANS estimates the embedding for words/phrases with negative sampling. Nevertheless the major challenge in learning is that a reasonable size of the phrases need to be carefully retained and less confident ones are constantly pruned during training. PLANS address this with an online block algorithm which refreshes the set of phrases based on their "frequencies" in the corpus periodically. In addition, simulated annealing (SA) is applied in the sampling process to stabilize the learned phrase set.

Chapter 2

Background

For self-containedness, we provide a short reference to the mathematical tools that we have been frequently used in PLVMs. Readers familiar with the theory of conjugate duality and EM algorithm can skip the content of this chapter. And for a comprehensive account, please refer to the book (Hiriart-Urruty and Lemarechal, 1993).

2.1 Conjugate Duality

The conjugate in optimization context refers to the transformation of a problem to another accompanying problem. The transformation is also known as the *conjugacy* operation or the *Legendre-Fenchel* transformation. It plays an important role in the Lagrangian duality as well as the general convex optimization. To start our discussion, we formally define the conjugate of a function as:

Definition 2.1.1. The conjugate of a convex function [1] f is the function f^* defined by

$$f^*(s) = \sup\{\langle s, x\rangle - f(x)\}, \quad \forall x \in \operatorname{dom} f \tag{2.1}$$

An geometrical interpretation of the conjugate of a *subdifferentiable* function is illustrated in Figure 2.1. A immediate result is that:

Theorem 2.1.1. *For any* $x^* \in \arg\max\{\langle s^*, x\rangle - f^*(s^*)\}$*, we have that* $x^* \in \partial f^{-1}(s^*)$

In addition, the conjugacy transformation is generally symmetric: $f^{**} = f$ for convex functions. To be exact, the identity between the bi-conjugate f^{**} and f is equivalent to the requirement that the convex f is lower semi-continuity (l.s.c): $\liminf_{x \to x_0} \geq f(x_0)$, a sufficient condition of which is that f is subdifferentiable.

Log-Partition and Negative Entropy

One important instance of the conjugate in PLVMs is between log-partition and negative entropy, which are defined as:

[1] we make a stronger assumption that f is convex which can relaxed to the existence of a affine function memorizing f on $\operatorname{dom} f$.

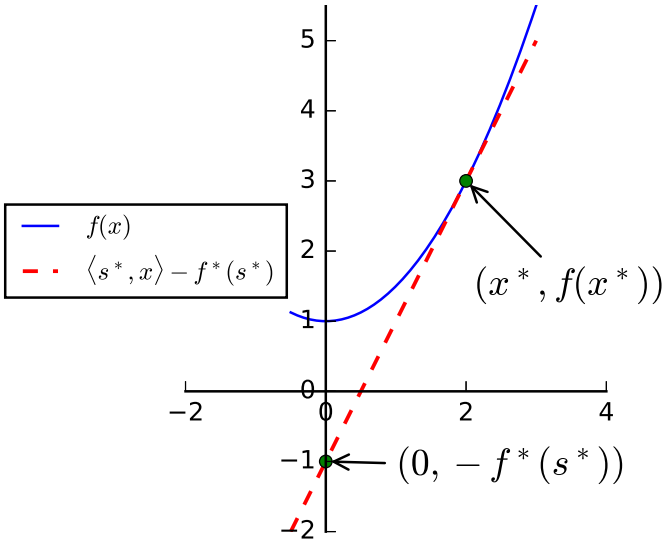


Figure 2.1: A illustration of the relationship between f and its conjugate f^*. For a given s^*, since $f(x) \geq \langle s^*, x\rangle - f^*(s^*)$ always holds, which means that in the plot the curve of $f(x)$ is always above (or on) the line of $\langle s^*, x\rangle - f^*(s^*)$. As a limiting case, $< s^*, x^* > -f^*(s^*)$ is cutting $f(x)$ at $x = x^*$. In addition, the affine function intersects the vertical axis $x = 0$ at the altitude $-f^*(s^*)$. The plot also shows the relationship between s^* and x^* can be described by the *gradient mapping*: $x^* \in \partial f^{-1}(s^*)$, or equivalently $s^* \in \partial f(x^*)$.

$$\text{Log-Parition:} \qquad A(\mathbf{x}) = \log \sum_{i=1}^{N} \exp(x_i) \tag{2.2}$$

$$\text{Negative Entropy:} \quad -H(\mathbf{p}) = \sum_{i=1}^{N} p_i \log p_i \tag{2.3}$$

where $\mathbf{p}$ is an element in the simplex set which is defined as:

$$\Delta_N = \{\mathbf{p} \in \mathbb{R}^N : p_j \geq 0, \sum_{j=1}^{N} p_j = 1\}$$

The log-partition function is often seen in Maximum Entropy models, energy-based models, as well as Markov Random Fields, etc. The straight-forward computation involves a summation over N items, which can be computationally challenging if N is large. For example, in Markov Random Fields, $N = m!$ where m is the number of nodes in the random fields. Computing the log-partition function is often the bottleneck for training such a model.

It is easy to verify that both functions are convex and smooth. Their connection is presented in the theorem below.

Lemma 2.1.1.1. *Assume that*

$$\mathrm{P}(i;\mathbf{s}) = \frac{\exp(s_i)}{\sum_{j=1}^{N}\exp(s_j)}$$

and

$$A(\mathbf{s}) = \log\sum_{j=1}^{N}\exp(s_j)$$

The conjugate duality between the log-partition function and negative entropy states:

$$\begin{aligned} A(\mathbf{s}) &= \max_{\boldsymbol{\mu}\in\Delta_N}\{\sum_{j=1}^{N}\mu_j s_j - \sum_{j=1}^{N}\mu_j\log\mu_j\} \\ &= \max_{\boldsymbol{\mu}\in\Delta_N}\{\mathbf{E}_{\boldsymbol{\mu}}[s_j] + \mathbf{H}(\boldsymbol{\mu})\} \end{aligned} \tag{2.4}$$

where the maximizer is attained at:

$$\mu_j^* = \mathrm{P}(j;\mathbf{s}), \quad 1 \le j \le N \tag{2.5}$$

Proof. In light of Theorem 2.1.1, the general proof of the conjugacy transformation between f and f^* is to verify that $x = \partial f^*\big(\partial f(x)\big)$. And it is easy to show that

$$\mathbf{s} = -\partial H\big(\partial A(\mathbf{s})\big)$$

However, it is much more intuitive to alternatively prove by showing the equivalence in Equation (2.4). We follow the derivation:

$$\begin{aligned} \mathbf{E}_{\boldsymbol{\mu}}[s_j] + \mathbf{H}(\boldsymbol{\mu}) &= -\sum_{j=1}^{N}\mu_j\log\frac{\mu_j}{\mathrm{P}(j;\mathbf{s})} + \log\sum_{j=1}^{N}\exp(s_j) \\ &= -D_{KL}(\boldsymbol{\mu}||\mathrm{P}) + A(\mathbf{s}) \end{aligned}$$

where $D_{KL}(\boldsymbol{\mu}||\mathrm{P})$ is the Kullback-Leibler (KL) divergence.

Note that KL-divergence is always nonnegative:

$$D_{KL}(\boldsymbol{\mu}||P) \ge 0$$

and:

$$D_{KL}(\boldsymbol{\mu}||P) = 0 \quad \Longleftrightarrow \quad \boldsymbol{\mu} = P$$

It follows that:

$$\boldsymbol{\mu}^* = \arg\min_{\boldsymbol{\mu} \in \Delta_N} D_{KL}(\boldsymbol{\mu}||P) = P$$

□

2.2 EM Algorithm: a Modern Reinterpretation

Equipped with the conjugate duality, here we offer a new interpretation of the famous EM algorithm. Part of the idea presented here is also shared by the work (Iusem and Teboulle, 1992).

Suppose that there is a distribution $\mathrm{P}(Z|\Theta)$ where the data $Z = (X, Y)$ is partially observed and can be decomposed into the observations X and the unseen variables Y. Given a set of data $X_1, \ldots, X_N$, MLE solves the problem:

$$\max_{\Theta} \log \mathrm{P}(X_1, \ldots, X_N|\Theta)$$

Using the conjugate duality proved in Theorem 2.1.1:

$$\begin{aligned} \log \mathrm{P}(X_1, \ldots, X_N; \Theta) &= \sum_{i=1}^{N} \log \sum_{Y_i} \mathrm{P}(X_i, Y_i; \Theta) \\ &= \sum_{i=1}^{N} \max_{\boldsymbol{\mu}_i \in \Delta} \Big(\sum_{Y_i} \boldsymbol{\mu}_{i,Y_i} \log \mathrm{P}(X_i, Y_i; \Theta) + \mathbf{H}(\boldsymbol{\mu}_i) \Big) \end{aligned} \tag{2.6}$$

Therefore, the MLE with incomplete observation amounts to:

$$\max_{\substack{\Theta \\ \boldsymbol{\mu}_i \in \Delta, 1 \le i \le N}} \underbrace{\sum_{i=1}^{N} \Big(\sum_{Y_i} \boldsymbol{\mu}_{i,Y_i} \log \mathrm{P}(X_i, Y_i; \Theta) + \mathbf{H}(\boldsymbol{\mu}_i) \Big)}_{\mathcal{F}(\Theta, M)}$$

And for fixed Θ, the optimality condition for μ_i is:

$$\text{E-step:} \quad \frac{\partial \mathcal{F}}{\partial \mu_i} = 0 \quad \Longleftrightarrow \quad \mu_{i,Y_i} = \mathrm{P}(Y_i|X_i; \Theta) \tag{2.7}$$

which is exactly the E-step in EM algorithm.

In addition, to optimize Θ while fixing M:

$$\text{M-step:} \quad \max_{\Theta} \sum_{i=1}^{N} \sum_{Y_i} \boldsymbol{\mu}_{i,Y_i} \log \mathrm{P}(X_i, Y_i; \Theta) \tag{2.8}$$

In the EM algorithm, $\sum_{i=1}^{N} \sum_{Y_i} \boldsymbol{\mu}_{i,Y_i} \log \mathrm{P}(X_i, Y_i; \Theta)$ is referred as *evidence lower bound* (ELBO) function, and the above maximization is identical to the M-step in the EM algorithm.

Using this interpretation, it is also straight-forward to view the EM algorithm as a coordinate-descent algorithm where the objective function is constructed as $\mathcal{F}(\Theta, M)$, which is always a lower bound of the log-likelihood. In below, we briefly discuss two important variants of the EM algorithm.

Variant 1: Relaxation by Approximation

In the above basic version of EM, we assume that μ_i can freely choose any element from the simplex Δ. Nevertheless, it often posits a computational difficulty when solving the posterior distribution $\mathrm{P}(\cdot|X_i; \Theta)$. And it makes sense to trade accuracy of μ_i for computational efficiency, and to compute an approximation of μ_i by a tractable surrogate, which motivates us to study different approximation approaches in the variational inference. In below, we discuss a few well adopted methods.

Mean-Field Approximation: The simplest strategy for approximation is to restrict μ_i to be chosen from a subset, say, $\mathcal{S}$ instead of Δ. Then the optimization problem for μ_i with a constant Θ becomes:

$$\min_{\mu_i \in \mathcal{S}} D_{KL}\big(\mu_i \;||\; P(\cdot|X_i; \Theta)\big) \tag{2.9}$$

When $\mathrm{P}(\cdot|X_i; \Theta) \notin \mathcal{S}$, Equation (2.6) will not hold. In such cases, the solution of Θ will neither converge to that of MLE. Moreover, because of the restricted $\mu_i \in \mathcal{S} \subsetneq \Delta$, the EM algorithm with mean-field approximation is in fact maximizing a (strict) lower bound of the log-likelihood objective.

Approximation by Sampling: As discussed above, it is not uncommon that the posterior $\mathrm{P}(\cdot|X, \Theta)$ does not yield a feasible solution. However instead of compute the density analytically, it is generally possible to use a Gibbs sampler to efficiently sample from the distribution. And when incorporating such sampling-based E-step into the EM framework, it is advantageous to run the Gibb sampler for only a few iterations (before its converging) to collect the statistics for maximization in M-step (Wang et al., 2016a), the idea of which can be justified similarly as that of Contrastive Divergence (Carreira-Perpinan and Hinton, 2005).

General Density Approximation: General methods for approximation of $\mathrm{P}(\cdot|X,\Theta)$ digress from the optimization framework of EM algorithm by substituting the objective of Equation (2.9) with other forms of measurement for closeness. For example, Belief propagation (Yedidia et al., 2005), Bethe approximation (Burgess and Tully, 1978) as well as expectation propagation (Minka, 2001a), when used in EM do not yields a lower bound nor upper bound of the log likelihood. Nevertheless, they are extensively investigated for their empirical improvement in terms of efficiency and performance. Especially, the expectation propagation (EP) method was applied to replace the E-step in the EM framework and it outperforms the mean-field alternatives in cases when evidence is limited (Wang and Blei, 2012). The EP method can be viewed as an approximation to the minimization of the reversed KL-divergence (Minka et al., 2005):

$$\min_{\mu_i \in \mathcal{S}} D_{KL}\big(P(\cdot|X_i;\Theta) \;||\; \mu_i\big) \tag{2.10}$$

Comparing Equation (2.10) to Equation (2.9), we see that the order of two distributions in the KL-divergence is reversed. The in-depth discussion of this topic is beyond the scope of this thesis, and we refer the readers to the brochure on variational inference (Wainwright and Jordan, 2008) and the Ph.D thesis of Minka (2001b) on approximation in Bayesian inference for a complementary review.

Variant 2: Bayesian Variational Inference

EM algorithm is also investigated in Bayesian setting although most techniques remain the same. Specifically, Θ is viewed as a distribution which is governed by hyperparameter Γ, and thus the log-likelihood function involves not only marginalizing the latent variable Y but also the parameter Θ.

In the Bayesian setting, EM is more often called as Bayesian variational inference method. Mathematically, Θ is also a latent variable, no different from Y, and we can still employ the EM algorithm. However, with sufficient observations, the optimization of hyperparmeter Γ is less interested and the M-step is generally skipped. More importantly, by carefully choosing the form of the prior distribution $\mathrm{P}(\Theta;\Gamma)$ (as conjugate prior of $\mathrm{P}(Y;\Theta)$), we have the posterior $\mathrm{P}(\Theta|Y;\Gamma)$ in the same family of distributions as the prior. This is appealing since the update of μ_i in Equation (2.6) can be maximized exactly easily.

2.3 Minimax Theory

In this section we will review some results in the minimax theory which gives the conditions under which the following equality is hold:

$$\max_{z\in Z}\min_{x\in X}\phi(x,z) = \min_{x\in X}\max_{z\in Z}\phi(x,z) \tag{2.11}$$

von Neumann is credited with the first investigation of this problem. There are many different sufficient conditions that guarantees the above equation. Modern analysis employs Farkas Lemma in the *min common/max crossing* framework and an excellent formal discussion can be found in (Bertsekas et al., 2003). In this thesis, we only present an earlier version of minimax theory by Sion (Sion et al., 1958), which is one of several celebrated generalizations of von Neumann's minimax theorem (von Neumann, 1928):

Theorem 2.3.1 (Sion's Minimax Theorem)**.** *Let X and Z both be a compact convex set. Let ϕ be a real-valued function on $X \times Z$ such that:*

1. *$\phi(x, \cdot)$ is upper semi-continuous and quasi-concave on Y for any $x \in X$*
2. *$\phi(\cdot, y)$ is lower semi-continuous and quasi-convex on X for any $y \in Y$*

Then,

$$\max_{z \in Z} \min_{x \in X} \phi(x, z) = \min_{x \in X} \max_{z \in Z} \phi(x, z)$$

An elementary proof of Sion's minimax theorem can be found in (Komiya, 1988). The derivation is simple, short and elegant. Also, the assumption made in theorem 2.3.1 is often satisfied for most practical problems under mild assumptions of ϕ. In general, Equation (2.11) holds when solving problems involving the dual formulation in PLVMs.

Chapter 3

Understanding the Evolution of Research Themes: a Probabilistic Generative Model for Citations

3.1 Introduction

In this chapter, I demonstrate that by modeling literature citations as observations of a generative model with latent variables, research topics as well as evolution themes of research can be identified and described inactively. It exemplifies that PLVM is an effective means for knowledge discovery in data mining problems. Though we use literature citation as the test bed for PLVM, the method presented here can be easily applied to any general network data as well.

How to leverage information technologies to improve the productivity of scientific research is a highly important challenge with clearly huge impact on the society. One bottleneck in research productivity is that as a research community grows, it would be increasingly difficult for researchers to see the complete picture of how a field has been evolving, given the fact that large volume new literatures are written based on previous works. Junior researchers can often get lost in the overwhelming amount of related papers. Researchers who seek to shift to a new topic may spend lots of time preparing a reading list on his/her own. All these clearly hinder the progress of scientific research, and it would be highly beneficial to develop mining techniques to help researchers more easily and more efficiently understand research themes in scientific literature. In general, two aspects of analysis are needed for understanding research themes: First, we need to analyze *each research topic* to answer the following questions: Which papers are the milestone papers that best represent a topic and how to quantify their impact? When did the topic become popular and is it still attracting attention today? Can the topic be summarized accurately with a few keywords? Furthermore, when *investigating topics collectively*, which are the most dominant topics extensively studied? During the evolution, what are the newly generated topics initiated by the old one? Can we identify the underlying evolution patterns among topics?

To answer the questions raised above, ideally, we would like to automatically construct a *"research theme evolution graph"*, which we illustrate in Figure 3.1. With such a graph, when zooming into the scope of individual topics, multiple types of information are provided to facilitate users to understand the research topic:

- *Topic Milestone Papers*: It is critical to recognize the papers that are best representative for a topic in the course

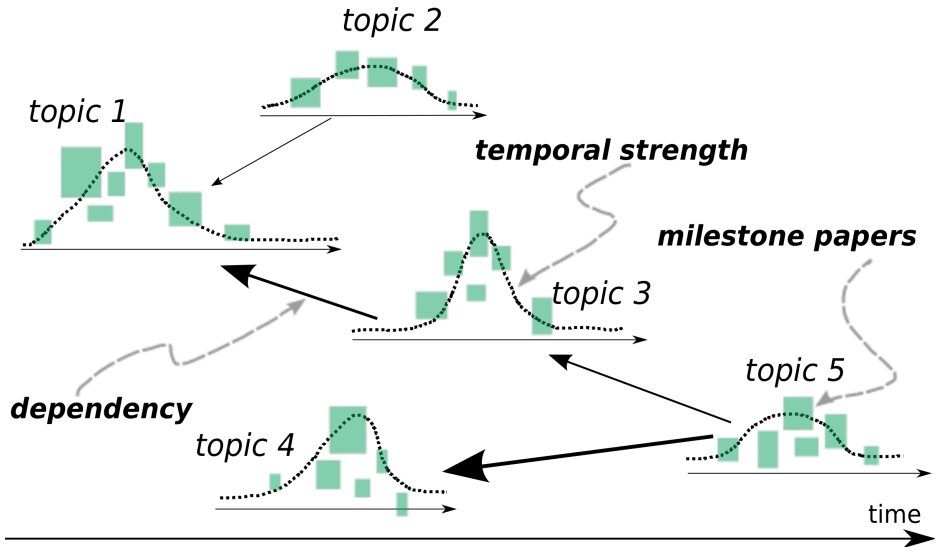


Figure 3.1: An illustration of the proposed evolution graph. We show 5 topics, and their dependency. Topic 2 and 3 are enabled by Topic 1 while Topic 5 is enabled by Topic 3 and 4.

of understanding topics. We refer to them as "topic milestone papers". Milestone papers of a topic provide a good picture how a topic is formed. In Figure 3.1, milestone papers are shown in each topic as rectangles and the "size" reflects their importance with respect to topics.

- *Topic Temporal Strength*: The relative popularity of topics at different times reveals the temporal nature of topics, which can help users to identify *current* vs. *previous* research topics as well as the rough topic life spans. Intuitively, when many milestone papers occur, the topic draws more attention and becomes popular.

- *Topic Keywords*: Extracting keywords that can properly summarize a topic would enable users to obtain a brief idea about the topic even without reading its relevant papers, allowing users to fast navigate among topics in search of the most interesting ones.

While zooming out to see the big picture of all related topics in the theme, there is also meaningful information to explore:

- *Topic Importance*: Quantifying the importance of topics helps a user to discriminate the *major* vs. *minor* topics in a research theme. Topic importance also reflects how well the topic is recognized by the community.

- *Topic Dependency*: Many new topics are built on top of the old ones. Discovering the dependency relation between topics provides a good guidance for users when searching for *origin/continuing* topics. In Figure 3.1, we visualize the dependency strength between topics by the "thickness" of edges.

- *Evolution Patterns*: Connecting topics by their dependency illustrates the underlying evolution patterns for research themes. Is there any trend that different topics get merged together to form a new (interdisciplinary) topic, such as Topic 3 and Topic 4 are merged into Topic 5? Or is there a general topic branched into multiple topics that address specialized problems, such as Topic 1 has led to Topic 2 and Topic 3?

To automatically construct such an evolution graph as shown in Figure 3.1, the two major computational tasks are:

- ***Discovering the research topics***, which includes finding milestone papers, computing the temporal strength, and extracting keywords for each individual topic.
- ***Discovering the theme evolution***, which includes identifying the topic importance and learning the dependency relation between topics, as well as recognizing the underlying evolution patterns.

Existing approaches, notably those of topic modeling, can generate some (not all) of these components in the evolution graph, but they are far from adequate for the following reasons: First, though there are many works that aim to construct evolution map over time, they rely on pre-segmentation of text streams into fixed time windows, due to either computational issue (Blei and Lafferty, 2006; Mei and Zhai, 2005; Wang and McCallum, 2006) or modeling issue (Wang et al., 2012). Consequently, the topic evolution result would be inevitably sensitive to the choice of temporal granularity of how time is discretized and sliced. Suboptimal granularity of time might result in missing important topics or even lead to inaccurate evolution analysis. Second, the edges in most of the existing evolution graphs, do not reflect the *dependency relation* between topics, and can only reveal the *topic similarity* and *correlation* (Blei and Lafferty, 2006, 2007; Mei and Zhai, 2005; Wang et al., 2012). The fundamental limitation is that content-based topic modeling approaches are built on *word co-occurrence*, which essentially is *undirected* unlike the dependency relation. Third, it is difficult for any aforementioned models (including Pairwise Link-LDA (Nallapati et al., 2008)) to assess the impact of documents with respect to different topics, i.e., identifying the milestone papers. Their approaches model topics as distributions over words, and although the text similarity between document and topic can be computed, it would be a substantially different measurement from the document *impact* on a topic.

As hinted above, a major reason why existing topic models are insufficient is that they have not fully exploited citation relations to discover topics. In this chapter, we address these limitations by doing joint analysis of citations and text. Indeed, we will rely more on citation links than on document content, which makes our work different from (Nallapati et al., 2008) and all others. Specifically, we leverage a similar idea to topic modeling and analyze the citation graphs in a *probabilistic* manner. We directly model the generation of citations, which are direct evidence related to *"impact"* of document as well as *"dependency"* between topics. Through citation generation, we are enabled to address the core problem of assessing milestone papers based on impact, and estimating the topic dependency. More importantly, our key insight here is that "co-cited papers" are good indicators of research topics, more effective than

relying on text similarity as in most existing work. Empirical study (Boyd-Graber et al., 2009) has already noticed that it is a subjective yet difficult task to annotate for each word its belonging topic even manually. However, for citations in a published paper written by experienced authors, it would be much easier to determine the topic since most authors make citations prudently and thus citation is much *less noisy* than text.

To discover topics based on citations, we propose a novel probabilistic approach to analyze citations by viewing citation graphs as a set of "citation documents" where each is a research paper represented as a *"bag of citations"*. A paper that cites k other (possibly duplicated) papers would simply be viewed as a *"document"* with k *"tokens"*, each corresponding to the ID of a cited paper. With this view, we can model all these citation documents with a generative topic model where we introduce latent topic variables over the citations. This is analogous to the application of a probabilistic topic model to model topics in text documents, but with the important difference that the discovered topics with our model would be characterized by a (multinomial) *distribution over research papers*, rather than over words as in conventional content-based topic models. In addition, when combined together with additional information, particularly the *published time* and the *title* of each paper, our model can address the computational tasks of discovering both *the research topics* and *the theme evolution*, and constructing *the evolution graph* as well.

In the rest of the chapter, we first review some of the related work in Section 3.2, which is followed by presenting our probabilistic model for literature citations in Section 3.3. After the derivation about one specific model Citation-LDA, we focus our discussion on how to construct the theme evolution graph in Section 3.4. Experiment setup and extensive evaluation results will be given in Section 3.5. Finally, we conclude our work with future direction in Section 3.6.

3.2 Related Work

In recent years, many literature search engines as well as digital libraries have come into use, including Microsoft Academic Search [1], Google Scholar [2], DBLP [3] and ACM Digital Library [4]. They provide knowledge about scientific literatures through ranking and search interface, which in turn, relies on algorithms that utilize citation-related indicators such as H-index (Hirsch, 2005) and Impact Factor (Garfield, 2006).

In the research community, one thread of study treats scientific literature as citation graphs. To assess the importance of papers, graph ranking algorithms such as PageRank and its variants have been applied (Ghosh et al., 2011; Radev et al., 2009; Sayyadi and Getoor, 2009; Walker et al., 2007). In (Ghosh et al., 2011), the authors further take time into consideration in order to overcome the recency bias that favors "old" papers. Apart from this, graph clustering is investigated to identify meaningful topics, such as (Bolelli et al., 2006; Flake et al., 2004; Popescul et al., 2000;

[1] http://academic.research.microsoft.com/
[2] http://scholar.google.com/
[3] http://www.informatik.uni-trier.de/ ley/db/
[4] http://dl.acm.org/

Qazvinian and Radev, 2008). In (Popescul et al., 2000), it is pointed out that efficient graph clustering can be combined with temporal information to identify the trends of topics in literature. Particularly, one recent paper (Jo et al., 2011) is close to our work. It leverages both citation and text (title and abstract) to generate the evolution map in computer science community. Specifically, their method relies on the temporal order of papers and the document language model to detect the formation of new topics, and then it computes the strength between two topics with the "cross citation count" (total citation numbers between the two topics), which however ignores the directed relation of topic dependency. Their method is difficult to be applied to address our problem because their method does not distinguish the difference in topic importance, nor does it recognize milestone papers through assessing the impact based on citations.

While on the other hand, existing probabilistic topic modeling over text (Blei et al., 2003; Griffiths and Steyvers, 2004; Hofmann, 2001) has been throughly studied, treating documents as mixtures of latent topics. Early attempt in modeling the topic evolution (Mei and Zhai, 2005) investigates the Probabilistic Latent Semantic Index (PLSI) (Hofmann, 2001) to extract topics and models the evolution process as transitions between topics in Hidden Markov Model (HMM). Later, Topic Over Time (TOT) model (Wang and McCallum, 2006) is developed based on Latent Dirichlet Allocation (LDA) (Blei et al., 2003). The key difference between between LDA and TOT is that TOT explicitly assumes time as generated from topics, which jointly models time and word, thus enabling itself to discover time-aware topics as well as topic temporal strength. Besides, Dynamic Topic Models (Blei and Lafferty, 2006; Wang et al., 2012) address the problem of topic evolution by modeling topics (distributions over words) changing over time. In the discrete case (Blei and Lafferty, 2006), topics at the next time-stamp deviate from the current ones by a Gaussian noise; while, in the continuous case (Wang et al., 2012), the change of topics over time is generalized as Brownian motion. One limitation of these models (Blei and Lafferty, 2006; Mei and Zhai, 2005; Wang et al., 2012; Wang and McCallum, 2006) is that they all rely on the pre-segmentation of time: without appropriate time granularity selected, they could fall into difficulty in finding important topics. Ideally, the selection of correct time span should be made automatically. In addition to these studies, others consider the problem of modeling topic correlation (Blei and Lafferty, 2007) and document hyperlink generation (Chang and Blei, 2009), for which the essential difficulty is that they cannot model the *"dependency"* relation between topics. The only exception we are aware of so far is the paper (Nallapati et al., 2008) which jointly models text and citation generatively. One of its proposed model, named "Pairwise Link-LDA", explicitly includes the topic dependency as model parameters by extending the idea of mixed-membership block stochastic models (Airoldi et al., 2006). In words, the chance of generating a particular citation is determined by the topics of citing and cited documents, which indeed addresses the topic dependency directly. Nevertheless, the Pairwise Link-LDA is not able to fulfill all the tasks we listed such as recognizing the milestone papers and so on.

To our best knowledge, there is no existing approach that can address all the questions as we raised before, i.e., the discovery of *research topics* and *theme evolution*. To this end, we directly model the generation of the citation links

among literatures in this work. In the same spirit of topic modeling, citations are generated stochastically according to a distribution with respect to the underlying topic. It is worth noting that applying the topic modeling approaches to study graphs was previously investigated for discovering communities from coauthorship networks in (Henderson and Eliassi-Rad, 2009; Zhang et al., 2007). Nevertheless, our model not only discovers the topics, but also explores their dependency relationships and yields meaningful knowledge about the evolution of topics.

3.3 Probabilistic Modeling of Literature Citations

In contrast to most existing work on citation analysis, where citations are often modeled as network or graph, we propose to represent citation graph as a set of "citation documents" where each is a research paper represented as "bag of citations", and model these citation documents with a probabilistic generative model. Such a new approach has several advantages over pure graph analysis methods. First, by using a latent topic variable, we can naturally associate topics with papers and citations, enabling ranking the paper based on citation within each topic, through which milestone papers can be identified. Second, by modeling the whole set of papers in a field, we can obtain a set of topics that summarize well the major research topics in the field, with (probabilistic) weights quantifying their importance. Third, by estimating the topic level citation structure, it is possible to compute the strength of dependency relation between topics and picturing the evolution paths of research themes. Last, distribution over papers for each topic obtained by such a model can be easily used to compute a distribution over time or keywords when used together with other information such as paper published time and title, allowing modeling the topic temporal strength and summarizing topics with keywords.

Compared with pure content-based topic models, our use of topic model is entirely on capturing topics through citation structures, roughly corresponding to discovering topics based on *co-citation relation*, which is intuitively more accurate in finding research topics: if there is a *"stable"* set of *"core papers"* that are often cited together, then it generally indicates the existence of a major research topic and the core papers are actually *milestone papers* in that topic. Specifically, we use a probabilistic model to explain how an author generates the references (citations) for a paper (which we may also refer to as a document for convenience sometimes). More specifically, given a paper, he/she would "generate" all the references cited in the paper independently. When generating each citation, the author would first sample a topic according to a document-specific topic distribution (doc_topic distribution), and then draw a reference document to cite from the citation distribution of the sampled topic ($topic\_doc$ distribution). One may easily notice that such a generation process is essentially similar to the one over words for documents assumed in probabilistic topic models for text data. Indeed, our work is a novel way of using topic models for citation analysis, and just as topic models are very effective for discovering and analyzing topics in *text documents*, our model can also be very useful

for discovering and analyzing topics in *scientific literatures* where the citation graph is available. Another advantage over content-based topic models we may anticipate is that the computational complexity is greatly reduced because the number of citations is much less than the number of words in the corpora.

3.3.1 The General Model

Formally, suppose each document d cites a subset of other documents $\{c_t\}$ $(t = 1, 2, \ldots)$, where c_t is a cited reference. We assume the following generation process for a citation that links to document c_t in document d (i.e., document d cites document c_t):

- Draw topic sample: $z_t \sim D_{doc\_topic}(z; d)$
- Draw citation sample: $c_t \sim D_{topic\_doc}(c; z_t)$

The doc-topic distribution $D_{doc\_topic}(\cdot; d)$ and topic-doc distribution $D_{topic\_doc}(\cdot; z)$ are parameterized by the citing document d and the topic z respectively, and are the two key components in the model that would enable many interesting ways to analyze topics and evolution relations among topics. Indeed, $D_{doc\_topic}(\cdot; d)$ gives us a probability distribution over (latent) topics conditioned on document d, and can be interpreted as the *topic coverage* in document d when generating citations, whereas $D_{topic\_doc}(\cdot; z)$ gives a *"reverse"* conditional distribution of documents given a topic, and can be interpreted as how a topic is characterized by a set of papers (documents) that are cited. Thus if a document c_i has a higher probability than c_j according to $D_{topic\_doc}(\cdot; z)$, it would suggests that c_i better characterizes topic z than c_j, or c_i represents topic z better as being a more important paper with higher impact upon z than c_j. With such a distribution over papers, we can easily compute the *expected time* for a topic based on the time when the paper was published as well as the *topic keywords* based on the paper titles (or abstracts if available). Note that a substantial advantage of such a probabilistic model is that it can *"decode"* why document d cites document c_t by inferring the latent topic associated with this citation relation and quantifying with uncertainty, which enables "disambiguation" of citation relations to some extent. As will be further discussed, we can use such a model to perform the computational analysis for discovering research topics and theme evolution, which finally lead to the construction of evolution graph as proposed in Figure 3.1.

3.3.2 Citation-LDA

Though we may have different ways to refine the general probabilistic model defined above, in this work as a first step, we focus on exploring the use of the basic Latent Dirichlet Allocation (LDA) (Blei et al., 2003) model, which we call "Citation-LDA" and show that even with this simple model setting, we can already discover a lot of interesting knowledge that is useful for understanding research theme evolution.

Specifically, Citation-LDA assumes that D_{doc_topic} and D_{topic_doc} are multinomial distributions with parameters drawn from conjugated Dirichlet prior α and β respectively [5]. We follow the convention to denote $D_{doc\_topic}(\cdot; d)$ and $D_{topic\_doc}(\cdot; z)$ by θ_d and ϕ_z respectively, and we have: $\theta_d \sim \mathrm{Dir}(\alpha)$ and $\varphi_z \sim \mathrm{Dir}(\beta)$. The citation generation process for document d_{i^*} is:

- Sample a topic $z = k^* \sim \mathrm{Multi}(\theta_{i^*})$
- Sample a document to cite $c = d_{j^*} \sim \mathrm{Multi}(\varphi_z)$

We use the collapsed Gibbs sampling (Griffiths and Steyvers, 2004) to make inferences with the model. The sampling is initialized by assigning random topic labels $\{z\}$ and updates each of them iteratively. In particular, for the t-th citation that links to d_{j^*} in document d_{i^*}, the topic assignment is updated according to the probability [6]:

$$\begin{aligned}
&\Pr(z = k^* | c_{i^*,t} = d_{j^*}, Z_{\neg(i^*,t)}, C^{\neg(i^*,t)}) \\
&\propto \left(\alpha_{k^*} + \#^{\neg(i^*,t)}(z = k^*, d = i^*)\right) \times \frac{\beta_{j^*} + \#^{\neg(i^*,t)}(z = k^*, c = d_{j^*})}{\sum_j \beta_j + \#^{\neg(i^*,t)}(z = k^*, c = d_j)}
\end{aligned} \tag{3.1}$$

The sampling converges to the true posterior distribution after the burn-in stage [7]. Posterior expectation of θ_{i^*,k^*} and φ_{k^*,j^*} is given by [8]:

$$\hat{\theta}_{i^*,k^*} = \left\langle \frac{\#(d = i^*, z = k^*) + \alpha_{k^*}}{\sum_k \#(d = i^*, z = k) + \alpha_k} \right\rangle \tag{3.2}$$

$$\hat{\varphi}_{k^*,j^*} = \left\langle \frac{\#(z = k^*, c = j^*) + \beta_{j^*}}{\sum_j \#(z = k^*, c = j) + \beta_j} \right\rangle \tag{3.3}$$

In addition, the empirical posterior distribution over topics can be computed as:

$$\hat{\Pr}(z = k^* | C) = \left\langle \frac{\#(z = k^*)}{\sum_k \#(z = k)} \right\rangle \tag{3.4}$$

[5] In experiments, α and β are symmetric prior with weight 1×10^{-3} to encourage sparse topic distributions

[6] We use $\#(\cdot)$ as the *count* function that computes the number of instances satisfy the conditions specified in ( ), and $\neg(i^*, t)$ denotes all the citations except the t-th citation in document d_{i^*}

[7] In experiments, this is empirically measured by parallel gibbs sampling

[8] We use $\langle\cdot\rangle$ to denote averaging the statistics specified over the iterations in sampling

3.4 Construction of Theme Evolution Graph

The results obtained from Equations (3.2) to (3.4) form the basis for exploring the knowledge that leads to the construction of the evolution graph, which includes the discovery of not only individual research topics but also theme evolution. We investigate them in details in following discussion.

3.4.1 Discovery of Research Topics

Zooming into individual topics identified by Citation-LDA, we are interested in finding *milestone papers*, generating *keywords*, and computing the *temporal strength* for each topic.

Topic Milestone Papers

The *topic-doc* distribution $\{\hat{\varphi}_{k,j}\}$, as computed in Equation (3.3) indicates how well a single paper d_j represents the topic z_k. The ranking of papers based on $\{\hat{\varphi}_{k,j}\}$ in essence provides the topic-aware impact assessment for papers with the milestone papers for topic z_k ranked at the top.

There are advantages over naive ranking of papers based on the citation counts, which can be inaccurate since there are cases that in one area people tend to include more references than people from another area. Even sophisticated citation-based measurement, e.g., (Ghosh et al., 2011; Radev et al., 2009; Sayyadi and Getoor, 2009; Walker et al., 2007), without taking into account of topics, can lead to bad judgement: a well recognized theoretic paper about graphic model in "Bayes learning" might receive less credit in "data engineering" and "very large database" due to the computational difficulty that limits its application.

Topic Temporal Strength

For topic z_k, there is a time point when it began attracting attention, a time point when it enjoyed its glory days with most important milestone papers emerged, and possibly a time point when interest decreased and the topic faded out. If it is a long lasting topic, it might span over decades while if not, the active period can be as short as only a few years.

Topic temporal distribution sufficiently maintains the information. Viewing topic z_k as a distribution over papers, the proportion of accumulated probability for published papers until time t forms the cumulative distribution function (CDF):

$$\begin{aligned}\Pr(time \leq t | z = k) &= \sum_{j, time(d_j) \leq t} \Pr(c = j | z = k) \\ &= \sum_{j, time(d_j) \leq t} \hat{\varphi}_{k,j} \end{aligned} \tag{3.5}$$

For the discrete time case, which is also our case, the probability mass function (PMF) for temporal distribution of z_k is:

$$\Pr(time = t|z = k) = \sum_{j,time(d_j)=t} \hat{\varphi}_{k,j} \tag{3.6}$$

In addition, the expectation can be computed as:

$$\mathbf{E}_{c|z=k}[\text{time}(c)] = \sum_j \text{time}(d_j)\hat{\varphi}_{k,j} \tag{3.7}$$

The standard deviation can also be easily computed, which, together with *topic expected time*, concisely show the major occurring time and provide a rough estimation about the life span for a topic.

Topic Keywords

In general it would be desirable to summarize the topic with only a few words (Boyd-Graber et al., 2009). With Citation-LDA, we accomplish this by leveraging words in title (or abstract if available) as tags for each paper and summarize the topic by those words with high *expected occurrences*. Specifically, to compute the word occurrence expectation over $\{\hat{\varphi}_{k,j}\}$ for word w in topic z_k:

$$\mathbf{E}_{c|z=k}[\#(w, c)] = \sum_j \hat{\varphi}_{k,j} \cdot \#(w, d_j) \tag{3.8}$$

As shown later in experiments, the topic keywords generated from titles are surprisingly indicative yet discriminative for especially seemingly similar topics.

3.4.2 Discovery of Theme Evolution

In order to help a researcher see the big picture of all research topics, we can also easily use Citation-LDA to discover the theme evolution, which would involve the exploration of assessing the *topic importance* as well as the *topic dependency relation*, and recognizing the underlying *evolution patterns*.

Topic Importance

By Equation (3.4), the distribution of $\{\hat{\Pr}(z = k)\}$ represents the chance of documents from one topic getting cited. Consequently, it can be associated as the topic importance in the research community since topics with higher importance are those who receive more citations and vice versa. The top important topics reflect the major research progress and reveal the dominant research interest in one area.

Topic Dependency

In Citation-LDA, topics are represented as multinomial distributions over papers $\{\hat{\varphi}_{k,j}\}$ while the *doc-topic* distribution $\{\hat{\theta}_{i,k}\}$ implies the topic mixture of document d_i. More precisely, $\hat{\theta}_{i,k^{(2)}}$ is the probability of topic $k^{(2)}$ occurring in document d_i with an (outlink) citation. Consequently, when marginalizing over papers d_j discounted by $\{\hat{\varphi}_{k^{(1)},j}\}$, the probability of citing topic $k^{(2)}$ (by topic $k^{(1)}$) conditioned on topic $k^{(1)}$ is:

$$\begin{aligned}
&\Pr(k^{(1)} \to k^{(2)}|k^{(1)}) \\
=&\mathbf{E}_{c|z=k^{(1)}}[\Pr(z=k^{(2)}|d=c)] \\
=&\sum_j \Pr(c=j|z=k^{(1)})\Pr(z=k^{(2)}|d=j) \\
=&\sum_j \hat{\varphi}_{k^{(1)},j}\hat{\theta}_{j,k^{(2)}}
\end{aligned} \tag{3.9}$$

An intuitive explanation of Equation (3.9) is: whenever randomly drawing a document d_j from topic $k^{(1)}$, and then emitting a citation from that document, $\Pr(k^{(1)} \to k^{(2)}|k^{(1)})$ is the chance of that citation being associated with *latent* topic $k^{(2)}$.

More importantly, Equation (3.9) explains the *topic level citation structure*, as well as quantifies the *topic dependency* between any two topics precisely — the amount of influence of topic $k^{(2)}$ upon topic $k^{(1)}$, from which we can tell if a topic is developed on top of another.

Evolution Patterns

Topic level citation structure $\{\Pr(k^{(1)} \to k^{(2)}|k^{(1)})\}_{K\times K}$ reveals the topic dependency. Nevertheless, it is indeed a $K \times K$ matrix with most entries being sparse. In our work, we propose two pruning criteria:

- *Threshold cutting-off*: By setting a threshold ξ [9] empirically, all citation dependencies between topics with strength less than ξ would be removed.
- *Temporal regularization*: As previously investigated in (Jo et al., 2011; Mei and Zhai, 2005), the citation dependencies of the "old" topics upon the "new" topics can be roughly regarded as noise and safely discarded.

After applying pruning to the *topic level citation structure*, significant yet meaningful influences between topics are kept. Closely dependent topics form the themes, in which different *evolution patterns* can be found: some topics may get merged into a new topic which is highly dependent on them (*merging*). Alternatively, one topic might have multiple

[9] $\xi = 0.1$ in experiments

Table 3.1: Top 10 High Impact Papers in Topic "Sentiment Analysis" (Topic 89, AAN)

$\hat{\varphi}$	Venue	Paper Title
0.078533	EMNLP'02	Thumbs Up? **Sentiment Classification** Using Machine Learning Techniques
0.067202	ACL'02	Thumbs Up Or Thumbs Down? **Semantic Orientation** Applied To Unsupervised Classification Of Reviews
0.048269	HLT'05	Recognizing Contextual Polarity In Phrase-Level **Sentiment Analysis**
0.043634	ACL'04	A Sentimental Education: **Sentiment Analysis** Using Subjectivity Summarization Based On Minimum Cuts
0.036498	ACL'97	Predicting The **Semantic Orientation** Of Adjectives
0.031173	COLING'04	Determining The **Sentiment Of Opinions**
0.030686	HLT'05	Extracting Product Features And **Opinions** From Reviews
0.028673	EMNLP'03	Towards Answering **Opinion** Questions: Separating Facts From **Opinions** And Identifying The **Polarity Of Opinion Sentences**
0.027851	EMNLP'03	Learning Extraction Patterns For **Subjective Expressions**
0.016856	ACL'05	Seeing Stars: Exploiting Class Relationships For **Sentiment Categorization** With Respect To Rating Scales

Table 3.2: Top 10 High Impact Papers in Topic "Air Pollution" (Topic 175, PMC)

$\hat{\varphi}$	Venue	Paper Title
0.035435	Environ_Health_Perspect	Ultrafine Particulate **Pollutants** Induce Oxidative Stress and Mitochondrial Damage
0.018051	Environ_Health_Perspect	Ambient **Air Pollution** and Atherosclerosis in Los Angeles
0.017836	Environ_Health_Perspect	Effects of **Air Pollution** on Heart Rate Variability: the VA Normative Aging Study
0.014414	Environ_Health_Perspect	Acute Blood Pressure Responses in Healthy Adults during Controlled **Air Pollution** Exposures
0.014233	Environ_Health_Perspect	The Effect of Particulate **Air Pollution** on Emergency Admissions for Myocardial Infarction
0.013984	Environ_Health_Perspect	Diabetes, Obesity, and Hypertension May Enhance Associations Between **Air Pollution** and Markers of Systemic Inflammation
0.013690	Environ_Health_Perspect	Nanotoxicology: an Emerging Discipline Evolving from Studies of **Ultrafine Particles**
0.013266	Environ_Health_Perspect	Association of Fine **Particulate** Matter From Different Sources With Daily Mortality in Six U.S. Cities
0.013090	Environ_Health_Perspect	**Ultrafine Particles** Cross Cellular Membranes by Nonphagocytic Mechanisms in Lungs and in Cultured Cells
0.012830	Environ_Health_Perspect	**Ambient Particulate Air Pollution**, Heart Rate Variability, and Blood Markers of Inflammation in a Panel of Elderly Subjects

subsequent topics that are developed on top of it (*branching*). In other cases, topics stop evolution and gradually *fade out*. We will discuss evolution patterns with concrete examples in the following experiment section.

3.5 Experiments & Results

In this section, we first formally describe the two datasets AAN and PMC on which we demonstrate our Citation-LDA. Further, extensive evaluation results of discovery of research topics and theme evolutions are discussed. Last, we show

that our Citation-LDA over-performs conventional Content-LDA baseline with two evaluation metrics: *forward-citation* and *journal conditioned entropy*.

Due to space limit, here we only show some representative results in our work. The complete results as well as the source code for Citation-LDA can be found at: http://sifaka.cs.uiuc.edu/~xwang95/citation_lda/

3.5.1 Dataset

In our experiments, two public scientific literature datasets are investigated: AAN from natural language processing domain and PMC from biomedical and life sciences.

ACL Anthology Network (AAN)

The ACL Anthology Network (AAN) (Radev et al., 2009) is a public dataset which includes all papers published by Association for Computational Linguistics (ACL) and related organizations over the period from 1965 till now. Major conference and journal papers in the area of natural language processing (NLP) can be found in the dataset. In our experiments, there are are in total $18,041$ papers (including citing and cited papers) from 13 venues with $82,944$ citations.

PubMed Central (PMC)

The PubMed Central (PMC) [10] is a free archive of biomedical and life sciences journal literature. Compared with AAN, it is a much larger yet sparser dataset, with a coverage of much wider areas than NLP. In our experiments, we includes the papers published after year 1960 and there are $145,317$ article papers with $274,133$ citations from $1,726$ journals.

Unlike AAN, the large number of journals in PMC provide a *"coarse topical annotation"* for papers, as in life sciences journals are commonly specialized in only a few research topics. For example, the journal *"Nucleic Acids Research"* covers research on nucleic acids such as DNA and RNA, but the journal *"Environmental Health Perspectives"* mainly publishes research on environmental health such as toxicology, exposure science and public health, etc. Later, we would utilize the journal information to evaluate the modeling performance of Citation-LDA and Content-LDA.

3.5.2 Results of Research Topics Discovery

Before the discussion of the results, however, a nontrivial question is how to determine the *number of topics* to be modeled? In following experiments, we perform the Citation-LDA with 100 topics in AAN and 500 topics in PMC, leaving the discussion of selecting the topic number in Section 3.5.4.

[10] http://www.ncbi.nlm.nih.gov/pmc/

Finding Milestone Papers

Milestone papers for two topics: "sentiment analysis" from AAN and "air pollution" from PMC, both of which are of great importance, are presented in Tables 3.1 and 3.2 respectively (10 milestone papers for each topic). Together, the *topic-doc* probability $\hat{\varphi}_{k,j}$ and the venue/journal sources are included. Clearly, the milestone papers listed are truly representative and recognized by the community based on the impact with respect to the topic.

One might notice that the top milestone papers in Table 3.2, unlike those of topic "sentiment analysis" from AAN, are actually all from one journal "Environmental Health Perspectives", which is generally regarded as among the most top tier journals in the area of "environment health" with especially established reputation in the topic "air pollution". In fact, the top milestone papers for topics in PMC being from the same (or only a few) journal(s) are actually quite common. Given that the journals in PMC are closely related to a variety of specialized topics, it can be taken as "noisy" topic labels of fair quality for evaluation purpose.

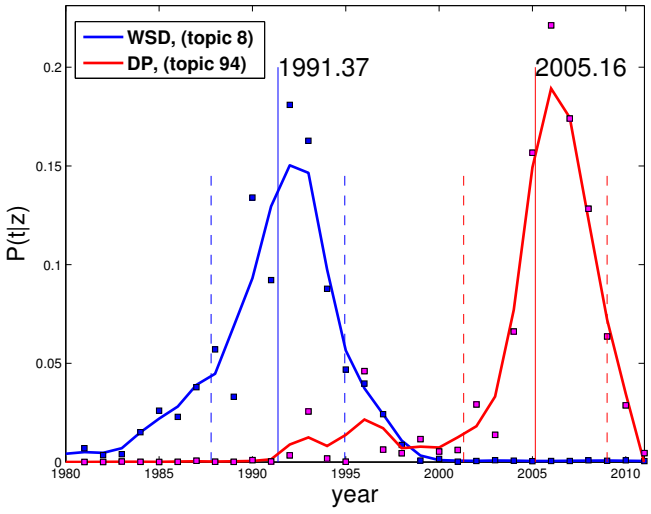


Figure 3.2: Topic Temporal Strength for "WSD" and "DP"

Discovering Temporal Strength

To demonstrate that our model discovers the topic over time correctly, we show the topic temporal strength of two topics, namely "word sense disambiguation" (WSD) and "dependency parsing" (DP) from AAN in Figure 3.2, and the computational details can be found in Equations (3.5) to (3.7).

In fact, the topic "WSD" was once a popular topic around early 90s while "DP" was newly popularized around year 2005. Based on our model, "WSD" has the expected time 1991.37, with a standard deviation 3.58. For "DP", the expectation is 2005.16 and standard deviation is 3.84. These estimations are all consistent with the expert knowledge.

Table 3.3: Dominant 10 Topics in AAN (100 topics)

Topic	Weight	$\mathbf{E}(t)$	$\mathrm{stdev}(t)$	Top Keyword Phrases
94	0.02806	2005.16	3.84	dependency parsing, non-projective, shared tasks, multilingual
89	0.02761	2004.64	3.25	sentiment classification, opinion analysis, orientation, learning
8	0.02509	1991.37	3.58	word sense disambiguation, lexical semantics
92	0.02428	2004.98	3.26	machine translation, phrase-based models, alignment
96	0.02277	2005.45	3.59	machine translation, online, margin, discriminative learning
84	0.02093	2003.94	3.36	semantic role labeling, shared tasks
80	0.02069	2003.44	3.83	machine translation, reordering, alignment
73	0.01965	2002.76	4.09	discriminative parsing, sequential labeling, part-of-speech
50	0.01908	2000.87	4.13	machine translation, minimum error rate training, BLEU evaluation
72	0.01804	2002.74	4.45	coreference resolution, machine learning, anaphora, pronoun

Table 3.4: Dominant 10 Topics in PMC (500 topics)

Topic	Weight	$\mathbf{E}(t)$	$\mathrm{stdev}(t)$	Top Keyword Phrases
484	0.00624	2006.45	8.95	protein, molecular interaction, biomolecular, database
499	0.00504	2007.36	9.89	ensemble, gene, genome, resources
488	0.00478	2006.48	19.37	gnome-scale metabolic reconstruction, escherichia coli, malaria
175	0.00450	2004.48	10.67	air pollution, ambient particulates, heart rates, exposure
373	0.00388	2005.35	11.77	non-coding RNA, sequence alignment, structure prediction, genome
492	0.00382	2006.56	11.39	sorcerer II, global ocean sampling, metagenomics, atlantic
61	0.00351	2003.22	12.12	children exposure, agricultural spraying, pesticides, organophosphorus
2	0.00350	1998.00	13.85	yeast, actin,saccharomyces cerevisiae, protein, myosin, cell
38	0.00338	2002.67	12.78	cell, regulatory T cell, CD4, CD25, human, Foxp3, expression, induction
86	0.00320	2003.64	14.12	phthalate exposure, human, urine, infants, metabolites, prenatal, health

Extracting Topic Keywords

We list the extracted keywords (phrases) [11] in Tables 3.3 and 3.4. As will be explained in details later, the topics are the dominant 10 topics in AAN and PMC datasets. The extracted keywords are mainly about the *problem, task, model* and *methodology* of the topics. For Topic 73 in AAN, it shows that the topic investigates the problem of "part-of-speech tagging", models the problem as "sequential labeling", and approaches it with "discriminative parsing" methods. For Topic 61 in PMC, the nature of the topic can be recovered as research on the risks of "children exposure" against "agricultural spraying" such as "pesticides" and "organophosphorus". In general, it is easy to conclude the research problems or detailed methodology for each topic through the extracted keywords along. Besides, based on the spotted keywords, Topic 92, Topic 96, Topic 80, and Topic 50 in AAN are all about the research theme "statistical machine translation". But keywords reveal that topics differ from each other as concerning about *distinct* methods/models (phrase-based models (92) *v.s.* discriminative learning (96)) or problems (reordering, alignment (80) *v.s.* evaluation (50)), which evidently substantiates that the keywords are adequately discriminative even for quite related topics, serving as accurate yet succinct summary for topics.

[11] Top word phrases are generated from top 20 keywords and then matched with n-grams in titles of the milestone papers

3.5.3 Results of Theme Evolution Discovery

Identifying Important Topics

As earlier implied, Tables 3.3 and 3.4 show the dominant 10 topics for AAN and PMC, which are selected based on the topic weight $\{\hat{\Pr}(z = k)\}$ as computed in Equation (3.4). Identified dominant topics cover major research progress and interest in NLP and life sciences. In AAN, it is obvious that the research theme "statistical machine translation" plays the most important role in the community, thriving and diverse with multiple different topics such as Topic 92, 96, 80, and 50. In PMC, many topics related to "public health" are dominant such as Topic 175, 61, and 86, though the detailed research topics are distinguishable from the keywords.

Taking the topic temporal strength into account,

$$\Pr(z = k, time = t) = \Pr(time = k|z = k) \cdot \hat{\Pr}(z = k)$$

is the joint probability of topic strength and time, allowing us to compare the topic strength in different time periods *with each other topics*. We visualize this for AAN and PMC in Figures 3.3 and 3.4, and it shows that the major research development occurred after year 2000 for both two dataset [12], except that Topic 8 ("word sense disambiguation") of AAN was dominant compared with others in early 90s while Topic 2 of "yeast", "saccharomyces cerevisiae" in PMC was a extensively studied around entire 90s.

Topic Dependency & Evolution Patterns

After applying the pruning to the *topic level citation structure* the evolution graph for research themes can be plotted. We show the evolution graph of AAN with 100 topics in Figure 3.5: each node represents a topic and the importance of topics are discriminated by the size of nodes. The green nodes are new topics while the red ones are *relatively* old. In addition, the dependency between topics are reflected by the thickness of edges .

There are three major connected component, each of which contains themes developing over time: Component 3 is about the theme "grammar", and corresponding topics entirely *faded out* during early 90s. Nevertheless, Component 2 has the theme of "discourse/dialogue" and "summarization", showing mildly progress recently (e.g., Topic 72 (2003) of *"machine learning" based "coreference resolution"*). Observing the Component 1, which is the largest, is interesting with discovery of various theme evolution patterns: Topic 8 (1991) about "word sense disambiguation" was *branched* into many topics, with one of them (Topic 18) being about "prepositional phrase attachment" (1994). Soon, Topic 18 further enabled Topic 34 (1999) of "statistical parsing", and again Topic 73 of "discriminative parsing" was established by 2003 on top of Topic 34. Later, Topic 94 of "dependency parsing" raised and has grown as one dominant topic since

[12] However, there is possibility that our datasets are biased as being rich in citations after year 2000

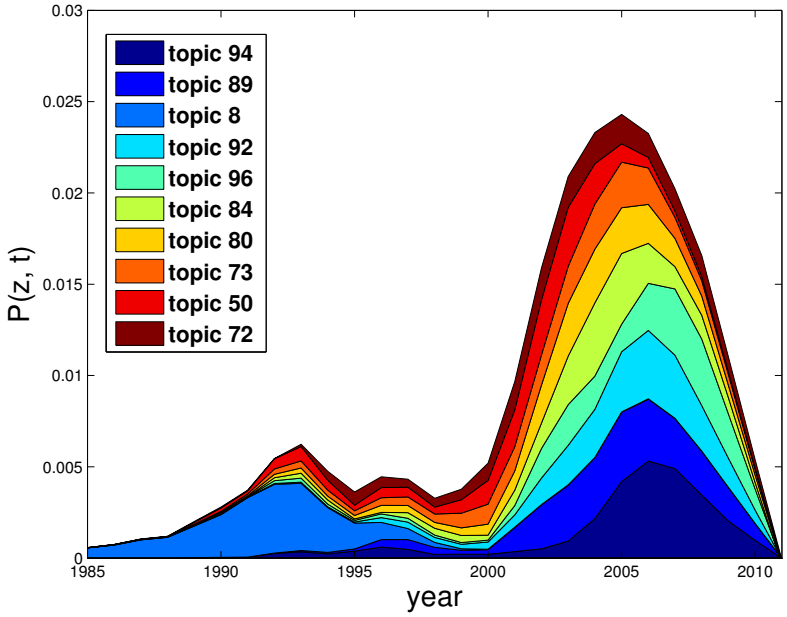


Figure 3.3: Topic-Temporal Joint Strength In AAN

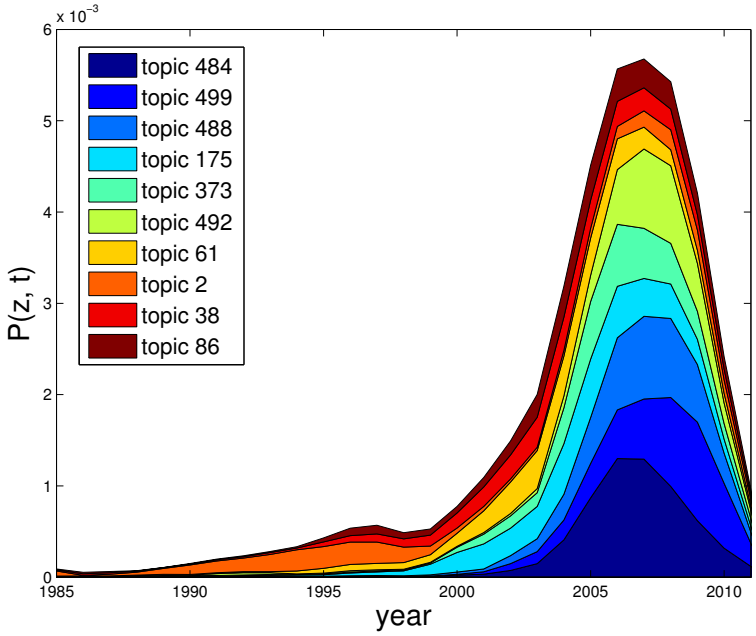


Figure 3.4: Topic-Temporal Joint Strength In PMC

2005.

Another key thread of theme in Component 1 was initiated by Topic 20, which was the very beginning topic of the theme "statistical machine translation" (SMT). The topics along the theme evolution path are presented in Table 3.5, including 4 topics (Topic 20, 29, 50, and 93), together with the milestone papers (top 3 for each). In addition,

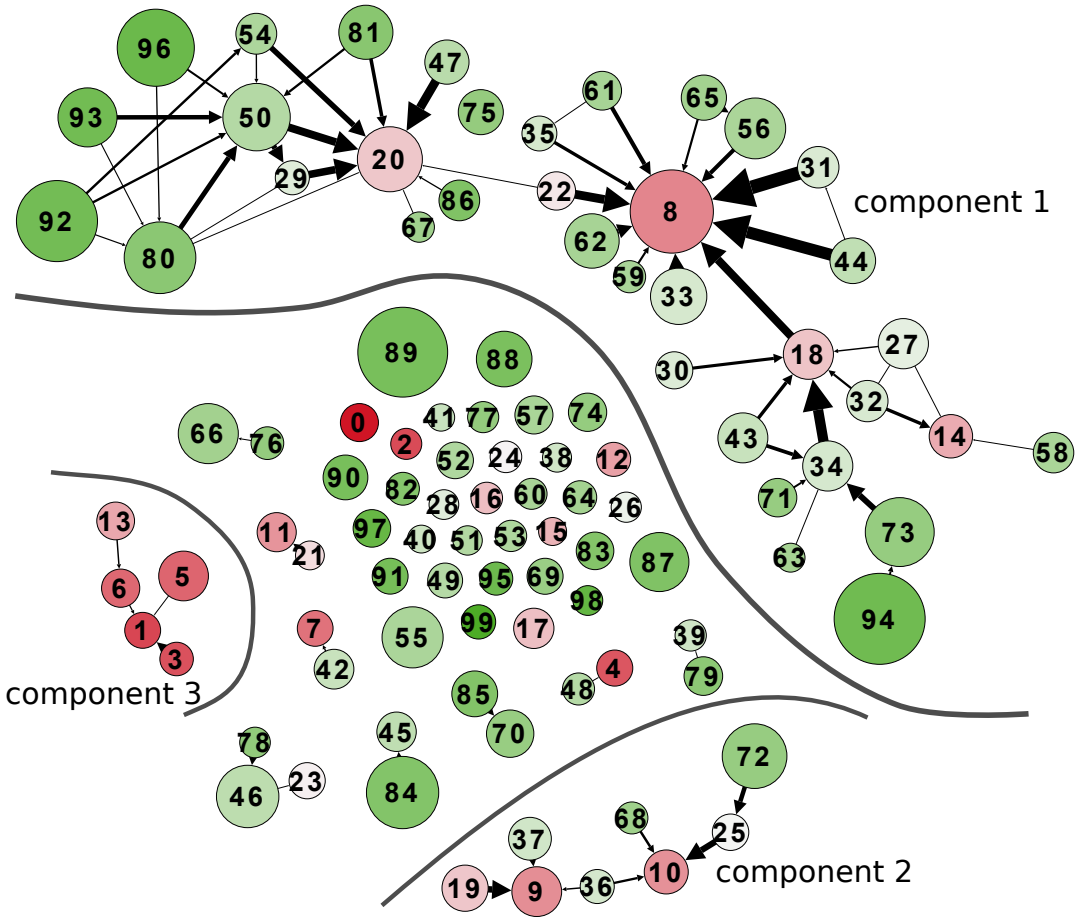


Figure 3.5: Theme Evolution Graph of AAN

the temporal distribution over time is given in Figure 3.6, where the citations among the milestone papers, and the dependency strength between consecutive topics are also depicted.

Specifically, Topic 20 began increasing its impact around early 90s, introducing basic statistical methods to machine translation; Later, around 1998, its popularity was shifted to Topic 29 which was specialized in subproblems such as "decoding", "alignment" and "reordering" in SMT; By 2002, however, Topic 50 emerged, and soon grew as the new dominant topic by proposing "BLEU" as the standard evaluation metric and investigating "discriminative methods" such as "minimum error rate training"; The current state of the art approach in SMT, "phrase-based model", accompanied by the raise of Topic 93, was actually built on top of previous work, especially milestone papers of $P7$-$P9$ of Topic 50.

In Figure 3.6, citation links among milestone papers across topics are illustrated, which clearly show the formation of topics through the "stable core set" of milestone papers that *get cited together* (co-cited). More importantly, it is evident that the "co-citation" of "core" papers is the direct contributing factor in the dependency relation between two consecutive topics.

Table 3.5: SMT Example for Theme Evolution

Topic	Year	Paper ID	Paper Title	$\hat{\varphi}$
Topic 20	1990	P1	A Statistical Approach To Machine Translation	0.036542
	1991	P2	A Program For Aligning Sentences In Bilingual Corpora	0.047619
	1993	P3	The Mathematics Of Statistical Machine Translation: Parameter Estimation	0.060931
Topic 29	1996	P4	HMM-Based Word Alignment In Statistical Translation	0.097162
	1997	P5	Decoding Algorithm In Statistical Machine Translation	0.030390
	1999	P6	Improved alignment models for statistical machine translation	0.036367
Topic 50	2002	P7	BLEU: A Method For Automatic Evaluation Of Machine Translation	0.087902
	2002	P8	Discriminative Training & Maximum Entropy Models For Statistical Machine Translation	0.027799
	2003	P9	Minimum Error Rate Training In Statistical Machine Translation	0.027027
Topic 93	2003	P10	Statistical Phrase-Based Translation	0.036239
	2005	P11	A Hierarchical Phrase-Based Model For Statistical Machine Translation	0.022442
	2007	P12	Hierarchical Phrase-Based Translation	0.043163

3.5.4 Model Selection & Comparison Results

We now discuss how to select the topic numbers for Citation-LDA and compare the performance with Content-LDA on two metrics, namely, *Forward Citation* and *Jounral Conditional Entropy*.

We investigate the conventional Content-LDA (Blei et al., 2003) as our baseline, using the title and abstract to represent the papers in both datasets. In order to make the output of Content-LDA aligned with that of Citation-LDA, we need to derive the missing *topic-doc* distribution: the distribution over papers (instead of tokens) for each topic. As in our experiments, we assume $\Pr(d|k) \propto \Pr(k|d) \cdot \Pr(d)$ whereas $\Pr(d) \propto |d|$ with $|d|$ being the document length for d.

Evluation on Forward Citation for AAN

We compute the *topic forward citation* probability based on the topic dependency (Equation (3.9)) and expected topic time (Equation (3.7)). In words, the forward citation probability reflects the chance a topic *cites* future topics that arise after itself (though it is impossible for a paper to cite a future paper). We compute the model's loss on topic k by the topic *future citation probability* , which is given by: $l(k) = \sum_{\tilde{k}, t(\tilde{k}) > t(k)} \Pr(k \to \tilde{k}|k)$ for topic k. To assess the total *loss for Forward Citation* of a model, we define it as follows:

$$\text{Loss}_{FC} = \sum_{k} \Pr(k) \cdot l(k)$$

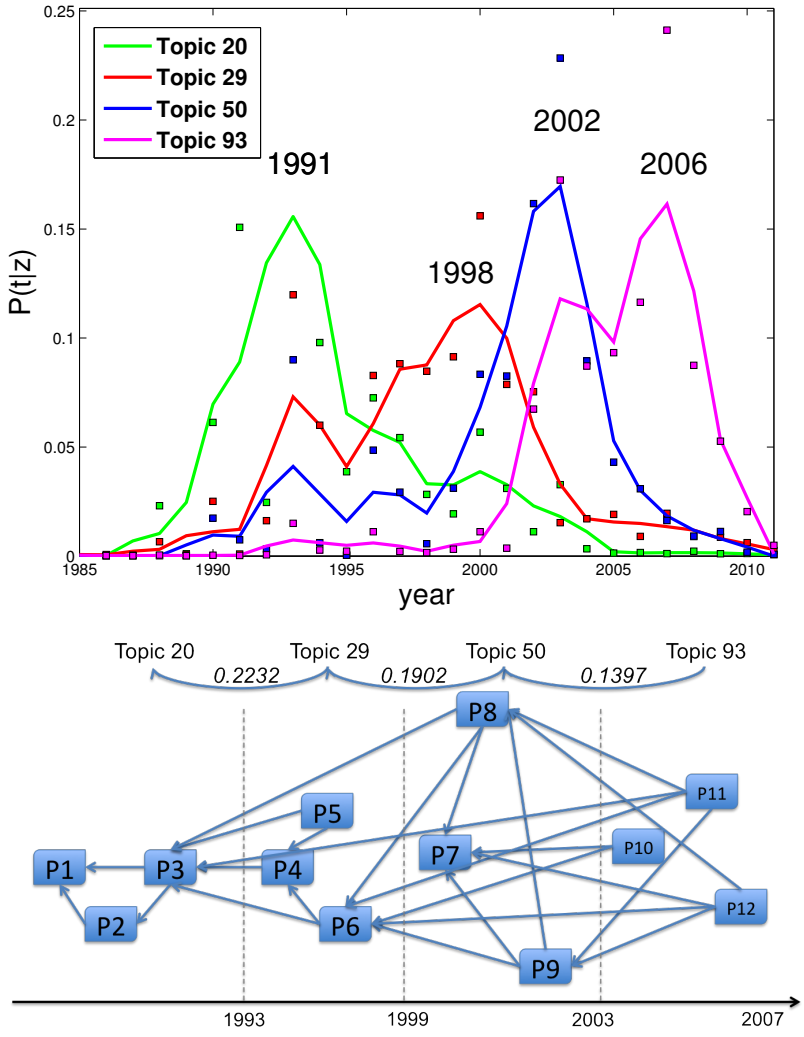


Figure 3.6: Temporal Evolution in Topics of Theme SMT

Table 3.6: Loss on Forward Citation (AAN)

#topic	20	100	200
Citation-LDA	0.3148	**0.1917**	0.2488
Content-LDA	0.3745	0.3816	0.3924

We show the evaluation based on *Forward Citation* for AAN in Table 3.6, from which we see: 1) Citation-LDA has better performance on Forward Citation compared with Content-LDA and 2) 100 topics are a good choice for AAN dataset.

Evaluation on Journal Conditional Entropy for PMC

As discussed before, the journal sources are fairly good "coarse" annotation for topics in PMC. For topic k, we can derive the *journal conditional distribution on topic* k, yielding the conditional entropy [13]:

$$H(J|z) = \sum_{z=k} \Pr(z = k) \cdot H(J|z = k)$$

The $H(J|z)$ would have low value if the journal labels and topic labels are *consistent*, by which we mean that for papers with the *same topic label* (in a probabilistic sense), there is *one journal label* being as dominant as possible, ideally being purely the only journal label. Hence, we can compute the *loss for Journal Conditional Entropy* of a model as:

$$\text{Loss}_{CE} = H(J|z)$$

Table 3.7: Loss on Journal Conditional Entropy (PMC)

#topic	100	300	500	1000
Citation-LDA	3.5047	3.2144	**3.18729**	3.4118
Content-LDA	4.2048	4.2805	4.06496	4.4725

Based on the journal conditional entropy on topics (Table 3.7), we again demonstrate the advantage of Citation-LDA over Content-LDA: the topic formed in Citation-LDA is more consistent with the "journal labels" than Content-LDA. In addition, we verify that for PMC dataset, 500 topics might be a reasonable choice.

3.6 Notes and Conclusion

In this chapter, we proposed a novel approach for analyzing research theme evolution of scientific literature data where citation links are available. 1) to discover research topics, which includes finding milestone papers, computing topic temporal strength, and extracting keywords for topics; 2) to discover theme evolution, which includes identifying topic importance, learning topic dependency relation, and recognizing the evolution patterns. These computational components together enable us to understand evolution of research themes by constructing the evolution graph. In experiments, we investigated two datasets, namely AAN and PMC from two domains, with extensive results showing that our proposed model, Citation-LDA, which represents article paper as "bag of citations" and model the generation of citation links within a probabilistic framework, can effectively accomplish the tasks defined above, with the performance

[13] Entropy $H(X) = -\sum_x \Pr(x) \log \Pr(x)$

better than Content-LDA. Our proposed Citation-LDA, together with the developed mining techniques, can be very useful to help researchers digest literature quickly, thus speeding up scientific research discovery and delivering very broad positive impact on the society.

In general, our model can also be applied to any graph data for tasks such as network clustering and ranking, as well as modeling the evolution of network generation, which we leave as future work directions.

Chapter 4

Blind Men and The Elephant: Thurstonian Pairwise Preference for Ranking in Crowdsourcing

4.1 Introduction

From the first Chapter we have seen that PLVM can be applied to model network data with good performance. Nevertheless, there are other types of data which possesses significantly different nature from the networks. In this chapter I present a framework where *ranked list* is inferred from pairwise preferences labelled by non-expert workers in crowdsourcing. Our approach leverages PLVM where latent variables are introduced to model query difficulty and query domain, as well as worker expertise and truthfulness. It also demonstrate that by employing latent variables, intractable distributions can be effectively sampled, and thus efficient computation is accomplished.

Collecting reliable annotation at scale has been a critical issue in the development of machine learning techniques. Crowdsourcing services make it possible to collect huge amount of annotations from less trained crowd workers in an inexpensive and efficient manner. The general philosophy of crowdsourcing is that instead of collecting one single expert-annotated label for each instance, multiple labels per example are collected from non-expert crowd workers at low cost to infer the ground truth (Welinder et al., 2010; Whitehill et al., 2009).

4.1.1 Motivation

In different tasks of learning, the form of labels can be as simple as binary/pairwise judgements, but can also be structured and complex. An example of the latter case is a ranked list of documents with respect to a query. *Ranked lists* offer the most informative knowledge for training and testing in various data mining and information retrieval tasks such as *learning to rank* (Valizadegan et al., 2009; Yue et al., 2007). Nevertheless, unlike making binary or pairwise judgements, labeling complex structures such as ranked lists by crowd workers is subject to large variance and low efficiency. In order to generate a ranked list of N items, a worker needs to consider a number of $N!$ possibilities. Annotation in such a huge labeling space is time consuming and uneconomic. Furthermore, the non-expert nature of crowdsourcing workers makes it even more difficult to reach consensus on the ground-truth ranked lists than binary/pairwise judgements.

The fact that ranked lists are highly useful but hard to be directly annotated motivates us to seek for alternative

strategies. Our idea is based upon a metaphor in which we can only *learn* what *an elephant* is like through *a group of blind men*. Each one holds onto a different part, but only one part, such as the side or the tusk. In the original story, they then discuss their observations which leads to argument and complete disagreement. However, a smarter treatment is to analysis all the observations and to find an probable explanation that most fits. In this chapter, we implement such idea by decomposing the task of labeling ranked lists into a series of smaller and easier tasks: *annotating pairwise preferences*, each of which requires a worker to compare only a pair of items out of the entire set. In addition, the pairwise judgements by crowd workers are more reliable and can be easily scaled up. Pairs of items can be randomly generated out of the set and will be labeled by multiple workers. The goal is to infer the true *ranked list* out of the *crowdsourced pairwise* annotations.

4.1.2 Challenges

Leveraging pairwise preferences to infer the full ranked list is promising but also challenging. The key challenge comes from *incomplete* and *inconsistent* annotations.

Pairwise preferences can be *incomplete* due to time and budget constraints. Not every two items are compared either directly or indirectly (For items A, B and C, an *indirect* annotation of $A \succ B$ may be obtained if *direct* annotations of $A \succ C$ and $C \succ B$ have been given). The available annotations can also be *inconsistent*, resulting from either the disagreements between multiple workers, or the intrinsic uncertainty within one single worker. A common mistake of the latter case is that one labels $A \succ B, B \succ C,$ and $C \succ A$ at the same time. The discussion below reveals a number of factors that lead to inconsistent annotations:

- *Query difficulty:* More difficult queries, such as ambiguous and vague queries, demand more effort to interpret and to judge, making them intrinsically more prone to errors.
- *Worker expertise across domains:* Different workers have different domain expertise; the same worker can also have varying domain knowledge across different task, making the quality of their labels vary accordingly. In practice, neither the task domain nor the worker's expertise is known apriori.
- *Truthfulness of Workers:* Truthfulness of workers is a prevailing issue in crowdsourcing tasks. Two typical adversarial groups are spammer workers and malicious workers: Spammers give random judgments and offer little information about the ranked lists; Malicious workers, on the other hand, sabotage the utility of annotations by giving false preferences.

Identifying the sources of such incompleteness and inconsistency, and properly modeling them, are critical to infer the true ranked list from the crowdsourced pairwise annotations.

4.1.3 Our Proposal

We propose a novel generative model called "Thurstonian Pairwise Preference" (TPP) to bind pairwise preferences of the crowd into rankings. The key modeling challenges that TPP addresses are to resolve the inevitable incompleteness and inconsistency of judgements, as well as to model variable query difficulty and different labeling quality resulting from workers' domain expertise and truthfulness.

TPP is built on top of the Thurstonian Ranking Model (TRM) (Thurstone, 1927), which takes noisy ranked lists of items as observations and estimates the true rankings. When applied to crowdsourcing, TRM models the generation of the noisy ranked lists annotated by crowd workers, taking variable query difficulty into account. It infers the relevance score of each item to form the ranked list. In contrast to TRM, the observations of TPP are pairwise preferences. Specifically, TPP naturally simulates the generative process of incomplete pairwise annotations, and seamlessly integrates a worker-aware layer with the original query-aware layer to model the inconsistency of the labeling process. The advantage of TPP is that it does not require full rankings as observations, and pairwise preferences can be efficiently labeled at scale.

While there have been earlier research efforts on (pairwise) ranking aggregation with similar goals, most of them investigated a "non-crowd" setting, or only a subset of the above factors are taken into account (See Section 4.6 for details). In sharp contrast, TPP provides a unified and principled strategy to handle various influential factors, which effectively binds pairwise preferences of the crowd into rankings.

Organization. We briefly introduce the original Thurstonian Ranking Model in Section 4.2, and present our proposed Thurstonian Pairwise Preference model (TPP) in Section 4.3. The inference of TPP is given in Section 4.4. We provide the experimental study in Section 4.5, review related work in Section 4.6 and conclude our study in Section 4.7.

4.2 Thurstonian Ranking Model

The original Thurstonian ranking model (TRM) (Thurstone, 1927) is devised for analyzing ordinal data. Suppose in a ranking annotation task, K workers $\{t_k\}_{k=1}^{K}$ are given Q queries $\{q_l\}_{l=1}^{Q}$ and D documents $\{d_i\}_{i=1}^{D}$. It is postulated that the optimal ranked list[1] for query q_l is determined by the *ground truth relevance score* $s_{l,i}$ of each document d_i. Precisely, the larger the value of $s_{l,i}$, the higher rank is assigned to d_i. Each worker t_k produces a ranked list $\sigma_l^{(k)}$ by ordering documents according to his *perceived relevance scores* $s_{l,i}^{(k)}$, which are assumed to be Gaussian distributed: $s_{l,i}^{(k)} \sim \mathrm{N}(s_{l,i}, \delta_l^2)$. The variance δ_l^2 quantifies the *query difficulty* of q_l: δ_l^2 is larger for more difficult query, and the perceived score can deviate more from the ground truth score.

The plate notation of the above generative process is given in Figure 4.1. With the workers' annotated rankings

[1] a permutation of documents

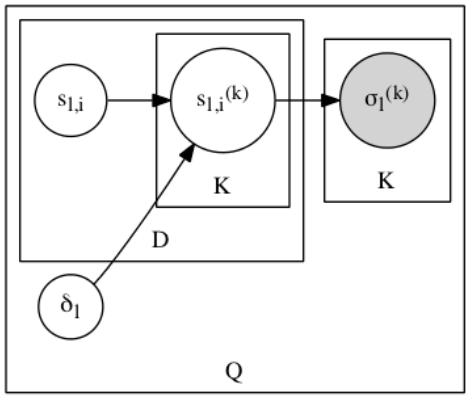


Figure 4.1: Plate notation for TRM

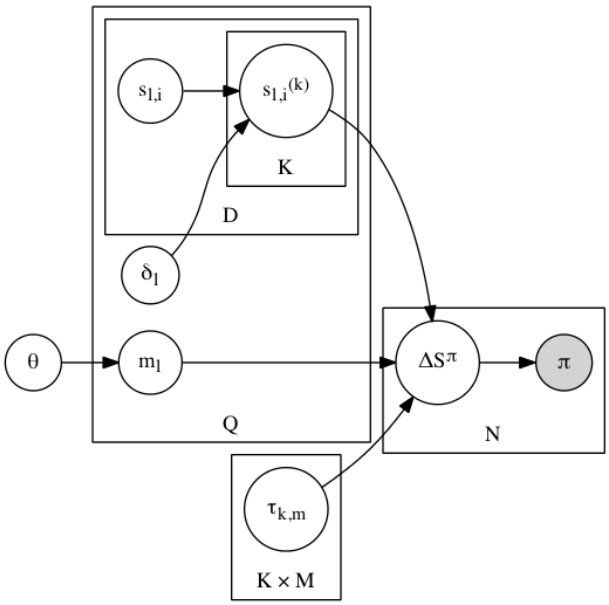


Figure 4.2: Plate notation for TPP

$\{\sigma_l^{(k)}\}$ given as observations, the goal of TRM is to infer $\{s_{l,i}\}$ as well as $\{\delta_l^2\}$. Algorithmic development for inference previously investigated includes maximum likelihood estimation (Böckenholt, 1993) and Bayesian inference (Yao and Böckenholt, 1999). A derivation of the maximum likelihood estimation is given in Appendix A.2.

4.3 Thurstonian Pairwise Preference

TRM specifies the generation of ranked lists in a crowdsourced setting, with variable query difficulty taken into account. However, the difficulty in obtaining annotated *ranked lists* makes it hardly applicable in practice. We propose a novel generative model called "Thurstonian Pairwise Preference" (TPP), which extends TRM to accommodate *pairwise*

Table 4.1: Summary of Notations

Notation	Explanation
t_k, q_l, d_i	worker t_k, query q_l and document d_i
$s_{l,i}$	ground truth relevance score of d_i *w.r.t.* q_l
δ_l^2	the difficulty of query q_l
m_l	the domain of query q_l
$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M)^T$	the distribution of query domains, $m_l \sim \text{Mult}(\boldsymbol{\theta})$
$\tau_{k,m}$	worker t_k's expertise & truthfulness on domain m
$s_{l,i}^{(k)}$	worker t_k's perceived score of d_i *w.r.t.* q_l
$\pi = \langle k, l, i_1, i_2 \rangle$	pairwise preference π: t_k prefers document d_{i_1} to document d_{i_2} *w.r.t.* q_l
$\tilde{s}_{i_1}^{\pi}, \tilde{s}_{i_2}^{\pi}$	noisy scores of d_{i_1} and d_{i_2} to determine pairwise preference π
Δs^{π}	noisy score difference $\Delta s^{\pi} = \tilde{s}_{i_1}^{\pi} - \tilde{s}_{i_2}^{\pi}$
$\boldsymbol{\Theta} = \{s_{l,i}, \delta_l^2, \theta_m, \tau_{k,m}\}$	model parameters
$\mathbf{Z} = \{m_l, s_{l,i}^{(k)}\}$	latent variables of interest
$\mathbf{V} = \{\Delta s^{\pi}\}$	auxiliary latent variables
$\mathbf{D} = \{\pi\}$	observations

preferences as observations. Meanwhile, TPP seamlessly integrates a *worker-aware* layer with the original query-aware layer to incorporate workers' variable expertise across different domains and their truthfulness, which explains the generation of the inconsistent pairwise preferences at modeling time.

The plate notation of TPP is given in Figure 4.2. The notations used throughout this chapter are summarized in Table 4.1. Suppose worker t_k compares documents d_{i_1} and d_{i_2} *w.r.t.* query q_l. The pairwise preference π is either t_k prefers d_{i_1} to d_{i_2}, denoted by $\langle k, l, i_1, i_2 \rangle$, or $\pi = \langle k, l, i_2, i_1 \rangle$ if t_k prefers d_{i_2}[2]. The preference depends on query difficulty, as well as the domain expertise and truthfulness of the worker.

TPP first generates the workers' perceived scores in the same way as TRM does. Then it introduces a worker-aware layer to simulate the generation of pairwise annotations, which involves a delicate modeling of query domains. We assume there are M domains. For query q_l, its domain m_l is drawn from a multinomial distribution: $m_l \sim \text{Mult}(\boldsymbol{\theta})$. In order to generate the pairwise preference π, worker t_k generates two noisy scores $\tilde{s}_{i_1}^{\pi}$ and $\tilde{s}_{i_2}^{\pi}$, which are Guassian distributed: $\tilde{s}_{i_1}^{\pi} \sim \text{N}(\text{sgn}(\tau_{k,m_l}) s_{l,i_1}^{(k)}, \tau_{k,m_l}^{-2})$ and $\tilde{s}_{i_2}^{\pi} \sim \text{N}(\text{sgn}(\tau_{k,m_l}) s_{l,i_2}^{(k)}, \tau_{k,m_l}^{-2})$.[3] The parameter $\tau_{k,m}$ encodes worker t_k's expertise and truthfulness on domain m. Specifically, the sign of $\tau_{k,m}$ indicates whether worker t_k is truthful or malicious on domain m. A malicious worker would have a negative $\tau_{k,m}$, giving false preferences by "flipping" his perceived scores. The absolute value of $\tau_{k,m}$ measures the expertise of t_k on m: a larger $|\tau_{k,m}|$ means

[2] We adopt the assumption made in TRM that no ties exist in rankings. However, if two documents are indeed equally relevant, the workers shall randomly prefers either one, and the ground truth relevance scores of the two documents would be close.

[3] $\text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{otherwise} \end{cases}$

a smaller variance of the noisy score, i.e., t_k is more knowledgeable on m; for a very small $|\tau_{k,m}|$, the noisy score is nearly uniformly distributed, implying t_k likely to be a spammer. Given the noisy scores $\tilde{s}^{\pi}_{i_1}$ and $\tilde{s}^{\pi}_{i_2}$, the pairwise preference is uniquely determined: $\pi = \langle k, l, i_1, i_2 \rangle$ if $\tilde{s}^{\pi}_{i_1} - \tilde{s}^{\pi}_{i_2} \geq 0$ and vice versa. We define the *noisy score difference* in this case as:

$$\Delta s^{\pi} = \tilde{s}^{\pi}_{i_1} - \tilde{s}^{\pi}_{i_2} \tag{4.1}$$

and thus $\mathrm{P}(\pi = \langle k, l, i_1, i_2 \rangle) = \mathrm{P}(\Delta s^{\pi} \geq 0)$.

The generative process of TPP is summarized as follows:

- **Generate Perceived Scores:** Generate worker t_k's perceived score of document d_i *w.r.t.* query q_l: $s^{(k)}_{l,i} \sim \mathrm{N}(s_{l,i}, \delta^2_l)$
- **Generate Query Domains:** For query q_l, draw its domain: $m_l \sim \mathrm{Mult}(\boldsymbol{\theta})$.
- **Generate Noisy Scores:** To compare two documents d_{i_1} and d_{i_2}, worker t_k generate noisy scores $\tilde{s}^{\pi}_{i_1}$ and $\tilde{s}^{\pi}_{i_2}$.

$$\tilde{s}^{\pi}_{i_j} \sim \mathrm{N}(\mathrm{sgn}(\tau_{k,m_l}) s^{(k)}_{l,i_j}, \tau^{-2}_{k,m_l}) \ (j = 1, 2) \tag{4.2}$$

- **Generate Pairwise Preferences:** The pairwise preference π is determined by the noisy score difference: $\pi = \langle k, l, i_1, i_2 \rangle$ if $\Delta s^{\pi} = \tilde{s}^{\pi}_{i_1} - \tilde{s}^{\pi}_{i_2} \geq 0$, and $\pi = \langle k, l, i_2, i_1 \rangle$ if $\Delta s^{\pi} < 0$.

Figure 4.3 illustrates the generation of two pairwise preferences by a crowd worker for a given query. The ground truth scores for three documents A, B, C imply the true ranking to be $A \prec B \prec C$. The worker's perceived scores deviate from the ground truth scores due to query difficulty. In fact, the perceived scores imply $A \prec C \prec B$, which contradicts with the true ranking. We further assume that the worker is truthful and has reasonable domain knowledge (This example does not include the generation of query domains for the sake of clarity). The worker generates noisy scores which are close to his perceived scores, and gives pairwise preferences ($A \prec C$, $C \prec B$) accordingly. It is worth noting that a pair of noisy scores are drawn each time a worker judges a pair of documents. Thus TPP respects intra-worker inconsistency as well as inter-worker inconsistency.

4.4 Inference

The model parameters $\boldsymbol{\Theta} = \{s_{l,i}, \delta^2_l, \theta_m, \tau_{k,m}\}$ are learned by Maximum Likelihood Estimation (MLE) with the Expectation-Maximization (E-M) (Dempster et al., 1977) algorithm. The posterior distribution of the *latent variables of interest* $\mathbf{Z} = \{m_l, s^{(k)}_{l,i}\}$ given the observations $\mathbf{D} = \{\pi\}$ is approximated via alternate sampling of $\mathbf{Z}$ and the *auxiliary latent variables* $\mathbf{V} = \{\Delta s^{\pi}\}$. The inference algorithm of TPP is summarized in Algorithm 1.

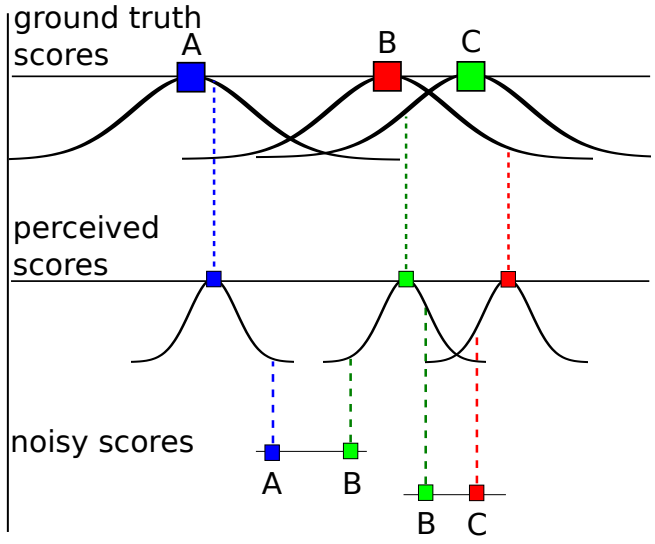


Figure 4.3: An Illustration Example of TPP: The generation of two pairwise preferences by a crowd worker for a given query

The true ranking is determined by the ground truth scores of each document. The perceived score of each document is Gaussian distributed based on the true score and the query difficulty. Each time a worker is asked to compare a pair of documents, The perceived scores, together with the domain expertise and truthfulness of the worker, specify another two Gaussian distributions from which the noisy scores are drawn. The pairwise preference is given accordingly. The worker is truthful in this example.

Algorithm 1: Inference of TPP

Input: Pairwise preferences $\mathbf{D}$
Output: Model parameters $\mathbf{\Theta}$
Initialize $\mathbf{V}, \mathbf{Z}, \mathbf{\Theta}$;
while *convergence criteria not met* **do**
    (E-step) Sample the posterior distribution of $\mathbf{V}$ and $\mathbf{Z}$;
    (M-step) Update $\mathbf{\Theta}$;
    Model rescaling;

4.4.1 Model Parametrization

The pairwise preference $\pi = \langle k, l, i_1, i_2 \rangle$ between two documents d_{i_1} and d_{i_2} hinges on $\Delta s^\pi = \tilde{s}^\pi_{i_1} - \tilde{s}^\pi_{i_2}$. We introduce *auxiliary latent variables* $\mathbf{V} = \{\Delta s^\pi\}$ to parameterize TPP.

Our results rely on the following lemma of (truncated) Gaussian distribution, the proof of which can be found in (Chopin, 2011)):

Lemma 4.4.0.1. *If x_1 and x_2 are independently sampled from $x_i \sim \mathrm{N}(\mu_i, \sigma^2),\ i = \{1, 2\}$, then we have*

(a) $x_1 - x_2 \sim \mathrm{N}(\mu_1 - \mu_2, 2\sigma^2)$ *and* $\mathrm{P}(x_1 - x_2 \geq 0) = Q(-\frac{\mu_1 - \mu_2}{\sqrt{2}\sigma})$*, where $Q(\cdot)$ denotes the tail probability of the standard normal distribution: $Q(s) := \Pr(x \geq s)$, $x \sim \mathrm{N}(0, 1)$.*

(b) $x_1 - x_2 | x_1 - x_2 \geq 0 \sim \mathrm{TN}_0^{\infty}(\mu_1 - \mu_2, 2\sigma^2)$*, where* $\mathrm{TN}_a^b(m, s^2)$ $(a < b, a, b \in \mathbb{R} \cup \{\pm\infty\})$ *is the truncated Gaussian distribution bounded by interval* (a, b) *with the embedded Gaussian distribution being* $\mathrm{N}(m, s^2)$*.* □

Given Equations (4.1) and (4.2), it follows from Lemma 4.4.0.1(a) that the auxiliary latent variable Δs^{π} follows the truncated Gaussian distribution:

$$\tilde{s}_{i_1}^{\pi} - \tilde{s}_{i_2}^{\pi} | \tau_{k,m_l}, s_{l,i_1}^{(k)}, s_{l,i_2}^{(k)} \sim \mathrm{N}\left(\mathrm{sgn}(\tau_{k,m_l})(s_{l,i_1}^{(k)} - s_{l,i_2}^{(k)}), 2\tau_{k,m_l}^{-2}\right) \tag{4.3}$$

and we have

$$\mathrm{P}(\Delta s^{\pi} \geq 0 | \tau_{k,m_l}, s_{l,i_1}^{(k)}, s_{l,i_2}^{(k)}) = \mathrm{Q}\left(-\frac{\tau_{k,m_l}}{\sqrt{2}}(s_{l,i_1}^{(k)} - s_{l,i_2}^{(k)})\right) \tag{4.4}$$

In view of the above results, the joint probability of $\mathbf{D}, \mathbf{Z}, \mathbf{V}$ can be factorized as:

$$\begin{aligned}
&\mathrm{P}(\mathbf{D}, \mathbf{Z}, \mathbf{V} | \boldsymbol{\Theta}) = \mathrm{P}(\mathbf{Z} | \boldsymbol{\Theta}) \mathrm{P}(\mathbf{V} | \mathbf{Z}) \mathrm{P}(\mathbf{D} | \mathbf{V}) \\
=& \prod_l \mathrm{P}_{\texttt{Mult}}(m_l | \boldsymbol{\theta}) \cdot \prod_{k,l,i} \mathrm{P}_{\mathrm{N}}(s_{l,i}^{(k)} | s_{l,i}, \delta_l^2) \\
& \prod_{\pi = \langle k,l,i_1,i_2 \rangle \in \mathbf{D}} \left(\mathrm{P}_{\mathrm{N}}\left(\Delta s^{\pi} | \mathrm{sgn}(\tau_{k,m_l})(s_{l,i_1}^{(k)} - s_{l,i_2}^{(k)}), 2\tau_{k,m_l}^{-2}\right)\right. \\
& \prod_{\pi = \langle k,l,i_1,i_2 \rangle \in \mathbf{D}} \mathbf{1}(\Delta s^{\pi} \geq 0)
\end{aligned} \tag{4.5}$$

Integrating out $\mathbf{V}$, we get the joint probability of the observations $\mathbf{D} = \{\pi\}$ and the latent variables of interest $\mathbf{Z} = \{m_l, s_{l,i}^{(k)}\}$:

$$\begin{aligned}
&\mathrm{P}(\mathbf{D}, \mathbf{Z} | \boldsymbol{\Theta}) = \int_{\mathbf{V}} \mathrm{P}(\mathbf{D}, \mathbf{Z}, \mathbf{V} | \boldsymbol{\Theta}) \, \mathrm{d}\mathbf{V} \\
=& \prod_l \mathrm{P}_{\texttt{Mult}}(m_l | \boldsymbol{\theta}) \cdot \prod_{k,l,i} \mathrm{P}_{\mathrm{N}}(s_{l,i}^{(k)} | s_{l,i}, \delta_l^2) \\
& \prod_{\pi = \langle k,l,i_1,i_2 \rangle \in \mathbf{D}} \mathrm{Q}\left(-\frac{\tau_{k,m_l}}{\sqrt{2}}(s_{l,i_1}^{(k)} - s_{l,i_2}^{(k)})\right)
\end{aligned} \tag{4.6}$$

The model parameters $\boldsymbol{\Theta} = \{s_{l,i}, \delta_l^2, \theta_m, \tau_{k,m}\}$ are learned by optimizing the log likelihood $\boldsymbol{\Theta} = \arg\max_{\boldsymbol{\Theta}} \ln \mathrm{P}(\mathbf{D} | \boldsymbol{\Theta})$ with the E-M algorithm. At the t-th iteration, the posterior distribution of $\mathbf{Z} | \mathbf{D}, \boldsymbol{\Theta}^{(t)}$ is computed (E-step), followed by

the model update, i.e. maximizing the expected joint log likelihood: $\mathbf{\Theta}^{(t+1)} = \arg\max_{\mathbf{\Theta}} \mathcal{Q}(\mathbf{\Theta}; \mathbf{\Theta}^{(t)})$ (M-step), where

$$\mathcal{Q}(\mathbf{\Theta}; \mathbf{\Theta}^{(t)}) = \mathbf{E}_{\mathbf{Z}|\mathbf{D},\mathbf{\Theta}^{(t)}}[\ln \mathrm{P}(\mathbf{D}, \mathbf{Z}|\mathbf{\Theta})] \tag{4.7}$$

4.4.2 Posterior Sampling

The analytic calculation of $\mathcal{Q}(\mathbf{\Theta}; \mathbf{\Theta}^{(t)})$ is impossible due to the intractability of $\mathrm{P}(\mathbf{Z}|\mathbf{D}, \mathbf{\Theta})$. Instead, we approximate the posterior distribution by sampling. Nevertheless, sampling $\mathbf{Z}$ from $\mathrm{P}(\mathbf{Z}|\mathbf{D}, \mathbf{\Theta})$ is still difficult because we cannot effectively integrate over Equation (4.6) to obtain the distribution of $s_{l,i}^{(k)}|\mathbf{Z} \setminus \{s_{l,i}^{(k)}\}, \mathbf{D}, \mathbf{\Theta}$. Therefore, we reintroduce the auxiliary latent variables $\mathbf{V}$. A blocked Gibbs sampler (Geman and Geman, 1984) is applied to sample $\mathbf{V}$ and $\mathbf{Z}$. Each block of variables, i.e., query domains $\{m_l\}$, perceived scores $\{s_{l,i}^{(k)}\}$, and noisy score differences $\{\Delta s^\pi\}$, are sampled in sequence.

Sample Query Domain m_l

It follows from Equation (4.5) that the posterior distribution of the domain m_{l^*} for a query l^* is given by the following multinomial distribution:

$$\begin{aligned} &\mathrm{P}(m_{l^*} = m^*|\mathbf{D}, \mathbf{Z} \setminus \{m_{l^*}\}, \mathbf{V}, \mathbf{\Theta}) \\ \propto\ & \theta_{m^*} \prod_{\substack{\pi=\langle k,l,i_1,i_2\rangle \in \mathbf{D} \\ l=l^*}} \mathrm{P}_{\mathtt{N}}\left(\Delta s^\pi | \mathrm{sgn}(\tau_{k,m^*})(s_{l^*,i_1}^{(k)} - s_{l^*,i_2}^{(k)}), 2\tau_{k,m^*}^{-2}\right) \end{aligned} \tag{4.8}$$

Note that there is no coupling (inter-dependency) among $\{m_l\}$, and the multinomial sampling can be accelerated with parallel implementation.

Sample Perceived Score $s_{l,i}^{(k)}$

It follows from Equation (4.5) that the posterior distribution for the perceived score is given by:

$$\begin{aligned} &\mathrm{P}(s_{l^*,i^*}^{(k^*)} = s^*|\mathbf{D}, \mathbf{Z} \setminus \{s_{l^*,i^*}^{(k^*)}\}, \mathbf{V}, \mathbf{\Theta}) \\ \propto\ & \mathrm{P}_{\mathtt{N}}(s^*|s_{l^*,i^*}, \delta_{l^*}^2) \\ & \prod_{\pi=\langle k^*,l^*,i^*,i\rangle \in \mathbf{D}} \mathrm{P}_{\mathtt{N}}(\Delta s^\pi | \mathrm{sgn}(\tau_{k^*,m_{l^*}})(s^* - s_{l^*,i}^{(k^*)}), 2\tau_{k^*,m_{l^*}}^{-2}) \\ & \prod_{\pi=\langle k^*,l^*,i,i^*\rangle \in \mathbf{D}} \mathrm{P}_{\mathtt{N}}(\Delta s^\pi | \mathrm{sgn}(\tau_{k^*,m_{l^*}})(s_{l^*,i}^{(k^*)} - s^*), 2\tau_{k^*,m_{l^*}}^{-2}) \end{aligned} \tag{4.9}$$

To derive the sampling rule for perceived score $s_{l,i}^{(k)}$, we employ the following lemma that an exponential-family distribution is uniquely determined by its sufficient statistics and natural parameters (Stuart et al., 1968):

Lemma 4.4.0.2. *If* $\mathrm{P}(x)$ *is a valid distribution and* $\mathrm{P}(x) \propto \exp(c_1 x + c_2 x^2)$*, then* $x \sim \mathrm{N}(-\frac{c_1}{2c_2}, -\frac{1}{2c_2})$ □

And it follows immediately that:

$$s_{l^*,i^*}^{(k^*)} \sim \mathrm{N}(\frac{a_1}{a_2}, \frac{1}{a_2}) \tag{4.10}$$

where

$$\begin{aligned} a_1 = & \frac{1}{\delta_{l^*}^2} s_{l^*,i^*} \\ & + \frac{1}{2\tau_{k^*,m_{l^*}}^{-2}} \left(\sum_{\substack{i \\ \pi=\langle k^*,l^*,i^*,i\rangle \in \mathbf{D}}} s_{l^*,i}^{(k^*)} + \mathrm{sgn}(\tau_{k^*,m_{l^*}}) \Delta s^{\pi} \right) \\ & + \frac{1}{2\tau_{k^*,m_{l^*}}^{-2}} \left(\sum_{\substack{i \\ \pi=\langle k^*,l^*,i,i^*\rangle \in \mathbf{D}}} s_{l^*,i}^{(k^*)} - \mathrm{sgn}(\tau_{k^*,m_{l^*}}) \Delta s^{\pi} \right) \end{aligned} \tag{4.11}$$

$$a_2 = \frac{1}{\delta_{l^*}^2} + \frac{1}{2\tau_{k^*,m_{l^*}}^{-2}} \left(\sum_{\substack{i \\ \langle k^*,l^*,i^*,i\rangle \in \mathbf{D}}} 1 + \sum_{\substack{i \\ \langle k^*,l^*,i,i^*\rangle \in \mathbf{D}}} 1 \right) \tag{4.12}$$

Intuitive Interpretation. Here is an intuitive interpretation of the above calculation which provides more insights into the behaviors of TPP:

First, the mean value $\frac{a_1}{a_2}$ is a weighted average of three sources of estimation:

- s_{l^*,i^*}, the ground truth relevance score (1st term in Equation (4.11)). It is discounted by the query difficulty $\delta_{l^*}^2$. The easier the query, the more it contributes to the perceived score $s_{l^*,i^*}^{(k^*)}$.
- $\left(s_{l^*,i}^{(k^*)} + \mathrm{sgn}(\tau_{k^*,m_{l^*}}) \Delta s^{\pi} \right)$ where $\pi = \langle k^*, l^*, i^*, i \rangle \in \mathbf{D}$, (2nd term in Equation (4.11)). It corresponds to a pairwise preference π when t_{k^*} *prefers* d_{i^*} to the other document d_i. It estimates $s_{l^*,i^*}^{(k^*)}$ by combining the perceived score $s_{l^*,i}^{(k^*)}$ of the less preferred document d_i and the noisy score difference $\Delta s^{\pi} = \tilde{s}_{i^*}^{\pi} - \tilde{s}_i^{\pi}$ multiplied by the worker's truthfulness $(\mathrm{sgn}\,(\tau_{k^*,m_{l^*}}))$. This estimation is then weighted by the worker's domain expertise $(\frac{1}{2}\tau_{k^*,m_{l^*}}^2)$.
- The third source of estimation (3rd term in Equation (4.11)) corresponds to the case when d_{i^*} is less preferred by t_{k^*}. The analysis is analogous to that of the 2nd term.

In addition, the variance $\frac{1}{a_2}$ in Equation (4.10) is the harmonic average of the query difficulty $\delta_{l^*}^2$ and the worker's domain expertise $2\tau_{k^*,m_{l^*}}^{-2}$, which determines the uncertainty of the perceived score $s_{l^*,i^*}^{(k^*)}$. The sampled perceived scores are more localized to the mean value $\frac{a_1}{a_2}$ with easier queries and more knowledgeable workers.

Sample Noisy Score Difference Δs^π

Denote the pairwise preference by $\pi^* = \langle k^*, l^*, i_1^*, i_2^* \rangle$. It follows from Equation (4.5) that

$$\begin{aligned} & \mathrm{P}(\Delta s^{\pi^*} = \Delta s^* | \mathbf{D}, \mathbf{Z}, \mathbf{V} \setminus \{\Delta s^{\pi^*}\}, \mathbf{\Theta}) \\ \propto\ & \mathrm{P}_{\mathrm{N}}\left(\Delta s^* | \mathrm{sgn}(\tau_{k^*,m_{l^*}})(s_{l^*,i_1^*}^{(k^*)} - s_{l^*,i_2^*}^{(k^*)}), 2\tau_{k^*,m_{l^*}}^{-2}\right) \mathbf{1}(\Delta s^* \geq 0) \end{aligned} \tag{4.13}$$

By Lemma 4.4.0.1(b), the posterior distribution of Δs^{π^*} is a truncated Gaussian distribution:

$$\mathrm{TN}_0^\infty\left(\mathrm{sgn}(\tau_{k^*,m_{l^*}})(s_{l^*,i_1^*}^{(k^*)} - s_{l^*,i_2^*}^{(k^*)}), 2\tau_{k^*,m_{l^*}}^{-2}\right)$$

Efficient sampling from a truncated Gaussian distribution can be found in (Chopin, 2011).

With the above sampling rules, $\{m_l\}$, $\{s_{l,i}^{(k)}\}$, and $\{\Delta s^\pi\}$ are sampled in blocks. After the burn-in period, samples of $\mathbf{Z}$ are collected to approximate the posterior distribution $\mathrm{P}(\mathbf{Z}|\mathbf{D}, \mathbf{\Theta})$ (samples of $\mathbf{V}$ are discarded).

4.4.3 Model Updating

The model parameters are updated by

$$\begin{aligned} \Theta^{(t+1)} &= \arg\max_{\Theta} \mathcal{Q}(\mathbf{\Theta}; \mathbf{\Theta}^{(t)}) \\ \text{where } \mathcal{Q}(\mathbf{\Theta}; \mathbf{\Theta}^{(t)}) &= \mathbf{E}_{\mathbf{Z}|\mathbf{D},\mathbf{\Theta}^{(t)}}[\ln \mathrm{P}(\mathbf{D}, \mathbf{Z}|\mathbf{\Theta})] \end{aligned} \tag{4.14}$$

with the posterior distribution $\mathbf{Z}|\mathbf{D}, \mathbf{\Theta}^{(t)}$ approximated by blocked Gibbs sampling.

Optimization details are given in Appendix A.1. Closed forms are obtained for the update of ground truth scores $\{s_{l,i}\}$, query difficulties $\{\delta_l^2\}$, and the domain distribution $\{\theta_m\}$. (Inexact) Newton's method is applied to update the domain expertise and truthfulness of workers $\{\tau_{k,m}\}$.

4.4.4 Identifiability

Identifiability is a property which a model must satisfy in order for precise inference to be possible. In plain words, it requires that different values of the parameters must generate different probability distributions of the observable variables.

For modeling rankings of documents, the extra degree of freedom of the model can potentially lead to an arbitrary scaling of the ground truth scores (or parameters), and thus must be carefully avoided.

One may observe that for the same collection of observations $\mathbf{D}$, the following two models have the same likelihood $\mathrm{P}(\mathbf{D}|\mathbf{\Theta_1}) = \mathrm{P}(\mathbf{D}|\mathbf{\Theta_2})$ for any global factor $\sigma > 0$ and query-level biases $\{b_l\}$.[4]

$$\begin{aligned} \mathbf{\Theta_1} &= \{s_{l,i}, \delta_l^2, \theta_m, \tau_{k,m}\} \\ \mathbf{\Theta_2} &= \{(s_{l,i} - b_l)/\sigma, \delta_l^2/\sigma^2, \theta_m, \tau_{k,m}\sigma\} \end{aligned} \tag{4.15}$$

Therefore, these two sets of parameters are not identifiable.

To cancel such extra freedom, we regularize the model by adding the following two constraints:

Identification Conditions

$$\begin{cases} \sum_l \delta_l^2 = 1 & (4.16) \\ \min_i s_{l,i} = 0, \forall l & (4.17) \end{cases}$$

The constraints are imposed after the model update in each iteration. Rescaling in this way keeps the model from undesired drifting and scaling.

4.5 Experiments

In this section, we systematically evaluate the techniques presented in this work on both synthetic and real-world datasets. Code and datasets are available at the following repository: https://github.com/dragonxlwang/crowd_thurstonian

[4] This can be verified by comparing $\int_{\mathbf{Z}} \mathrm{P}(\mathbf{D}, \mathbf{Z}|\mathbf{\Theta})$ using Equation (4.6) for $\mathbf{\Theta} = \mathbf{\Theta_1}$ and $\mathbf{\Theta} = \mathbf{\Theta_2}$.

4.5.1 Simulated Study

Datasets

In order to test the effectiveness of TPP under various scenarios, we generate synthetic datasets with the following parameter settings.

The ground truth relevance scores of a list of documents $\{s_{l,i}\}_{i=1,2,...}$ for query q_l are generated from a uniform distribution $\mathcal{U}[0,1]$. Two different lengths are investigated: 5 (DOC5) and 30 (DOC30). Query difficulty δ_l^2 is generated from a uniform distribution $\mathcal{U}[0,0.1]$. To characterize the variable quality of answers given by crowd workers, we assume that worker t_k's expertise and truthfulness $\tau_{k,m}$ on domain m falls into one of the following categories:

- *Expert*: $\tau_{k,m} = 10$
- *Average*: $\tau_{k,m} = 5$
- *Spammer*: $\tau_{k,m} = 1$
- *Malicious*: $\tau_{k,m} = -10$

Three demographic groups are formed by changing the distributions over these four categories. Let p denote the categorical distribution over [*expert, average, spammer, malicious*]:

- DEMO1: $p = [0.2, 0.6, 0.1, 0.1]$. This group represents the most common case where average workers are dominant.
- DEMO2: $p = [0.2, 0.4, 0.3, 0.1]$. This group has a large proportion of spammers that can hurt the annotation quality.
- DEMO3: $p = [0.2, 0.4, 0.1, 0.3]$. The pairwise preferences given by this group can be overwhelmingly misleading due to the presence of too many malicious workers.

In order to simulate the incompleteness of annotations, which in real world often depends on factors such as time and budget constraints, we introduce a variable, *sparsity ratio* (SR), to control the probability that a pair of documents is judged by a worker. For example, if there are a list of 30 documents, and SR $= 0.05$, each worker will judge $\frac{30\times(30-1)}{2} \times 0.05 = 21.75$ randomly selected pairs.

Finally, the following 8 datasets are generated. Each of them contains 10 workers, 10 query domains and 100 queries: DOC5SR1.0DEMO1, DOC5SR0.5DEMO1, DOC5SR0.5DEMO2, DOC5SR0.5DEMO3, DOC30SR0.1DEMO1, DOC30SR0.05DEMO1, DOC30SR0.05DEMO2 and DOC30SR0.05DEMO3.

Table 4.2: Crowd Pairwise Preferences Binding Performance (Kendall's tau Distance)

Dataset	TPP	TPPUNIDOM	TPPUNIEXP	TPPUNIDIFF	CROWDBT
DOC5SR1.0DEMO1	**0.386** ±0.031	0.414 ±0.037	0.466 ±0.023	0.402 ±0.046	0.468 ±0.047
DOC5SR0.5DEMO1	**0.574** ±0.067	0.728 ±0.066	0.846 ±0.080	0.628 ±0.069	0.856 ±0.028
DOC5SC0.5DEMO2	**0.734** ±0.021	0.852 ±0.033	0.940 ±0.037	0.754 ±0.047	0.960 ±0.041
DOC5SR0.5DEMO3	1.592 ±0.237	1.760 ±0.077	2.550 ±0.029	**1.540** ±0.288	2.990 ±0.060
DOC30SR0.1DEMO1	**22.442** ±1.238	25.636 ±0.302	29.204 ±0.291	26.866 ±0.456	24.420 ±0.906
DOC30SR0.05DEMO1	**40.640** ±0.926	45.498 ±0.408	45.636 ±0.178	47.258 ±0.959	48.820 ±2.161
DOC30SR0.05DEMO2	**61.818** ±2.713	70.548 ±0.821	81.782 ±0.145	66.488 ±2.026	104.500 ±2.469
DOC30SR0.05DEMO3	**129.156** ±1.892	139.154 ±0.243	142.496 ±0.587	135.04 ±1.864	153.390 ±1.031

Baselines

We compare the performance of TPP against the following four baselines:

- TPPUNIDOM: TPP without modeling query domains, i.e., all queries are treated as from one single domain.
- TPPUNIEXP: TPP without modeling the domain expertise/truthfulness of workers, i.e., all workers have the same expertise and truthfulness for a given query domain: $\tau_{k_1,m} = \tau_{k_2,m} = \tau_m, \forall k_1, k_2$.
- TPPUNIDIFF: TPP with identical query difficulty, i.e., all queries are equally difficult: $\delta_l^2 = 1/Q,\ \forall l$ with some constant Q.
- CROWDBT: CROWDBT (Chen et al., 2013) is proposed to infer the ground truth scores out of pairwise preferences, which extends the Bradley-Terry model by taking worker accuracy into consideration. Specifically, a "worker-independent" pairwise preference between d_{i_1} and d_{i_2} for q_l is drawn from a Bernoulli distribution. The probability of $d_{i_1} \succ d_{i_2}$ is computed by the Sigmoid function:

$$\sigma(s_{l,i_1} - s_{l,i_2}) = \Big(1 + \exp\big(-(s_{l,i_1} - s_{l,i_2})\big)\Big)^{-1}$$

  Once the pairwise preference is drawn, each worker has a certain probability (accuracy) to report it truthfully or "flip" it. Compared with TPP, CROWDBT lacks the mechanism to model multiple query domains, thus incapable to characterize workers' domain-dependent expertise and truthfulness. Furthermore, it simplifies the generation of inconsistent annotations as solely a result from worker accuracy.

Performance Studies

We test all the methods on synthetic datasets under various parameter settings, and report Kendall's tau distance (Kendall, 1938) between the inferred optimal ranking and the ground truth ranking. Kendall's tau distance is often used to measure the dissimilarity between two ranked lists (Klementiev et al., 2008), which is computed as the number of discordant pairs of the two ranked lists. A pair of documents is discordant if their relative order is reversed in the two rankings. For example, suppose two ranked lists of length 5 are $d_1 \succ d_2 \succ d_3 \succ d_4 \succ d_5$ and $d_3 \succ d_4 \succ d_1 \succ d_2 \succ d_5$. There are in total $\frac{5(5-1)}{2} = 10$ pairs and 4 of them are discordant: $\{d_1, d_3\}$, $\{d_1, d_4\}$, $\{d_2, d_3\}$, $\{d_2, d_4\}$, thus the Kendall's tau distance is 4. A small Kendall's tau distance indicates good performance. We run each method on every dataset 5 times and report the mean and standard deviation in Table 4.2.

Overall Performance. TPP outperforms all other methods in general (the only exception is on DOC5SR0.5DEMO3, where TPPUNIDIFF gives the best result with a small margin). Among the three variants of TPP, TPPUNIEXP has the worst performance in recovering the ground truth rankings. This justifies the importance of modeling workers' domain expertise and truthfulness. Compared with CROWDBT, TPP consistently behaves significantly better, implying that the assumed generative process provides more flexibility in modeling and better explains the generation of inconsistent annotations.

Performance on Different Demographic Groups. Spammers and malicious workers have negative effects on all the methods. The decrease in performance due to malicious workers is more striking than that due to spammers. Nevertheless, the proposed TPP is more robust in resisting the attack from malicious workers than the baselines. Specifically, we observe that the Kendall's tau has increased by 88.516 for TPP when changing the dataset from DOC30SR0.5DEMO1 to DOC30R0.5DEMO3 [5], while this number is 93.656 for TPPUNIDOM, 96.860 for TPPUNIEXP, and 104.57 for CROWDBT. This demonstrates that TPP does a better job in recognizing adversarial workers.

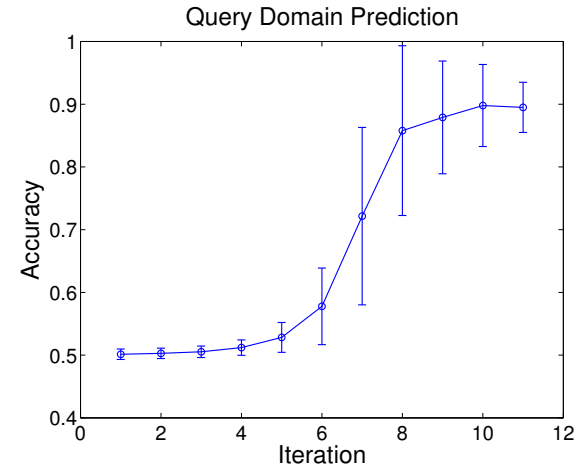
Performance w.r.t. *Sparsity Ratio.* Sparser annotations provide less evidence to infer the ground truth rankings. It is observed that the best performance on DOC5SR0.5DEMO1 (0.574) is still much higher (and thus worse) than the worst performance on DOC5SR1.0DEMO1 (0.468). Similar observations are obtained on the DOC30 datasets.

Query Domain Prediction

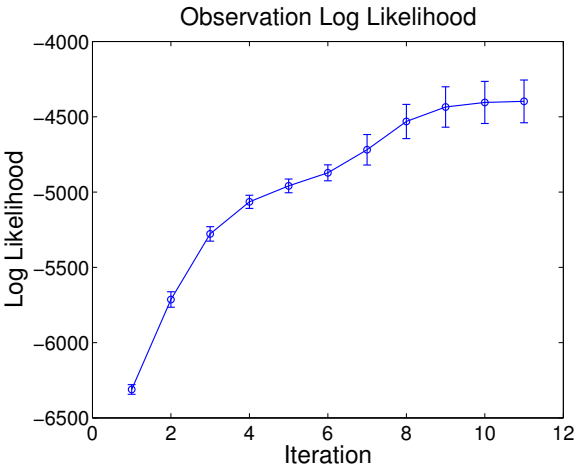
We investigate the capability of TPP in distinguishing between queries from different domains.

We use the setting of DOC5SR0.5DEMO1 and generate the pairwise preferences with only two domains evenly distributed among 100 queries, for the ease of illustration. We run TPP for 10 times and plot the prediction accuracy and the log likelihood. As shown in Figure 4.4, the algorithm starts from random guess with accuracy around 0.5, and converges to an accuracy around 0.895 in less than 10 iterations, implying that TPP is able to learn query domains

[5] The maximal Kendall's tau distance for DOC30 is $\frac{30(30-1)}{2} = 435$.



(a) Accuracy



(b) Log Likelihood

Figure 4.4: Domain Prediction Accuracy and Model Log Likelihood with Standard Deviations

effectively and efficiently.

More Workers but Sparser Annotation

In practice, when time is the constraining factor, it is plausible to employ a large number of crowd workers and each worker labels only a few pairs. However, the situation of *"More Workers but Sparser Annotation"* can potentially lead to a critical limitation for TPP. On one hand, the number of parameters $\{\tau_{k,m}\}$ grows with the number of workers. On the other hand, the amount of data to estimate each $\tau_{k,m}$ decreases.

To evaluate the performance in such scenarios, we create another four datasets under the setting of DOC30DEMO1 with more annotators (ANNO100 of 100 annotators and ANNO200 of 200 annotators) and lower sparsity ratios (SR0.01 and SR0.02).

As shown Table 4.3, the performance of TPP becomes worse with "More Workers but Sparser Annotation" as

Table 4.3: TPP Performance with More Workers but Sparser Annotation (Kendall's tau Distance)

Dataset	Kendall's tau	
ANNO100SR0.01	36.208	±0.292
ANNO100SR0.02	24.328	±0.451
ANNO200SR0.01	25.734	±0.394
ANNO200SR0.02	16.290	±0.435

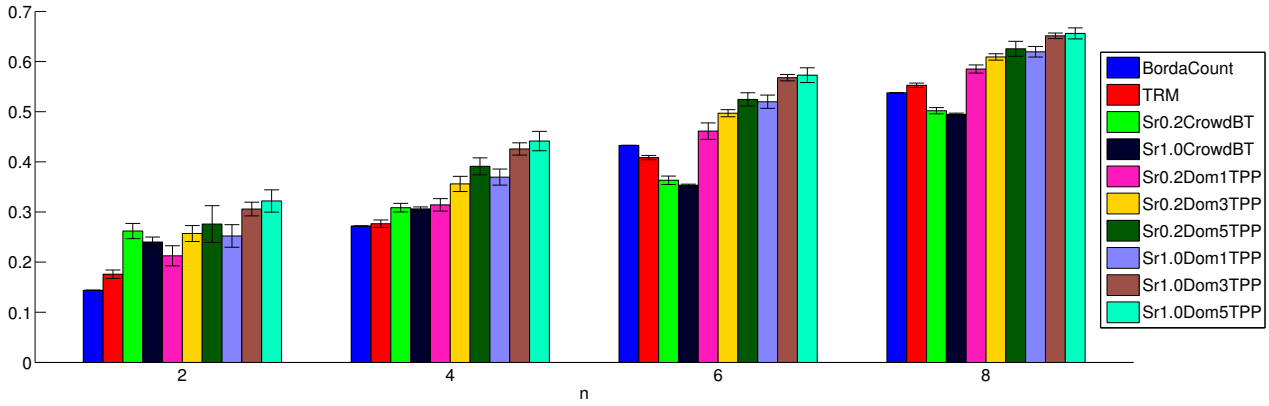


Figure 4.5: NDCG@n evaluated on MQ2008-agg Dataset

Kendall's tau increases from 22.442 (DOC30SR0.1DEMO1) to 36.208 (ANNO100SR0.01). This is anticipated because the two datasets have the same amount of pairwise judgements but ANNO100SR0.01 involves more workers and has sparser annotations. However, ANNO100SR0.01 drastically reduces the time cost and may take only a tenth of the time that DOC30SR0.1DEMO1 takes. In fact, by doubling the number of workers to 200 or doubling the sparsity ratio to 0.02, comparable performance can be achieved with DOC30SR0.1DEMO1. With an even more aggressive setting ANNO200SR0.02 (20 times the number of workers and five times sparser annotations), the performance further improves. Therefore we conclude that the performance of TPP is reasonably robust even at the situation of "more workers and sparser annotation."

Malicious Worker Detection

Identifying malicious workers is a difficult task since the number of malicious workers is usually small so that the classification is highly imbalanced. We assess the performance of malicious worker detection by plotting the averaged Receiver Operating Characteristic (R.O.C.) curves in Figure 4.6. In the experiment, with 100 workers from DEMO1 and SR $= 0.01$, TPP performs well with AUC $= 0.837$ (Area Under the Curve). When the annotation is denser (ANNO100SR0.02), AUC improves remarkably (0.924). However, with 200 workers (DEMO1), the difference of AUC between SR = 0.01 and SR = 0.02 is not significant. This can be explained by the fact that malicious workers

are easier to identify in a larger group, even with sparser annotations.

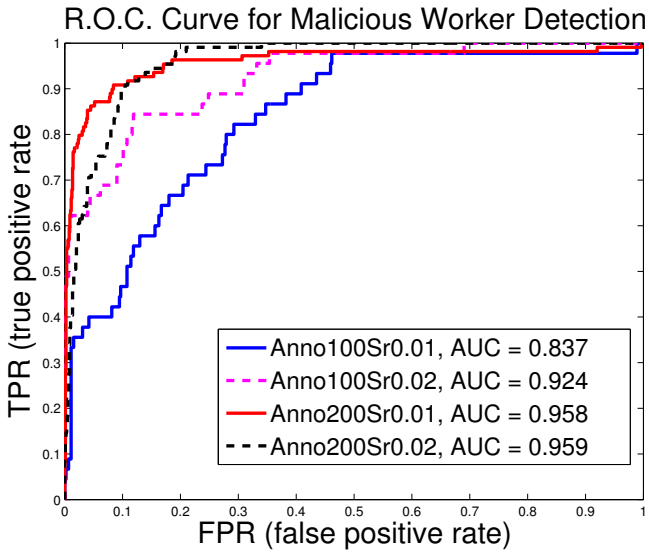


Figure 4.6: R.O.C. Curve for Malicious Worker Detection

4.5.2 Experiments on Real-World Data

To validate our proposed strategy of binding pairwise preferences into rankings, we utilize a real-world benchmark MQ2008-agg (part of LETOR 4.0[6]) which is originally devised for the rank aggregation (meta-ranking) task. The MQ2008-agg dataset consists of ranked lists from 25 retrieval systems (workers). Each document is labeled as *highly relevant* (2), *relevant* (1) or *irrelevant* (0). For rank aggregation algorithms (TRM and BordaCount), ranked lists generated from each retrieval system are taken as input to infer the true ranked list for each query. The pairwise preference binding algorithms (TPP and CROWDBT), on the other hand, estimate the true ranking out of the pairwise judgements from each retrieval system ("worker"). The pairwise judgements are randomly sampled with a sparsity ratio SR. In the experiment, we use sparsity ratios SR = 1.0 (all pairwise judgements are observed) and SR = 0.2. We evaluate TPP with 1, 3 and 5 domains. The performance is compared against both the pairwise preference binding algorithm CROWDBT, and the rank aggregation algorithms BordaCount (Aslam and Montague, 2001) and TRM (see Section 4.2 and Appendix A.2). In particular, Bordacount is a simple yet robust algorithm which is essentially a ranking version of *majority voting*. It infers the true ranking by averaging the rank positions from each worker. The performance is measured by NDCG (Normalized Discounted Cumulative Gain) (Järvelin and Kekäläinen, 2000). We use NDCG@n where $n = 2, 4, 6, 8$.

The results are presented in Figure 4.5. In general, similar performances are observed for the two rank aggregation

[6]http://research.microsoft.com/en-us/um/beijing/projects/letor

algorithms with TRM slightly outperforming Bordacount. With SR1.0, TPP and CROWDBT have the same amount of information from observations as the rank aggregation counterparts. However, SR1.0CROWDBT performs better than TRM and Bordacount only at NDCG@2 and NDCG@4, while it gets worse at NDCG@6 and NDCG@8. In contrast, TPP consistently outperforms all the baselines, with better performance achieved if more domains are incorporated.

When the available annotations become sparser (SR0.2), the performance of both TPP and CROWDBT become worse: NDCGs decrease across different settings. However, TPP still significantly outperforms CROWDBT even with a single domain. In addition, it also outperforms TRM and Bordacount although the annotation is incomplete. This is because that the flexible generative process of TPP properly resolves the inconsistency from multiple sources.

4.6 Related Work

Early research of crowdsourcing can be dated back to the study of *integration of labels from multiple annotators* for image classification (Smyth et al., 1995). Later on, studies including (Whitehill et al., 2009; Yan et al., 2010) began focusing on explicitly modeling annotator quality such as expertise, truthfulness in crowdsourcing settings. The dual tasks of inferring ground truth labels as well as worker quality have been investigated in some recent studies (Welinder et al., 2010; Whitehill et al., 2009; Yan et al., 2010), including this work.

Previous research mainly focused on simple tasks (classification, regression, etc.) while we tackle complex labeling problem such as ranking. In this direction, (Steyvers et al., 2009) reconstructs the order of facts from individual worker annotated *whole ranked lists* with the Thurstonian Ranking Model (TRM) (Thurstone, 1927) and the Mallows model (Mallows, 1957), which features a distance-based distribution of rankings (permutations) using Kendall's tau. Other studies on "Rank Aggregation" are also related to this work, including (Klementiev et al., 2008, 2009). They adapt the Mallows model for inferring ground truth rankings as well as the quality of ranking algorithms. However, the above approaches do not fit well for information retrieval and web search tasks as it is not practical for annotators to label the whole ranked lists. This motivates us to investigate binding pairwise preferences from crowd workers into rankings.

There is one recent study (Chen et al., 2013) that adopts a similar philosophy, which extends the Bradley-Terry model, a pairwise special case of the Plackett-Luce model (Luce, 2005; Plackett, 1975). Nevertheless, their model (CROWDBT) lacks the mechanism to model multiple query domains, thus incapable to characterize workers' domain-dependent expertise and truthfulness. CROWDBT does not take query difficulty into account either. Furthermore, unlike TPP, CROWDBT does not model the generation of rankings. Therefore, it is not capable of modeling the annotation inconsistency from multiple sources, which makes it less favorable as demonstrated by the experimental study.

4.7 Conclusions and Future Work

In this chapter, we present a novel generative model called "Thurstonian Pairwise Preference" (TPP) to infer the true ranked list out of a collection of crowdsourced pairwise annotations, which is highly useful in various data mining and information retrieval tasks such as *learning to rank*. TPP resolves the inevitable incompleteness and inconsistency of pairwise judgements, by carefully modeling variable query difficulty and different labeling quality resulting from workers' domain expertise and truthfulness. Experimental results on both synthetic and real-world datasets demonstrate that TPP can effectively bind pairwise preferences of the crowd into rankings and substantially outperforms previously published methods. To further explore the benefit from the inferred ranked lists, it is promising to extend TPP to jointly learn the ranking model of the end application, which we leave for future work.

Chapter 5

Dual-Clustering Maximum Entropy with Application to Classification and Word Embedding

5.1 Introduction

We have already witnessed that PLVM is an excellent tool for modeling data of different types. However, it is less explored whether we can leverage PLVM for scalable and efficient optimization as well. In this chapter, a novel approach, Dual-Clustering Maximum Entropy, is proposed to address the stability problem of Maximum Entropy when there is an extreme large number of items (classes/words) present. The key insight is that latent variables can be investigated to perform model reduction and to facilitate inference. By incorporating the modeling of latent variables, the dual space of the Maximum Entropy problem is explored and a K-means like clustering is conducted over the simplex space. We demonstrate that leveraging PLVM leads to an efficient algorithm, the complexity of which does not depend on the number of items.

Maximum Entropy (ME), also known by a variety of other names, including log-linear, Gibbs, exponential, softmax and multinomial logistic regression models, is one of the most widely applied machine learning techniques in various fields. As a classification method, ME has seen wide-scale applications in text mining and natural language processing, such as text classification (Nigam et al., 1999), part-of-speech tagging (Ratnaparkhi, 1996) and machine translation (Berger et al., 1996). In neural networks, ME (softmax) is the building block of network architectures to transform a vector of signals into probabilities (Collobert and Weston, 2008), and has been explored to learn neural probabilistic language models (Bengio et al., 2003). In recent literature, a number of word embedding algorithms have been proposed based on ME, including skip-gram, continuous bag-of-words (CBOW) (Mikolov and Dean, 2013; Mikolov et al., 2013) and log-bilinear models (Mnih and Hinton, 2007), among others.

ME establishes a distribution of the exponential form over items (classes/words) (Equation (5.1)). Scalability becomes a crucial challenge when the number of items is large, which occurs nowadays in many real-world problems. For example, in a text classification problem of predicting the publishing venue for research papers, the number of classes can easily exceed thousands on datasets such as ACM digital library[1]; for word embedding, commonly used training corpora, with the English Gigaword[2] as an example, typically have a vocabulary of hundreds of thousands, if

[1] http://dl.acm.org/
[2] https://catalog.ldc.upenn.edu/LDC2011T07

not millions of words.

The main computational difficulty in ME comes from the fact that one has to enumerate all items in order to obtain either the probability of a single item or the corresponding gradient (Mnih and Teh, 2012). Consequently conventional ME optimization techniques such as iterative scaling (Berger et al., 1996; Darroch and Ratcliff, 1972) and gradient-based algorithms (Gao et al., 2007; Tsuruoka et al., 2009) are very slow to train with large numbers of items. In practice, sampling-based methods (Bengio and Senécal, 2008; Gutmann and Hyvärinen, 2010; Mnih and Teh, 2012) are often adopted since the complexity does not hinge on the number of items. However, one drawback they possess is the inevitable sampling variance. Furthermore, only the model parameters associated with the sampled items get updated at each training instance, while the majority of the model is left unchanged, which leads to inefficient learning.

To achieve learning efficiency with affordable computational cost, we propose a Dual-Clustering Maximum Entropy (DCME) approach. It optimizes ME in a primal-dual fashion, where the multinomial *dual distribution* for each instance is exploited. The key step of DCME is to cluster the dual distributions and to approximate each of them by the corresponding *cluster center*. The dual clustering proceeds by alternating between an online update of each instance's cluster assignment and an offline calculation[3] of the cluster centers. This gives rise to an efficient updating scheme which splits the computation of the model subgradient into an *online* part and an *offline* part. Our proposed DCME enjoys two desirable properties: (1) The model parameters associated with *all* items are updated at *each* training instance, which ensures learning efficiency; and (2) The computational cost per instance scales as the product of the feature/word vector dimensionality[4] and the number of clusters, which yields fast training speed.

The rest of the chapter is organized as follows: Section 5.2 reviews the Maximum Entropy and existing approaches for learning with large numbers of items. The proposed DCME is presented in Section 5.3 with the derivation and complexity analysis. The overall algorithmic procedure is summarized in Section 5.4 where the theoretical advantages of DCME are also discussed. Experimental studies on text classification and word embedding are reported in Section 5.5, followed by conclusions in Section 5.6.

5.2 Background

In this section, we first provide a brief review on Maximum Entropy (ME) framework, together with a short account for the works on optimization of the ME. Then we discuss current research development for extreme classification, i.e., classification with a larger item number.

[3] In this chapter, the term "offline" is equivalent to "batch computation".

[4] To be precise, by taking advantage of sparsity, the complexity depends only on the number of non-zero elements instead of the dimensionality of the vector.

5.2.1 Maximum Entropy Framework

The general formulation of ME is simple. For a data instance t, ME establishes a distribution over N items:

$$\mathrm{P}_t(i;\boldsymbol{\Theta}) = \frac{\exp(f_t(i;\boldsymbol{\Theta}))}{\sum_{j=1}^{N}\exp(f_t(j;\boldsymbol{\Theta}))}, \quad i = 1,\ldots,N. \tag{5.1}$$

where $f_t(i;\boldsymbol{\Theta})$ is the scoring function with model parameters $\boldsymbol{\Theta}$, which quantifies the affinity between instance t and item i[5]. In this chapter, we investigate ME in two settings, namely, multi-class classification and word embedding.

For N-class classification, the dataset $\mathcal{D}$ consists of a collection of instances $\{(\mathbf{x}_t, i_t)\}$ with $\mathbf{x}_t$ being a D-dimensional feature vector and i_t a label chosen from items $1,\ldots,N$. The model $\mathbf{W} = [\mathbf{w}_1,\ldots,\mathbf{w}_N]$ is a $D \times N$ matrix which specifies the scoring function as:

Classification: $$f_t(j;\boldsymbol{\Theta}) = f_t(j;\mathbf{W}) = \mathbf{w}_j^T\mathbf{x}_t \tag{5.2}$$

In the word embedding setting, we focus our discussion on the continuous bag-of-words algorithm (CBOW) (Mikolov et al., 2013), but the analysis easily extends to other models. As a language modeling technique, it predicts the target word from a vocabulary of size N given its surrounding context. The t-th training instance contains a stream of words $w_{t,-c}, w_{t,-(c-1)},\ldots,w_{t,0},\ldots,w_{t,c-1},w_{t,c}$ with the target word $i_t = w_{t,0}$. CBOW calculates the compatibility between the j-th word in the vocabulary and the context as:

Embedding:
$$f_t(j;\boldsymbol{\Theta}) = f_t(j;\mathbf{V},\mathbf{H}) = f(\mathbf{v}_j,\bar{\boldsymbol{h}}_t) = \mathbf{v}_j^T\bar{\boldsymbol{h}}_t$$
$$\text{where} \quad \bar{\boldsymbol{h}}_t = \frac{1}{2c}\sum_{-c\le p\le c, p\neq 0}\mathbf{h}_{w_{t,p}} \tag{5.3}$$

The model parameters $\mathbf{V} = [\mathbf{v}_1,\ldots,\mathbf{v}_N]$ and $\mathbf{H} = [\mathbf{h}_1,\ldots,\mathbf{h}_N]$ are two $D \times N$ matrices of the "input" and "output" vector representations of words, respectively.

5.2.2 Optimization of ME

Various algorithms for ME have been studied in the literature. They approach the optimization by solving either the primal or the dual problem. The primal form maximizes the log-likelihood of the dataset. Methods of this direction, as surveyed in (Malouf, 2002; Yuan et al., 2012), include iterative scaling algorithms (Berger et al., 1996;

[5] In the context of energy-based models, $-f_t(i;\boldsymbol{\Theta})$ is often referred as the energy function (Bengio et al., 2003).

Darroch and Ratcliff, 1972), coordinate descent (Huang et al., 2010), stochastic gradient descent (Tsuruoka et al., 2009) and Quasi-Newton method (Gao et al., 2007), just to name a few. Their training complexity per instance is $\mathcal{O}(DN)$. This is a consequence of having to enumerate *all* items when computing the probability of a single item or the corresponding gradient. On the other hand, another line of research tackles the problem by maximizing the entropy of *dual distributions*. Constraint optimization techniques, such as exponentiated gradient (Collins et al., 2008) and dual coordinate descent (Yu et al., 2011), are investigated. Since the dimensionality of dual distributions is in fact the same as the number of items, their training complexity is still linear in N. Consequently, all these algorithms are impractical with large numbers of items due to the prohibitively expensive computational cost.

5.2.3 Learning with Large Item Number

Scaling algorithms for learning when the number of items N is large have become a recent research direction with focus on maintaining the training complexity sublinear in N. Among them, hierarchical approaches explore a taxonomy (of items) and convert the problem into a series of binary predictions along the tree branches, which potentially reduces the complexity from $\mathcal{O}(N)$ to $\mathcal{O}(\log N)$. Though efforts have been made in large multi-class (extreme) classification (Choromanska et al., 2013; Choromanska and Langford, 2015) and word embedding (Mikolov et al., 2013; Mnih and Hinton, 2009; Morin and Bengio, 2005), finding balanced tree structures that provide an effective partition of items is difficult by itself, and thus their use is limited in practice. Another work of extreme classification, (Yen et al., 2016), has developed a fast active set algorithm for max-margin classifiers by exploiting the sparsity of feature vectors. The training speed-up, nevertheless, is generally insignificant for dense data representations such as word embeddings. To the best of our knowledge, the most effective approaches for training ME models with a large N are sampling-based methods, for instance (Bengio and Senécal, 2008), offering a trade-off between speed and precision. In addition, as pointed out by (Mnih and Teh, 2012), noise-contrastive estimation (NCE) (Gutmann and Hyvärinen, 2010) is regarded as the state-of-the-art sampling algorithm which employs the idea of "learning by comparison": It reduces the N-item ME problem to a binary classification between samples from the training data and "noise" from the proposal distribution, and is guaranteed to converge to the solution of ME. Yet in practice, a slightly simpler variant, negative sampling (NS) (Mikolov and Dean, 2013), is proposed to train CBOW and skip-gram though mathematically it does not solve ME. However, one drawback is that algorithms of this kind inevitably suffer from sampling variance. More crucially, the computational efficiency is gained at the expense of only updating the model parameters associated with the sampled items, while the due change of the rest is discarded. Learning efficiency is therefore sacrificed.

5.3 Dual-Clustering Maximum Entropy

In this section, we present a Dual-Clustering Maximum Entropy (DCME) approach which has two advantages regarding learning and computational efficiency: (1) The model parameters associated with *all* items are updated at *each* training instance; and (2) The time complexity is independent of N.

5.3.1 Primal-dual ME

Different from existing approaches, DCME solves the ME problem in a primal-dual fashion. Suppose that the dataset $\mathcal{D}$ has M instances and N items where the t-th instance selects the i_t-th item. We start the derivation from the primal ME formulation which maximizes the log-likelihood:

$$\begin{aligned}\sum_{t=1}^{M} \log(\mathrm{P}_t(i_t;\boldsymbol{\Theta})) &= \sum_{t=1}^{M} \big(f_t(i_t;\boldsymbol{\Theta}) - \log \sum_{j=1}^{N} \exp f_t(j;\boldsymbol{\Theta})\big) \\ &= \sum_{t=1}^{M} \big(f_t(i_t;\boldsymbol{\Theta}) - A_t(\boldsymbol{\Theta})\big) \end{aligned} \tag{5.4}$$

where $A_t(\boldsymbol{\Theta})$ is referred to as the log-partition function and its conjugate dual is revealed by the following lemma (Hiriart-Urruty and Lemarechal, 1993; Wainwright and Jordan, 2008):

Lemma 5.3.0.1. *Assume* $\mathrm{P}(i;\mathbf{s}) = \exp(s_i)/\sum_{j=1}^{N}\exp(s_j)$ *and* $A(\mathbf{s}) = \log\sum_{j=1}^{N}\exp(s_j)$*, the conjugate duality between the log-partition function and negative entropy states:*

$$\begin{aligned} A(\mathbf{s}) &= \max_{\boldsymbol{\mu}\in\Delta_N} \{\sum_{j=1}^{N}\mu_j s_j - \sum_{j=1}^{N}\mu_j\log\mu_j\} \\ &= \max_{\boldsymbol{\mu}\in\Delta_N} \{\mathbf{E}_{\boldsymbol{\mu}}[s_j] + \mathbf{H}(\boldsymbol{\mu})\} \end{aligned} \tag{5.5}$$

where the simplex set $\Delta_N = \{\mathbf{p}\in\mathbb{R}^N : p_j \geq 0, \sum_{j=1}^{N} p_j = 1\}$

and the maximizer is attained at:

$$\mu_j^* = \mathrm{P}(j;\mathbf{s}), \quad 1 \leq j \leq N \tag{5.6}$$

Proof. We use the following equivalence:

$$\begin{aligned}\mathbf{E}_{\boldsymbol{\mu}}[s_j] + \mathbf{H}(\boldsymbol{\mu}) &= -\sum_{j=1}^{N} \mu_j \log \frac{\mu_j}{\mathrm{P}(j;\mathbf{s})} + \log \sum_{j=1}^{N} \exp(s_j) \\ &= -D_{KL}(\boldsymbol{\mu}||\mathrm{P}) + A(\mathbf{s})\end{aligned}$$

where $D_{KL}(\boldsymbol{\mu}||\mathrm{P})$ is the Kullback-Leibler (KL) divergence and note $D_{KL}(\boldsymbol{\mu}||P) \geq 0$ and $D_{KL}(P||P) = 0$. It follows that $\boldsymbol{\mu}^* = \underset{\boldsymbol{\mu}\in\Delta_N}{\arg\min}\, D_{KL}(\boldsymbol{\mu}||P) = P$. □

In view of Lemma 5.3.0.1, we arrive at the primal-dual form of ME:

$$\max_{\boldsymbol{\Theta}} \min_{\substack{\boldsymbol{\mu}_t\in\Delta_N \\ 1\leq t\leq M}} \sum_{t=1}^{M} \big(f_t(i_t;\boldsymbol{\Theta}) - \mathbf{E}_{\boldsymbol{\mu}_t}[f_t(j;\boldsymbol{\Theta})] - \mathbf{H}(\boldsymbol{\mu}_t)\big) \tag{5.7}$$

where $\boldsymbol{\mu}_t$ is the *dual distribution* for instance t.

5.3.2 Dual Distribution Clustering

Lemma 5.3.0.1 implies that $\boldsymbol{\mu}_t^*$ is determined by $f_t(j;\boldsymbol{\Theta})$. In less mathematical terms, similar instances choose similar items (in probabilities). As real-world data instances generally possess a clustering structure instead of being randomly distributed, it is expected that dual distributions also form clusters. For the text classification example of venue prediction, if papers are grouped by topics, those in the same group should have similar chance of getting published at a particular venue; For learning word embedding, we anticipate contexts of similar semantics yield target word distributions that can be clustered together.

It is worth exploring the cluster structure of dual distributions to reduce complexity. DCME rests on the idea of "approximation by clustering": By clustering the dual distributions into K groups, each $\boldsymbol{\mu}_t$ is assigned to a cluster $c_t \in \{1,\dots K\}$, and is then approximated by the corresponding *cluster center* $\boldsymbol{\alpha}_{c_t} \in \Delta_N$ which best represents the group. The optimization problem of DCME can thus be formulated as:

$$\text{DCME:} \qquad \max_{\boldsymbol{\Theta}} \min_{\substack{\boldsymbol{\alpha}_k\in\Delta_N \\ 1\leq k\leq K}} \min_{\substack{1\leq c_t\leq K \\ 1\leq t\leq M}} \sum_{t=1}^{M} Q_t(\boldsymbol{\alpha}_{c_t};\boldsymbol{\Theta}) \tag{5.8}$$

$$\text{where} \quad Q_t(\boldsymbol{\alpha}_{c_t};\boldsymbol{\Theta}) = f_t(i_t;\boldsymbol{\Theta}) - \mathbf{E}_{\boldsymbol{\alpha}_{c_t}}[f_t(j;\boldsymbol{\Theta})] - \mathbf{H}(\boldsymbol{\alpha}_{c_t})\big)$$

5.3.3 Online-Offline Optimization

We employ Gauss-Seidel coordinate descent to solve Equation (5.8). Three blocks of variables, namely, the model parameters $\boldsymbol{\Theta}$, the cluster centers $\{\boldsymbol{\alpha}_k\}$, and the instances' cluster assignments $\{c_t\}$, are successively updated while keeping others constant. In particular, we devise a hybrid online-offline algorithm which breaks the computational bottleneck and leads to a time complexity that only scales as $\mathcal{O}(DK)$, as opposed to $\mathcal{O}(DN)$ in conventional ME algorithms.

Updating cluster assignments (Online)

DCME approximates $\boldsymbol{\mu}_t$ by $\boldsymbol{\alpha}_{c_t}$, and the cluster assignment c_t is solved by:

$$\arg\min_{1\le k\le K} -\mathbf{E}_{\boldsymbol{\alpha}_k}[f_t(j;\boldsymbol{\Theta})] - \mathbf{H}(\boldsymbol{\alpha}_k) \tag{5.9}$$

However, a naïve computation would cost $\mathcal{O}(DN + KN)$ time. It takes $\mathcal{O}(D)$ to evaluate $f_t(j;\boldsymbol{\Theta})$ for every item $1 \le j \le N$[6]; For each cluster, another $\mathcal{O}(N)$ is required to calculate $\mathbf{E}_{\boldsymbol{\alpha}_k}[f_t(j;\boldsymbol{\Theta})]$ and $\mathbf{H}(\boldsymbol{\alpha}_k)$ by enumeration.

Fortunately, when the scoring function is linear in the feature/context vector, the cost can be reduced to $\mathcal{O}(DK)$ per instance. To see this, from (5.2) and (5.3) we have:

$$\text{Classification:}\quad \mathbf{E}_{\boldsymbol{\alpha}_k}[f_t(j;\mathbf{W})] = (\mathbf{W}\boldsymbol{\alpha}_k)^T\mathbf{x}_t \tag{5.10}$$

$$\text{Embedding:}\quad \mathbf{E}_{\boldsymbol{\alpha}_k}[f_t(j;\mathbf{V},\mathbf{H})] = (\mathbf{V}\boldsymbol{\alpha}_k)^T\bar{\boldsymbol{h}}_t \tag{5.11}$$

The trick we apply here *trades memory for time*: By storing $\mathbf{W}\boldsymbol{\alpha}_k$, $\mathbf{V}\boldsymbol{\alpha}_k$ and $\mathbf{H}(\boldsymbol{\alpha}_k)$ for K clusters in the offline update, it is merely a D-dimensional dot product to calculate Equation (5.10) and Equation (5.11), and therefore the cost to online update c_t by Equation (5.9) is $\mathcal{O}(DK)$.

Updating cluster centers (Offline)

We update the cluster center $\boldsymbol{\alpha}_k$ as well as the cached $\mathbf{W}\boldsymbol{\alpha}_k$, $\mathbf{V}\boldsymbol{\alpha}_k$ and $\mathbf{H}(\boldsymbol{\alpha}_k)$ only in the offline computation. Let $\mathcal{I}_k$ denote the index set of instances in the k-th cluster, $\boldsymbol{\alpha}_k$ satisfies:

$$\arg\min_{\boldsymbol{\alpha}\in\Delta_N} -\mathbf{E}_{\boldsymbol{\alpha}}\left[\frac{1}{|\mathcal{I}_k|}\sum_{t\in\mathcal{I}_k} f_t(j;\boldsymbol{\Theta})\right] - \mathbf{H}(\boldsymbol{\alpha}) \tag{5.12}$$

[6] In word embedding, one can compute the scoring function in $\mathcal{O}(D)$ time. Note that the *asymptotic* complexity of computing $\bar{\boldsymbol{h}}_t$ in *every* sliding windows is $\mathcal{O}(D)$ (independent of window size) with the sum $\sum_{-c\le p\le c}\mathbf{h}_{w_t,p}$ maintained by adding the new word and subtracting the past word.

Invoking Lemma 5.3.0.1 again, Equation (5.12) has the following closed-form solution:

$$\alpha_{k,j} = \frac{1}{Z} \exp\big(\frac{1}{|\mathcal{I}_k|} \sum_{t \in \mathcal{I}_k} f_t(j; \mathbf{\Theta})\big) \tag{5.13}$$

where a normalization term Z is applied to keep $\sum_{j=1}^{N} \alpha_{k,j} = 1$. By the linearity of $f_t(j; \mathbf{\Theta})$, we express Equation (5.13) as:

$$\text{Classification:} \quad \alpha_{k,j} = \frac{1}{Z} \exp\Big(\mathbf{w}_j^T \frac{1}{|\mathcal{I}_k|} \sum_{t \in \mathcal{I}_k} \mathbf{x}_t\Big) \tag{5.14}$$

$$\text{Embedding:} \quad \alpha_{k,j} = \frac{1}{Z} \exp\Big(\mathbf{v}_j^T \frac{1}{|\mathcal{I}_k|} \sum_{t \in \mathcal{I}_k} \bar{\boldsymbol{h}}_t\Big) \tag{5.15}$$

which computes $\boldsymbol{\alpha}_k$ with $\mathcal{O}(D|\mathcal{I}_k| + DN)$ cost. In addition, it takes $\mathcal{O}(DN)$ and $\mathcal{O}(N)$ to update $\mathbf{W}\boldsymbol{\alpha}_k$, $\mathbf{V}\boldsymbol{\alpha}_k$ and $\mathbf{H}(\boldsymbol{\alpha}_k)$, respectively. By choosing the interval between consecutive offline updates such that $|\mathcal{I}_k| = \beta N$ for a constant β (1 for example), we obtain an average time complexity of $\mathcal{O}(D)$ per instance.

Updating model parameters (Online/Offline)

To optimize $\mathbf{\Theta}$ with subgradient descent:

$$\text{Classification:} \quad \frac{\partial Q_t}{\partial \mathbf{w}_j} = \underbrace{\mathbb{1}[i_t = j]\, \mathbf{x}_t}_{(a)} + \underbrace{(-\alpha_{c_t,j} \mathbf{x}_t)}_{(b)} \tag{5.16}$$

$$\text{Embedding:} \quad \frac{\partial Q_t}{\partial \mathbf{v}_j} = \overbrace{\mathbb{1}[i_t = j]\, \bar{\boldsymbol{h}}_t} + \overbrace{(-\alpha_{c_t,j} \bar{\boldsymbol{h}}_t)} \tag{5.17}$$

$$\frac{\partial Q_t}{\partial \mathbf{h}_{w_p^{(t)}}} = \frac{1}{2c}(\mathbf{v}_{i_t} - \mathbf{V}\boldsymbol{\alpha}_{c_t}) \tag{5.18}$$

$$\text{for all } -c \le p \le c,\ p \neq 0$$

where $\mathbb{1}[i_t = j]$ is the indicator function which evaluates to 1 when $i_t = j$ and 0 otherwise. In the following, we devise a hybrid *online-offline algorithm* which has an average expense of $\mathcal{O}(D)$ time per instance.

First, Term (a) in Equation (5.16) and Equation (5.17) only changes the model parameters associated with the correct item i_t, namely $\mathbf{w}_{i_t}$ and $\mathbf{v}_{i_t}$, and can be updated online in $\mathcal{O}(D)$ time. Similarly, for word embedding Equation (5.18), $\partial Q_t / \partial \mathbf{H}$ can also be updated online with $\mathcal{O}(D)$ cost by keeping track of the sum of $\partial Q_t / \partial \mathbf{h}_w$ for all overlapping

instances using a sliding window technique.

Second, Term (b) in Equation (5.16) and Equation (5.17) changes the model parameters of all N items. We make two crucial observations here: (1) Term (b) of different items share the *same* direction $-\mathbf{x}_t$ (or $-\bar{\boldsymbol{h}}_t$); and (2) The scale vector $\boldsymbol{\alpha}_{c_t}$ *only* depends on the cluster assignment c_t, but not the individual instance t. Thus it is logical to perform offline update of Term (b). The computation is postponed until $|\mathcal{I}_{c_t}|$ is large enough, and then Term (b) is calculated for all items $1 \leq j \leq N$ and instances $t \in \mathcal{I}_{c_t}$. Such "lazy" computation yields a total cost of $\mathcal{O}(D|\mathcal{I}_{c_t}| + DN)$, and we achieve an average $\mathcal{O}(D)$ expense per instance if we wait until $|\mathcal{I}_{c_t}| \geq \beta N$.

Tuning online/offline computation

The overall complexity per instance is $\mathcal{O}(DK)$ time, which is appealing as it does not hinge on N. Nevertheless, an inherent limitation in learning is the delay of computing Term (b) until $|\mathcal{I}_k| \geq \beta N$ in Equation (5.16) and Equation (5.17) , especially for items with large values $\alpha_{k,j}$. A heuristic improvement we find effective in practice tunes the computation between online and offline updates. By sorting items (using a heap) with decreasing $\alpha_{k,j}$, Term (b) of the top Q items are updated online while the others are updated offline. The resulting average cost per instance becomes $\mathcal{O}(DK + DQ + \log Q)$. Computational efficiency is preferred with a small Q while the priority shifts to learning efficiency with a large Q.

5.4 DCME Algorithm

We have already presented the DCME algorithm in previous section. In the following, we first give an overview about the DCME algorithm, and then illustrate its connections with K-means algorithm and the Dual ME respectively.

5.4.1 Overall Procedures

We summarize the learning procedure of DCME in Algorithm 2. DCME assigns each training instance t to a dual cluster c_t and performs the online model update. Once the size of a dual cluster reaches βN, an offline model update as well as the update of the dual cluster center are applied. Although the algorithm has a similar complexity as the sampling-based approaches such as noise contrastive estimation (NCE) and negative sampling (NS), DCME allows the *entire* model to learn from *every* training instance. In other words, the model parameters associated with all items get updated when a new training instance arrives, which yields superior performance over existing methods, as will be shown in the experimental study.

Algorithm 2: DCME algorithm

Input: M instances, a constant β, cluster number K, and top item number Q
Output: Model $\boldsymbol{\Theta}$

Initialize K clusters $\{\boldsymbol{\alpha}_k\}$;
while $\boldsymbol{\Theta}$ *is not optimal* **do**
- Select an index t from $\{1, \ldots, M\}$
- Find the cluster assignment c_t by (5.9)
- Perform online update of Term (a) (and Term (b) of the top Q items) in (5.16) (or (5.17)). For embedding, also update H by (5.18).
- Add t to $\mathcal{I}_{c_t}$;
  if $|\mathcal{I}_{c_t}| \geq \beta N$
  - Perform offline update of Term (b) in (5.16) (or (5.17)).
  - Update cluster center α_{c_t} and empty $\mathcal{I}_{c_t}$

5.4.2 Connection with K-means

So far, readers might have already been aware of the resemblance between the dual distribution clustering and the K-means algorithm. The following theorem formally proves their connection:

Theorem 5.4.1. *The dual distribution clustering in DCME is a generalized K-means algorithm using KL-divergence as the distance measurement in the simplex. Moreover, it converges as fast as K-means.*

Proof. Using Lemma 5.3.0.1, the dual clustering satisfies:

$$\min_{\substack{\boldsymbol{\alpha}_k \in \Delta_N, 1 \leq k \leq K \\ 1 \leq c_t \leq K, 1 \leq t \leq M}} \sum_{t=1}^{M} D_{KL}(\boldsymbol{\alpha}_{c_t} || P_t) \tag{5.19}$$

which minimizes the *within-cluster KL-divergence* between $\boldsymbol{\alpha}_{c_t}$ and P_t. It is the same minimization objective as K-means except that DCME measures the distance in the simplex space with KL-divergence[7]. To illustrate this, notice that the dual clustering proceeds by alternating between the following two steps (See Figure 5.1):

- Update $c_t = \arg\min_k D_{KL}(\boldsymbol{\alpha}_k || P_t)$, and t is assigned to the cluster whose center is nearest to P_t by KL-divergence.
- Update $\boldsymbol{\alpha}_k = \arg\min_{\boldsymbol{\alpha}} \sum_{t \in \mathcal{I}_k} D_{KL}(\boldsymbol{\alpha} || P_t)$ where the cluster center is found as the point in the simplex with the least within-cluster distance.

[7] Technically, KL-divergence is not a true metric of distance.

General convergence results for the subgradient methods can be applied. Specifically, the above two-step algorithm converges to the local minimum of the problem (5.19) as fast as the K-means algorithm (Bottou and Bengio, 1995), □

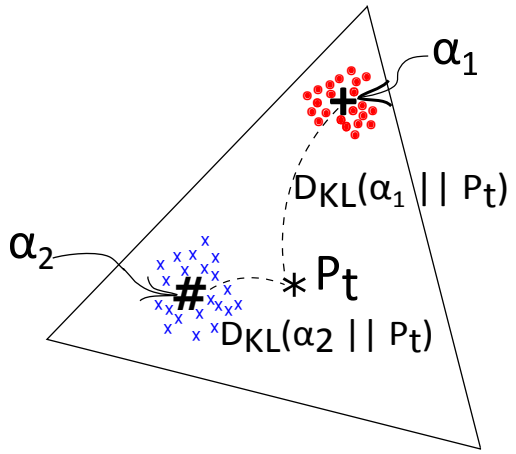


Figure 5.1: Dual Clustering in the Simplex with KL-divergence

5.4.3 Connection with Dual ME

The DCME is reminiscent of the dual ME, and we show the following results in the classification setting:

Theorem 5.4.2. *The dual form of DCME in classification is:*

$$\max_{\substack{\boldsymbol{\alpha}_k \in \Delta_N, 1 \le k \le K \\ 1 \le c_t \le K, 1 \le t \le M}} \quad \sum_{t=1}^{M} \mathbf{H}(\boldsymbol{\alpha}_{c_t}) \tag{5.20}$$

$$\textit{subject to} \quad \sum_{t=1}^{M} \mathbb{1}[i_t = j]\, \mathbf{x}_t = \sum_{t=1}^{M} \alpha_{c_t,j} \mathbf{x}_t, \;\; 1 \le j \le N$$

The proof is omitted because it is very similar to the derivation of dual ME. However, the dual form of DCME provides us with intuition of how DCME works: To approximate P_t, the cluster center is restricted to reproduce the observed statistics. Comparing it with the dual ME where $\boldsymbol{\mu}_t$ is in place of $\boldsymbol{\alpha}_{c_t}$, we see that the dual DCME has more restricted constraints. A limiting case that DCME becomes identical to ME is when $K = M$, i.e. each instance is a singleton cluster with the only member being itself.

5.5 Experiments

We conduct experiments on tasks of text classification and word embedding, evaluating the proposed DCME approach by examining its computational and learning efficiency. For comparison, we implement two sampling-based approaches,

noise contrastive estimation (NCE) and negative sampling (NS), as well as the maximum likelihood estimation using gradient descent (GD). In order for DCME and the sampling-based approaches to have comparable training speed, we set both the cluster number K of DCME and the sampling number of NCE and NS to 20, and also control the interval between offline updates in DCME with $\beta = 1$. Two variants of DCME, DCME-Q0 and DCME-Q10, are developed, the latter of which applies the online/offline tuning with $Q = 10$. All the algorithms are run with 20 threads in parallel on a 64-bit Linux machine with the Intel Xeon 3.60GHz CPU (20 core). Our code is implemented in C and available for download at: https://www.dropbox.com/s/e6b3fj2w0lq6jbt/code.tar.gz

5.5.1 Evaluation on Text Classification

We employ the ME model to predict the publishing venue of research papers using the abstract. A public dataset ACM Digital Library is investigated. It has $162,460$ papers published at $1,236$ conferences. We hold out 10% of the documents for testing. Each paper is represented by the word count features of the top $30,000$ frequent words.

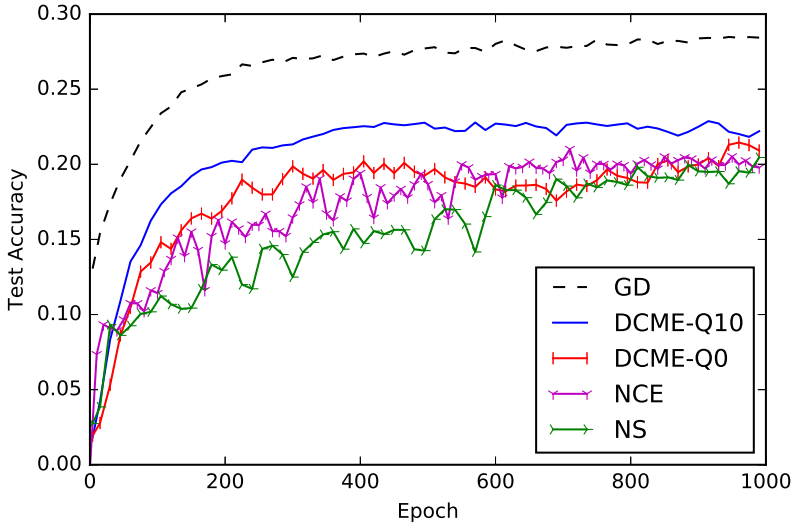
Figure 5.2a shows the learning curves of algorithms trained at each epoch, and Figure 5.2c reports the training speed. It is clear that GD does not scale well to large number (thousands or more) of items. DCME is 17-20 times faster than GD while the ratio is around 26 for sampling-based approaches. But it does give an estimation about the upper-bound performance by leveraging the exact gradient information. The curve of GD converges in the least number of iterations while the test accuracy is the highest.

DCME, on the other hand, achieves a computational efficiency similar to that of the sampling-based approaches, but the accuracy is considerably higher. Particularly, Figure 5.2a validates that DCME benefits from tuning the computation between online and offline updates. When $Q = 10$, more model parameters are updated online and there is thus less delay than that of DCME-Q0. We also note that NCE and NS produce larger variances, which is expected due to their sampling nature.

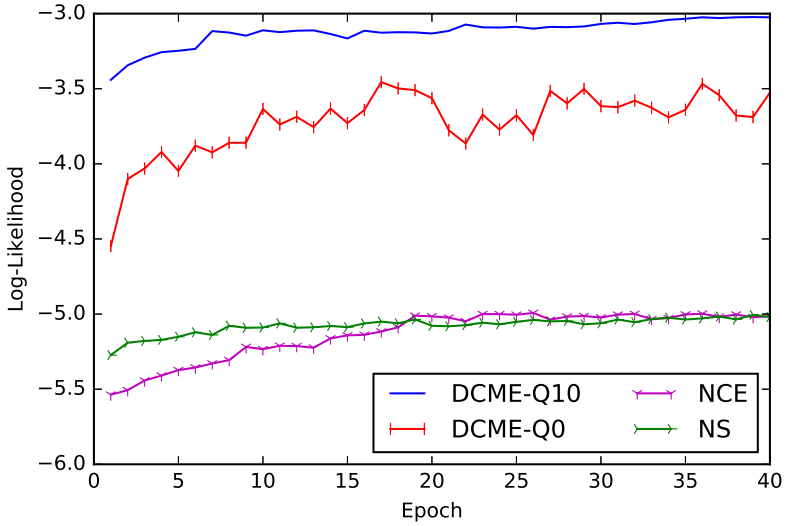
5.5.2 Evaluation on Word Embedding

For the word embedding task, we explore the New York Times (NYT) corpus from the English Gigaword (Fifth Edition). It has a total of 1.35 billion words with 10.84 million unique terms. We retain the top 1 million frequent terms in the vocabulary. To assess the performance, a randomly sampled 1×10^{-4} of the text is withheld for testing. We train the word embeddings using CBOW with a context window size of 10 and embedding dimensionality of 100.

We plot the test set average log-likelihood of each epoch in Figure 5.2b, and report the time-per-epoch statistics in Figure 5.2d. We do not evaluate GD in word embedding as it takes more than days to run one epoch. The time costs for other algorithms are similar. The results show that DCME remarkably outperforms NCE and NS. However, DCME-Q0 exhibits a large performance variance. One possible explanation is as follows. For N as large as 1 million, the interval



(a) Comparison of Test Accuracy for Classification
Trained on ACM Digital Library Dataset



(b) Comparison of Log-Likelihood for Embedding
Trained on NYT Dataset

DCME-Q10	DCME-Q0	NCE	NS	GD
9.50	7.94	6.21	6.16	165.91

(c) Time Cost (Second) per Epoch in Classification

DCME-Q10	DCME-Q0	NCE	NS
33.49	25.59	22.22	20.80

(d) Time Cost (Minute) per Epoch in Embedding

Figure 5.2: Performance on Text Classification and Word Embedding

between offline updates is so long that it creates two undesirable effects: (1) The delay results in a biased model which contributes to a large training error; (2) The offline computation changes the model drastically, as measured by the norm of the model difference, causing inconsistency when another thread accesses the model while the offline update is still in progress[8]. For DCME-Q10, minimal variance is observed. Indeed, it offers the best trade-off between learning and computational efficiency.

Model	Semantic	Syntactic	Overall
DCME-Q10	$\mathbf{-8.676}$	-8.648	$\mathbf{-8.654}$
DCME-Q0	-8.712	$\mathbf{-8.647}$	-8.663
NCE	-8.784	-8.782	-8.783
NS	-8.765	-8.679	-8.699

Table 5.1: Log-Likelihood on Semantic-Syntactic Word Relationship Dataset

To assess the quality of the trained embeddings, we use the word analogy task, which examines whether the embeddings learn the semantic/syntactic relationships of words. For instance, the question which word is similar to "small" in the same sense as "biggest" to "big" can be solved by predicting the target word with a context vector $\mathbf{h}_{biggest} - \mathbf{h}_{big} + \mathbf{h}_{small}$. We evaluate the trained word embeddings after 15 epochs. And the results on the Semantic-Syntactic Word Relationship test set (Mikolov et al., 2013) are summarized in Table 5.1, where the best performance is highlighted in bold. Again, it confirms that DCME achieves better model quality than sampling-based NCE and NS.

5.6 Conclusions

We propose a novel optimization method, Dual-Clustering Maximum Entropy (DCME), which solves the Maximum Entropy problem in its primal-dual form. Although it has a similar complexity as the sampling-based approaches, it allows the entire model to learn from every training instance, which we believe is the first algorithm that is efficient both in learning and computation. DCME exploits the dual clustering and approximates dual distributions by cluster centers. It maintains an affordable complexity using a hybrid online-offline optimization algorithm. Empirical studies demonstrate that DCME outperforms state-of-the-art algorithms such as NCE and NS in learning tasks with large numbers of items. A promising future research direction is to investigate the nonparametric mixture models for dual clustering. By taking advantages of probabilistic latent cluster assignments and learning the number of clusters from the data, we expect a better approximation for dual distributions.

[8]The model parameters are shared by all threads and there is no mutex locks on writing to the model, which is a common practice for efficiency in implementations including word2vec (https://code.google.com/p/word2vec/) and ours.

Chapter 6

PLANS: Phrasal Latent Allocation with Negative Sampling

6.1 Introduction

As the last work in this thesis, we present how PLVM can be leveraged to model the structure of sequence data and learns a meaningful representation. The modeling flexibility PLVM enjoys makes it possible to incorporate various insights into a unified model. Learning distributed representations (embeddings) of language has been a very attractive topic in recent development of natural language processing. Word embedding assign a (usually dense) low dimensional vector to each word which is supposed to retain the semantic information. For example, the differences of word vectors trained by Continuous bag-of-words or Skipgram (Mikolov and Dean, 2013), vec("$woman$") − vec("man") ≈ vec("$king$") − vec("$queen$"), are found to be close. The "common sense" of human perception, i.e., comprehending the semantics of words and their relationships, is encoded in the distributed representations. In machine learning tasks, natural language processing as an example, they are extremely valuable as being able to be learnt from abundant raw text data in a completely unsupervised manner.

Though it is advantageous to employ word embeddings for language representation, the effectiveness is inherently limited by its unigram assumption of language. On one hand, the semantics of a word is context-unaware. For instance, the word "bank" is represented by the same word embedding in the sentences "I made a deposit in the bank." and "We walked on the river bank." Therefore there is no means to discriminate the two occurrences of word "bank", whereas the semantics perceived by a human would be different based on the context. On the other hand, it treats the semantics of higher level units in language (phrase, sentence, and document) as an independent composition (linear function such as averaging) of that of each constituent word. An unappealing implication is that the formation process of meaning in language is oversimplified. Under the bag-of-words assumption, "the White House" would have a larger similarity to "a house in white" than "presidential residence", which contradicts the human understanding.

To address the two aforementioned limitations, a lot of efforts towards resolving the semantics of phrase (or sentence) have been made. Instead of using a simple averaging operation, phrase/sentence embeddings are calculated by more complicated and effective functions. Examples include convolution (Kim, 2014), attention mechanism, or general first-order or second-order transformations (Irsoy and Cardie, 2014; Le and Zuidema, 2015). Empirical evaluation

shows that by modeling the higher level language unit, the performance of text representation is greatly improved. The shared idea behind these approaches is that the semantics of the phrase/sentence can be inferred from those of its child units. Though reasonable in general, it lacks the flexibility to model the (large number of) phrases whose semantics does not depend on its constituents, taking "White House", "New York Times" as examples.

More importantly, by far most approaches assume that additional annotations are available to resolve the structure of the phrases/sentences. For instances, phrases segmentation (Yin and Schütze, 2014) is identified from anchor text in Wikipedia; And POS-tagging (Baroni and Zamparelli, 2010; Zhao et al., 2015) or syntactic parsing (Levy and Goldberg, 2014; Socher et al., 2013; Yu and Dredze, 2015) can be obtained from pre-trained parsers. Therefore it is difficult to adapt to text corpora of new domains or even another language.

The above analysis motivates us to investigate the problem of jointly recognizing the phrases and learning their embeddings. In this work, we consider a phrase as a consecutive sequence of words in a sentence and its semantics is represented by a embedding vector. Specifically, with a slight abuse of the notation, single-term words are also *phrases*. In addition, we do not assume any dependency between the phrase embedding and their constituent word embeddings, which allows us to model the meaning of phrases like "White House" and "New York Times" with sufficient flexibility. Therefore, once a phrase is identified, it will be treated no differently from a new term in the vocabulary.

Our task is essentially a much more challenging problem than settings in previous research since the segmentation of phrases is jointly learnt with the embeddings. As a preliminary step to embark on the joint learning problem, we base our analysis on the observation that *phrasal allocation* and *embedding learning* are two related tasks that can be mutually enhanced. On one hand, we have already witnessed that text representation via compositional embedding learning (Levy and Goldberg, 2014; Socher et al., 2013; Yu and Dredze, 2015) achieved better performance than word embedding learning where the text structure is explicitly given; On the other hand, (Collins and Brooks, 1995; Pantel and Lin, 2000) has demonstrated that contextual similarity, i.e., semantic similarity of context, can also be leveraged to significantly improve the resolution of the prepositional phrasal attachment. The strong mutual dependency between the structural learning and semantic learning inspires us to investigate the two subtasks in one framework.

To this end, we propose a algorithm named Phrasal Latent Allocation with Negative Sampling (PLANS), which jointly identifies the phrases and learns the embeddings. The first ingredient is the *transient Chinese restaurant process* (tCRP). We use tCRP to model the allocation of phrases as generating latent stochastic variable. Given a word in a sentence, its (left and right) boundaries of the *enclosing phrase* are generated from tCRP. Similar to Chinese restaurant process, tCRP also encourages "richer get richer", and phrases with higher frequencies are more likely to be chosen again. Nevertheless, a computational challenge confronting PLANS is to retain only a finite number of phrases while learning from a large corpus. tCRP addresses it by down-sizing the restaurant periodically: Every day a number of customers joins the restaurant and at the end of day, tables in tCRP are sorted and pruned by the number of

customers, and customers also leave the restaurant with a constant probability. Such down-size strategy can be viewed as a generalization of the *online frequency thresholding* where infinite-dimensional multinomial samples are drawn into a stream and a finite-dimensional multinomial distribution is estimated to approximate it. Another ingredient underlying PLANS, namely negative sampling (Mikolov et al., 2013), is a popular technique originally employed to train word embeddings. After the phrase allocation is determined by tCRP, negative sampling approximates the probability to generate context words given the allocated phrase, and optimizes the embedding to reflect the semantic relationships between the phrase and context words. Since Gibbs sampling is used to draw the allocated phrases and new phrases are added during the training, it is crucial to ensure numeric stability and the convergence of the phrases in tCRP. The last ingredient of PLANS, simulated annealing (SA) (Aarts and Korst, 1988; Brooks and Morgan, 1995), is investigated. In brief, SA plays a similar role in sampling as decreasing the learning step in gradient descent. When approaching the end of training, SA reduces the stochastic behavior of sampling as PLANS has more certainty about the semantics and the structure of the phrases, thus it should relies less on sampling to explore the phrasal allocation. This in turn benefits us by speeding up the convergence of the selected salient phrases and their embeddings learnt by tCRP.

Another contribution of our work is an efficient multi-thread implementation of PLANS. Hogwild (Recht et al., 2011) is investigated to optimize the phrase embeddings across threads. In addition, parallel sampling the phrasal allocation and adding new tables to tCRP are implemented with minimal lock mechanism, which strikes a balance between efficiency and robustness. We have tested our package on a Intel Xeon E5-2678 machine of 48 cores with one thread on each core and observes reliable performance at a speed of processing 2.28 million words per second.

6.2 Background

This chapter addresses the problem of jointly learning phrasal allocation and phrase embedding. To our best knowledge, this is the first work that integrates the two tasks into one framework. To give a fair account of related work, we discuss previous studies on each task.

6.2.1 Phrasal Allocation

It was a classic problem to extract phrases from unstructured text. Approaches by analyzing the co-occurrence frequencies (Lindsey et al., 2012; Wang et al., 2007; Witten et al., 1999), or information-theoretic measurements such as pointwise mutual information (PMI) (Church and Hanks, 1990; Fano and Wintringham, 1961) and generalized mutual information (GMI) (Magerman and Marcus, 1990) are proposed. One common drawback shared by methods of this line is that it is difficult to compare the importance of phrases containing variable length of words. One implication is that they are hardly scalable when learning from large corpora in the online stream setting as it is unrealistic to keep

track of all n-grams. Another aspect of limitation is that though it is hard to find segmentation of phrases with respect of the context. For example, in "New York Times Square", "New York Times" would likely to be recognized as a phrase even though a better segmentation should be "New York" and "Times square".

6.2.2 Embedding Learning

It was recently brought to people's attention that distributed representation of words can be leveraged for NLP learning tasks such as text classification. Among different word embedding algorithms, "Skip-gram" (SG) and "Continuous bag-of-words" (CBOW) are simple yet effective. The underlying idea is to model the relationship between words co-occurred in a short window. Take SG as an example, it computes the probability of seeing a context word given the target (center) word and optimizes the word embedding via Negative Sampling, which is a simplified version of the Noise-Contrastive Estimation (NCE) (Gutmann and Hyvärinen, 2010). Word embedding trained on large corpora shows good property by preserving the semantic relationship between words, for instance, vec("$woman$") $-$ vec("man") $\approx$ vec("$king$") $-$ vec("$queen$").

Beyond word embedding, compositional methods are investigated, where the embedding of a higher language unit such as phrase (or sentence): v_{AB} is computed by composition function $f(v_A, v_B)$. Since the input and the output are vectors of the same dimension, recursive neural networks are used to learn the composition, as in (Irsoy and Cardie, 2014; Le and Zuidema, 2015; Socher et al., 2013). However, they all require additional parsing tree for training and can only accommodate binary composition. Another composition, using convolution neural network, is also leveraged to model sentence embedding for classification, which achieves superior performance than simple word embedding averaging. Though compositional methods are more flexible than word embedding, they lacks the capability to model phrases whose semantics are not decomposable to its constituent words, such as "White House" and "New York Times". To address this, (Yin and Schütze, 2014) proposed a preprocessing strategy to perform phrase segmentation by a dictionary of phrases. Phrase embedding is learnt for each multi-term phrase and single-term word. Although all aforementioned works have enjoyed improved performance by modeling phrase or sentence embeddings, they did not include the phrasal allocation into their learning task.

6.3 Phrasal Latent Allocation with Negative Sampling

In this section, we present the proposed algorithm Phrasal Latent Allocation with Negative Sampling (PLANS). First, we introduce the transient Chinese Restaurant Process (tCRP), which extends the Chinese Restaurant Process to assign a prior probability to phrasal allocation. Second, we show how the phrasal embedding can be learnt with Negative Sampling (NS). Last, we show a sampling stabilizing technique, Simulated Annealing (SA), to improve the convergence

of training.

6.3.1 Phrasal Allocation as Transient Chinese Restaurant Process (tCRP)

Dirichlet Process is a stochastic process used in Bayesian nonparametrics, which extends the Dirichlet distribution to model the discrete count observations over an infinite number of outcomes. It has a nice interpretation, namely the Chinese Restaurant Process (CRP), which provides a intuitive metaphor: Suppose that there is an infinite number of tables in a Chinese restaurant, and the first customer enters the restaurant to sit at the first table. The second customer enters and decides either to sit with the first customer or alone at a new table. In general, the $n+1$-th customer either joins an already occupied table indexed by k with probability proportional to the number of customers already sitting there, or sits at a new table with probability proportional to a hyperparameter α.

We adopt CRP for phrasal allocation to allow modeling of infinite number of phrases of variable length. We assume that there is a table for each phrase and customers correspond to occurrences of phrases in the dataset. However, CRP posits two failings to properly model the phrasal allocation: (1). In a stream of words $\ldots, w_{t-1}, w_t, w_{t+1}, \ldots$, it is only reasonable for w_t to be enclosed by phrases of the form $\langle w_b, \ldots, w_e \rangle$ where $b \leq t \leq e$ and $e - b$ is small. Therefore when w_t in $\ldots, w_{t-1}, w_t, w_{t+1}, \ldots$ enters the restaurants, its choice of seating is limited by the context; (2). When learning on a corpus of very large size, such as in the online setting, it is unrealistic to run sampling of CRP over the data with sufficient epochs until convergence. Instead, customers are added into the restaurant one after anther without exiting (re-sampling). One unappealing effect is that the number of the tables as well as the customers are growing constantly over time, which will exceeds the capacity of computing resource eventually. To this end, transient Chinese Restaurant Process (tCRP) is proposed.

The first difference between tCRP and CRP is that the seating choice for a customer in tCRP is restricted. For word w_t in context $\ldots, w_{t-1}, w_t, w_{t+1}, \ldots$, seating is only possible at the tables corresponding to phrases of the form:

$$\langle w_b, \ldots, w_e \rangle$$

where $b \leq t \leq e$ and $e - b \leq L$ if we assume that the maximal length of phrases is L. Specifically, w can only join to form phrases which spans over itself and has the length no longer than L. Among those phrases there are two categories, either the ones corresponding to tables which already have customers sitting at or the ones corresponding to new tables. For the former case, tCRP assigns a probability of seating at an existing table proportional to the number of seated customers; while for the latter case, the total probability of sitting at new tables is proportional to the hyperparameter α and is shared evenly among the possible new tables. Therefore, tCRP is capable of balancing between generating existing phrases and exploring new phrases, which makes it significantly different from CRP.

Another distinguishing feature of tCRP is its "periodical shrinking mechanism". Unlike CRP where customers are constantly re-entering the restaurant in Gibbs sampling, tCRP operates in a stream fashion where infinite number of customers are entering. It is critical to maintain a economic and reasonable set of salient tables (phrases) given the limited computing resources (memory). In addition, it is also desirable to avoid the number of customers at each table increasing all the time. First, it would be numerically unstable or even causing overflow with increasing number of customers at a table; Second, in an online learning setting, models should be adaptive and pay more attention to recent data instead of obsolete samples. The "periodical shrinking mechanism" allows tCRP introduces a constant number $\mathcal{I}$ of customers per "day". At the end of each day, it sorts the tables by the number of existing customers and prunes those with fewer customers. And for each customer, he (or she) chooses to leave the restaurant with a predefined probability β. Only the remaining customers would be served in the following day. Naturally, it has an "aging" effect since the for a customer to stay in the restaurant after the i-th day the probability is $(1-\beta)^i$, which is decreasing exponentially with i. In this way, we maintain an affordable number tables (phrases) and customers (occurrences) in tCRP.

The above procedures of tCRP is summarized in Algorithm 3.

Algorithm 3: Transient Chinese Restaurant Process

for $t = 1, 2, \ldots$ **do**
  $\mathcal{P}_t \leftarrow \{\langle w_b, \ldots, b_e\rangle : b \le t \le e \text{ and } e - b \le L\}$ (feasible phrases) ;
  $\mathcal{V}_t \leftarrow$ existing tables in the restaurant;
  $\mathcal{A}_t \leftarrow \mathcal{P}_t \setminus \mathcal{V}_t$ (feasible new phrases);
  Let $\mathcal{N}(p_k)$, $(k = 1, \ldots, |\mathcal{V}_t|)$ be the number of customers sitting at the table of phrase p_k;
  Sample $k^* \in \{1, \ldots, |\mathcal{V}_t|\}$ with probability:
    if $k^* \in \mathcal{A}_t$ **then**
      $\mathrm{P}(k^*) \propto \frac{\alpha}{|\mathcal{A}_t|}$;
    else
      $\mathrm{P}(k^*) \propto \mathcal{N}(p_{k^*})$;
  if $\mathcal{I}$ *customers have been served* **then**
    Sort phrases by $\mathcal{N}(p)$;
    Prune by retaining only the top V phrases;
    Shrink for each phrase p: $\mathcal{N}(p) \leftarrow \beta\mathcal{N}(p)$;
end

6.3.2 Phrase Embedding Learning with Negative Sampling

The second ingredient of PLANS is negative sampling for estimating the phrase embeddings. We treat single-term words also as phrases. Each phrase p_k (the k-th table in tCRP) has an embedding $\mathbf{v}_k$ which is called the *output* vector. In addition, for each single-term word w (whether it is in tCRP or not), it also has a *input* vector $\mathbf{h}_w$.

Suppose that in the example of $\ldots, w_{t-1}, w_t, w_{t+1}, \ldots$ the enclosing phrase of w_t sampled from tCRP is $p_k = \langle w_b, \ldots, w_e\rangle$. Assuming that the context window is C, it's context is defined as the words of w_j where $b - C \le j < b$

or $e < j \leq e + C$. We follow the Skipgram algorithm and models the probability of seeing the phrase p_k given a context word w_j as specified by the following maximum entropy formula:

$$\mathrm{P}_{ME}(p_k|w_j) = \frac{\exp(\mathbf{v}_k^T \mathbf{h}_{w_j})}{\sum_{i=1}^{V} \exp(\mathbf{v}_i^T \mathbf{h}_j)} \tag{6.1}$$

which is computational expensive to directly optimize with the maximum likelihood estimation.

We adopt "Negative Sampling" (NS) to simplify the optimization. NS replaces the probability in (6.1) by a scoring function in the similar spirit of the noise-contrastive estimation (Gutmann and Hyvärinen, 2010). The idea converts the problem into a series of binary classification tasks, where the positive examples are the observed w_j while the negative samples are drawn from any noisy distribution $\mathcal{W}$ that is known and easy to draw sample from. Suppose that we are drawing Q samples $\{\mathcal{W}_l\} \sim \mathcal{W}$, and now we have the scoring function in NS as:

$$\mathcal{S}_{NS}(p_k|\{w_j\}, \{\mathcal{W}_l\}) = \exp\left\{ \sum_j \log \sigma(\mathbf{v}_k^T \mathbf{h}_{w_j}) + \sum_{l=1}^{Q} \log\left(1 - \sigma(\mathbf{v}_k^T \mathbf{h}_{\mathcal{W}_l})\right) \right\} \tag{6.2}$$

Intuitively, NS tries to tell apart the two groups of words, i.e., the observed context words and the noisy sampled words. Although using the scoring function $\mathcal{S}$ instead of the probability no longer preserves the statistical justification, it is computationally efficient and performs well in practice.

It is now ready to show the integration of the tCRP and NS in PLANS. For a word in sequence, the prior of selecting the enclosing phrase p_k specified by tCRP is $P_{tCRP}(p_k)$ and the likelihood is approximated by $\mathcal{S}_{NS}(p_k)$ in NS. And thus the posterior is thus to sample a phrase p_k is thus proportional to:

$$\mathrm{P}(p_k) \propto \mathrm{P}_{tCRP}(p_k)\mathcal{S}(p_k) \tag{6.3}$$

Note that the sampling is efficient: Given the maximal phrase length L and the context window C, numbers of negative samples as Q, and the embedding dimension as N, the complexity scales as $\mathcal{O}(NL^2(C+Q))$.

To learn the embeddings $\mathbf{V}$ and $\mathbf{H}$, the original optimization problem:

$$\underset{\mathbf{V},\mathbf{H}}{\text{maximize}}\, \mathbf{E}_t\left[\mathbf{E}_{k\sim \mathrm{P}_{tCRP}}[P_{ME}(p_k{}^t)]\right] \tag{6.4}$$

can now be written as:

$$\underset{\mathbf{V},\mathbf{H},\hat{k}\sim \mathrm{P}_{tCRP}}{\text{maximize}}\, \mathbf{E}_t\left[\mathcal{S}_{NS}(p_{\hat{k}}{}^t)\right] \tag{6.5}$$

where the marginalization over k has now been replaced by the posterior samples $\hat{k}$. Another way to view the

optimization is to solve the optimization problem (6.4) with Expectation-Maximization (E-M) (Dempster et al., 1977) algorithm and approximate the posterior distribution by its sampling.

6.3.3 Simulated Annealing

The stochastic behavior of posterior sampling not only affects the training of phrase and word embeddings, but also has a impact on the learnt phrases discovered in the tCRP. One potential issue is that the phrases in the restaurants may not converge fast enough. And to alleviate such stochastic randomness, we apply Simulated Annealing (SA) (Brooks and Morgan, 1995).

SA algorithm have been investigated to stochastic optimization problem where the objective is stochastic. Specifically, it is a metaheuristic to approximate global optimization in a large search space. The name and inspiration come from annealing in metallurgy, annealing a molten metal causes it to reach its crystalline state which is the global minimum in terms of thermodynamic energy. The simulated annealing algorithm was developed to simulate the annealing process. In the simulated annealing algorithm, artificial temperatures are introduced and gradually cooled, analagous to the annealing technique. This artificial temperature acts as a source of control over the stochasticity. Near the end of the annealing process, the parameters are hopefully inside the attractive local areas.

With the amount of trained data accumulating, PLANS is more certain about the phrasal allocation and the embeddings. Therefore, it is logical to decrease the stochasticity of sampling. Another motivation is to stabilize the phrase set when approaching the end of training. Intuitively speaking, this is the same idea of decreasing the learning step size for the gradient descent. Specifically, we investigate simulated annealing (SA) in PLANS, which modifies the posterior probability (6.3) for sampling with a temperature parameter T_t at time t:

$$\mathrm{P}_{SA}(p_k) \propto \mathrm{P}^{1/T_t}(p_k) \tag{6.6}$$

where $\lim_{t\to\infty} T_t = 0$. Under weak regularity assumption, it is easy to see that the probability in SA density concentrates on the mode of original distribution. In other words, the phrase with the maximum posterior probability will be deterministically selected. The temperature function, T_t, is yet to be specified. There are many annealing schedule that we can explore. The *geometric cooling*, computes the temperature as:

$$T_t = \gamma^t T_0 \tag{6.7}$$

where $0 < \gamma < 1$ is the cooling rate (Yuan et al., 2004). The geometric cooling is widely used for its quick cooling and convergence. We adopt it for scheduling the cooling and set the final temperature to 0.2 or 0.1.

6.4 A Multithread Implementation

With the development of computer hardware, it is now standard to have machines with 40 or more cores of CPU. Hogwild (Recht et al., 2011), a lock-free parallelizing stochastic optimization method, is therefore proposed.

Briefly speaking, Hogwild is an asynchronous "don't care" approach for stochastic gradient descent sharing the same parameters. That is, each thread runs training passes without explicitly synchronizing with the other threads, but they concurrently update the parameters by applying SGD updates. In practice, the threads will "race", i.e., write over each other occasionally, but that is affordable if the update over the parameters are sufficiently sparse, as in the case of embedding training.

6.4.1 Lock-Free Optimizing the Embedding

It is straightforward to optimize the (output) phrase and (input) word embedding with Hogwild since the number of phrases and words is large and it is not frequent to have collision of parameter updating. When multiple threads optimizes $\mathbf{H}, \mathbf{V}$ by (6.5) in parallel, the back-propagation only involves the sampled phrase p_k, the context words w_j and the negative samples $\mathcal{W}_l$. Since threads are scheduled to work on different sections of the corpus, and the negative samples are randomly drawn from the noisy distribution, it is hence of low probability for a racing condition to occur where the embedding of the same phrase/word is being updated by different threads at the same time.

When a racing condition "unfortunately" occurred, each thread is trying to apply its gradient multiplied by the learning step size to update the embedding vector. Since the learning step size is small, the update is also of small values, which can only result a small amount of uncertainty in the parameters after collision. And through the long time training, the pollution due to the racing can be forgiven.

6.4.2 Minimal-Lock for Phrasal Allocation

The racing condition becomes a crucial issue when updating the restaurants. Specifically, two operations are mostly impacted by the multithread computation: 1) adding a new table in the restaurant; and 2) periodically shrinking the restaurant.

The restaurant is stored in memory using hash table data structure. When two tables of the same hash value are added to the restaurant, it will cause the hash table to fail [1] However, we expect such racing condition to be rare since collision in hash table is not frequent in general. We solve the problem by assigning each hash slot a mutex lock. Adding phrase to the hash table with hash value h can only proceed when the mutex lock for h is successfully obtained. Otherwise, the thread will wait until other thread releases the lock.

[1] the detail of crash is implementation dependent. For example if for each hash slot a linked list is stored, it will cause one of the added table missing, or the linked list broken.

Another situation we need to consider is the periodical shrinking. Since each thread may invoke the shrinking independently, it is possible that more than one threads are shrinking the restaurant concurrently. We use another mutex lock to avoid the racing. Nevertheless, after sorting the tables, removing tables with fewer customers may cause failures the same way as adding tables. The difference between removing and adding is that only one table is added at a time while removing involves many tables consecutively. And thus it is not efficient to lock each corresponding hash slot at removal time. Instead, we construct another restaurant and only add retained tables to the new restaurant. After the construction of the new table, the thread will broadcast the change of the restaurant and all threads will start working on the new restaurant instead.

6.5 Experiments

We present experimental results in this section and evaluate Phrasal Latent Allocation with Negative Sampling (PLANS) quantiatively and qualitatively. As our work is the first to jointly identify phrasal allocations and to learn the embeddings, we will discuss the performance on each task separately. Furthermore, a sensitivity analysis is conducted where the parameters in PLANS are varied and an in-depth discussion is provided.

We assess the performance of PLANS by exploring a large corpus, the New York Times (NYT) corpus from the English Gigaword (Fifth Edition). It has a total of 1.35 billion words and we retain the top 0.1 million frequent terms in the vocabulary. In the experiment, hyperparameters in PLANS are set as: the maximal phrase length $L = 10$, context length $C = 5$, and number of negative samples $Q = 5$. We train the phrase (output) and word (input) embedding with a dimensionality of $N = 100$.

In tCRP, the concentration hyperparameter $\alpha = 5$. For periodical shrinking, $0.5M$ customers are admitted to the restaurant each day. We retain the top $0.75M$ tables after sorting the tables by the number of customers. However, we find that it is not economic to perform sorting immediately when the number of tables in the restaurant exceeds $V = 0.75M$. Instead, we only sort and prune the tables when there is $2V = 1.5$ million tables in the tCRP. And if the condition is met, we reduce the size of tCRP to V tables. Also, customers leave the restaurant each day with a probability $\beta = 0.99$.

We initialize the embeddings with uniform random values in the range $[-1 \times 10^{-4}, 1 \times 10^{-4}]$. A heuristic that we find effective practically is to add all single-term words as phrases into the restaurant with a small number of customers (e.g. 5) before training. Simulated Annealing by geometric cooling is incorporated in PLANS with an initial temperature at 1 and the final temperature at 0.2.

Our algorithm runs with 48 threads on a 64-bit Linux with an Intel Xeon E5-2678 v3 2.50GHz CPU. Our code is implemented in C and available for download at: https://www.github.com/dragonxlwang/phrase

6.5.1 Evaluating the Phrasal Allocation

To collect the groundtruth phrases, we followed the approach in (Yin and Schütze, 2014). Canonical phrases are extracted by finding the anchor text from Wikipedia. We sort them by the frequency and keep the phrases that appear more than 1000 times in NY Times, which leaves us 2249 phrases in the groundtruth set.

A simple baseline of Pointwise Mutual Information (PMI) is compared against PLANS. PMI is defined as:

$$PMI(X,Y) = \mathbf{E}[\frac{\mathrm{P}(X,Y)}{\mathrm{P}(X)\mathrm{P}(Y)}] \tag{6.8}$$

And phrases are generated by choosing the bi-grams with higher PMI values. After inspecting the PMI result, we identified a list of 2370 bi-grams as phrases.

To compare fairly with the PMI result, we select the 2370 phrases with most customers in tCRP from PLANS. We assess the precision, recall and F1 scores and report the result in the Table. 6.1 below:

	PLANS	PMI
Precision	0.435	0.234
Recall	0.458	0.222
F-1	0.446	0.228

Table 6.1: Phrasal Allocation Evaluation

From the table, we observe that PLANS achieves a much higher precision, recall and F1 scores than the PMI approach. Although PLANS shares the same property as PMI that the co-occurred words are encouraged to form into phrases, there are two characteristics that PLANS possesses which contribute to the better performance: First, PLANS takes the semantics of phrases into account; and second, PLANS is capable of modeling phrases of variable lengths.

To qualitatively evaluate the learnt phrases, the top 50 multi-term phrases and their number of customers in tCRP are listed in Table. 6.2. Phrases such as "NY Times news service" and "Standard & Poor 500" are all recognized and have large number of frequencies. We find that a number of top phrases are named entities the semantics of which are not easily decomposable into those of its constituent words. A simple analysis can be drawn from how PLANS works: It tries to predict the context words given the phrases. Take the phrase "Standard & Poor" as an example: If "poor" is sampled as a single-term phrase, then it is for "poor" to predict the context words such as "index", "stock", or "share"; However, since the meaning of the word "poor" is more often used as "lacking sufficient money to live", it is also for "poor" to predict words such as "money", "family", "person". Instead, if "Standard & Poor" is sampled as a single phrase then only "money", "family", "person" are the context of "poor" while "index", "stock", "share" are the context of "Standard & Poor", which gives higher flexibility for the model to find the optimal solution.

Phrases (1-25)	Customers	Phrases (26-50)	Customers
NY Times news service	9439.186835	billion yen	3650.450553
Standard & Poor 500	9333.408124	discount rate	3564.913991
Goldman Sachs	9208.101699	N.Y. times	3540.285354
Merrill Lynch	6696.802137	downgraded market	3530.532261
Hearst news service arizona	6640.307863	U.S bond	3492.175846
stock fall	6402.516529	stock market	3487.573450
stock rise	6400.599538	Canadian dollar	3365.092222
bad loans	6017.643467	attorney general	3300.376802
intel corp	5446.546919	photo service	3291.761087
30-year bond	5072.992919	moon phases	3287.815149
interest rates	4892.737120	Japanese bond	3192.634862
U.S treasury	4778.023305	domestic product	3173.692085
South Korea	4776.446476	San Francisco	3157.388493
Walt Disney Co.	4768.185523	computer corp	3088.544863
United States	4322.587908	please call	3064.043981
Lockheed Martin	4239.521394	internal revenue	3063.152179
coffee mug	4223.394590	White House	3039.420467
rating remained	4217.733959	Taxes Instruments	3033.578153
trade deficit	4173.724859	security inc	3025.247462
u.s cents	4165.137035	news service	2971.026775
outperform analyst expectation	4142.661312	daily weather	2937.203039
Los Angeles	4071.305252	banking system	2926.525300
per share	4038.729819	Sao Paulo	2925.857946
borrowing costs	4008.017119	Boston globe bos	2753.314104
earning rise	3980.897046	Nasdaq composite	2748.102835
world war II	3946.853762	Times Syndication Service	2743.282522

Table 6.2: Top phrases in tCRP

6.5.2 Evaluating the Phrase Embedding

PLANS also evaluates the embedding for each phrase in the restaurant. To assess the performance, we show 5 nearest neighbors for each phrase below as computed by cosine similarity.

In Table. 6.3, 10 phrases of location, person, scientific and economic terminology, and name of university are showed. Most nearest-neighbor phrases are of the same type as the query phrase. Also, they also share semantic similarity. For example, neighbors of "San Jose-based" are all locations where technology companies are located and those of "university of illinois at urbana-champaign" are universities in the mid-west or being famous for its engineering.

6.5.3 Sensitivity Analysis

The above experiments are run with the initial gradient descent step size at 1×10^{-3}, the final temperature at 0.2 and the shrinking rate $\beta = 0.99$. In the sensitivity analysis, we vary these parameters and examine the training behavior of PLANS.

Phrase	Similarity	Phrase	Similarity
NY Times		**White House**	
Bloomberg news	0.828	United States	0.814
according recent	0.798	House members	0.808
telephone interview	0.773	Clinton administration	0.798
front page	0.705	President Bush	0.781
New York Times Syndicate	0.701	Prime Minister	0.780
Keanu Reeves		**Linkin Park**	
Sigourney Weaver	0.965	Rascal Flatts	0.945
Ving Rhames	0.943	Def Leppard	0.941
Charlize Theron	0.941	Gnarls Barkley	0.937
Benicio Del Toro	0.940	Van Halen	0.918
Keira Knightley	0.938	Bon Jovi	0.905
macular degeneration		**Feng Shui**	
rheumatoid arthritis	0.922	home project	0.778
atrial fibrillation	0.905	zen	0.754
kaposi sarcoma	0.903	Tabula rasa	0.609
squamous cell	0.885	Joie de vivre	0.598
human immunodeficiency	0.870	De Botton	0.591
TWSE index		**Lee Teng-Hui**	
Heng Seng index	0.996	Masao Iwasato	0.946
KOSPI index	0.901	Chen Shui-Bian	0.883
Indu index	0.894	Kim Dae-Jung	0.837
Gudang Garam	0.891	Jiang Zemin	0.836
Japan Nikkei 225	0.871	Wen Jiabao	0.829
San Jose-based		**university of illinois at urbana-champaign**	
Santa Clara-based	0.966	university of wisconsin-madison	0.971
Mountain View-based	0.961	university of missouri-kansas	0.935
Palo Alto-based	0.929	university of witwatersrand	0.923
San Francisco-based	0.874	university of california-berkeley	0.911
Thousand Oaks-based	0.871	university of missouri-columbia	0.866

Table 6.3: Nearest Neighbors of Phrases

To this end, we specifically plot the curves of average customers per table, the number of tables, the number of days when tables are pruned, and the ratio between the number of customers being assigned to a new table and all customers. In Figure 6.1, the X-axis is the number of customers entering the restaurant so far. With 48 threads, it was found that appropriate gradient descent step size ranges from 5×10^{-4} to 4×10^{-3} Useful final temperature is from 0.1 to 0.5 and the shrink rate β from 0.95 to 0.999.

Note that the in the curve 6.1a, β is the most influencing factor. With a larger $\beta = 0.999$, fewer customers are leaving the restaurant per day. This also contributes to the fact that the tCRP will explore fewer new tables than with a smaller $\beta = 0.99$, as seen in Figure 6.1d; we see a smaller ratio of customers is assigned to new tables. In addition, from Figure 6.1c, we observe that both a larger β or a smaller final temperature can yield fewer number of table pruning. To see this, note that with a small final temperature, the stochasticity of PLANS is reduced exponentially.

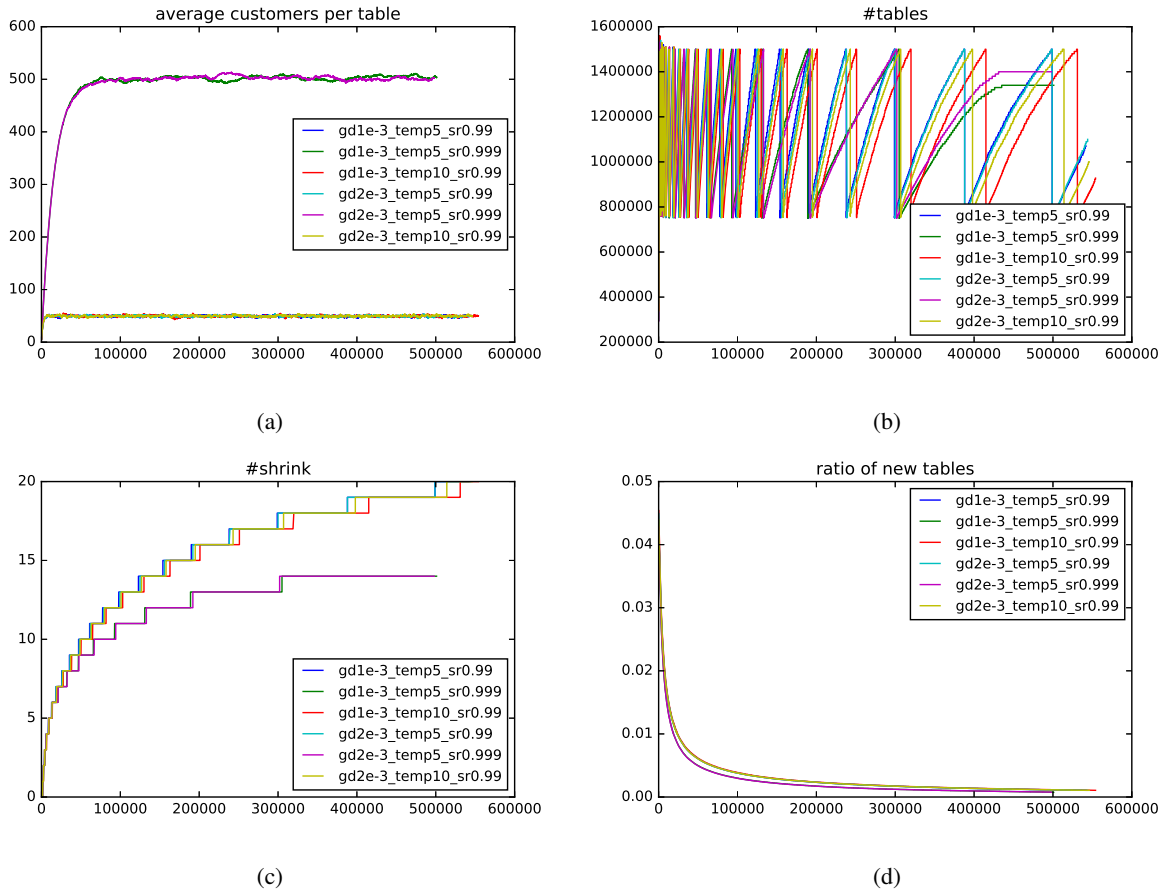


Figure 6.1: Curves of average customers per table, number of tables, the number of days when tables are pruned, and ratio of customers being assigned to a new table. The x-axis is the total number of customers so far.

6.5.4 Classification Experiment

In this part, we will use the learnt embedding of phrase to represent documents and see if that can benefit text classification. It was previously shown that by combining low-dimensional embedding with bag-of-words representation, the performance can be improved.

We use a classic sentiment classification dataset (Pang et al., 2002), which has a set of 700 positive and 700 negative processed movie reviews collected from the IMDB archive. The dataset is tokenized and words are normalized already. We followed the use of the data by dividing it into three equal-sized folds, maintaining balanced class distribution in each fold. All results reported below are the average of three-fold cross-validation evaluation on the dataset. We use the Logistic-Regression in Scikit-Learn package (Pedregosa et al., 2011) for the binary (positive/negative) polarity classification with different representation strategies for the review documents.

In the experiment, we find $31,240$ unique terms in the reviews. However, note not all of those words are the top 0.1M frequent words in the NY Times corpus and it might not always true that there is an embedding vector for

	Sparse Features	Sparse Dim	Dense Features	Dense Dim	Accuracy
(1)	Unigram	31,240	*N.A*	0	67.512
(2)	*N.A*	0	Word	100	58.471
(3)	*N.A*	0	Phrase	100	60.332
(4)	Unigram	31,240	Word	100	73.637
(5)	Unigram	31,240	Phrase	100	77.952

Table 6.4: Average three-fold cross-validation accuracies, in percent.

each word in the reviews. From the above Table 6.4, we see that with the sparse feature (unigram) only, the baseline performance is merely 67.512%. The unigram baseline is better than only using dense embedding as features. We offer two reasons to explain this result: 1) Some discriminative words for sentiment polarity in the dataset might not has an embedding learnt from the NY Times and therefore information of those words is lost in the embedding representation; and 2) The dimension of dense embeddings is only 100 while the sparse feature has a dimension of $31,240$. Thus models learnt with only dense embeddings is limited in its capacity to fit the training data and it is expected that the performance is worse than (1). Comparing (2) and (3), it is seen that with phrase embeddings learnt from PLANS, there is still a marginal improvement in performance.

Nevertheless, when combining sparse unigram and dense embedding together, we observe the best performances. The accuracy for "unigram+word" (4) is 73.637 while the accuracy reaches 77.952 for "unigram+phrase" (5). It is clear that phrase embeddings learnt by PLANS significantly boost the performance for polarity classification than that of word embedding. This is also consistent with the finding by (Pang et al., 2002) that bigrams can also benefit the sentiment classification.

6.6 Conclusion

We propose a novel model, Phrasal Latent Allocation with Negative Sampling (PLANS), to jointly learn the phrasal allocation and the embeddings. Although previous study have separately investigated either of the subtasks, PLANS is the first to address the two problems in a fully unsupervised fashion. PLANS has three main ingredients: 1). A transient Chinese Restaurant Process (tCRP) is proposed to model possibly infinite discrete observations in a stream while maintaining an economic and affordable size of tables and customers by periodical shrinking; 2). Negative sampling (NS) is integrated to efficiently estimate the embedding of phrases; and 3). Simulated Annealing (SA) with geometric cooling stabilizes PLANS by reducing the stochastic behavior towards the end of training. In addition, we implement PLANS with multi-threads with a modified Hogwild algorithm which ensures fast training. Empirical studies demonstrate that PLANS is able to identify meaningful phrases and accurately estimate the semantic embeddings.

Chapter 7

Conclusions

In this thesis, I describe a range of practical real-world applications where latent variables can be leveraged for effective knowledge discovery and efficient optimization. I designed novel probabilistic latent variable models which manage to model complex distributions through appropriate choices of the latent variables.

Firstly, I demonstrated that by modeling literature citations as observations of a generative model with latent variables, research topics as well as evolution themes of research can be identified and described inactively. The proposed model 1) discovers research topics, which includes finding milestone papers, computing topic temporal strength, and extracting keywords for topics; and 2) identifies research theme evolution, which includes identifying topic importance, learning topic dependency relation, and recognizing the evolution patterns. These computational components together enable us to understand evolution of research themes by constructing the evolution graph. This work can be very useful to help researchers digest literature quickly, thus speeding up scientific research discovery and delivering very broad positive impact on the society. In general, the model can also be applied to any graph data for tasks such as network clustering and ranking, as well as modeling the evolution of network generation.

Secondly, I proposed a framework where a ranked list can be inferred from pairwise preferences labelled by non-expert workers in crowdsourcing, which is highly useful in various data mining and information retrieval tasks such as learning to rank. Latent variables are introduced to model query difficulty and query domain, as well as worker expertise and truthfulness, effectively resolving the inevitable incompleteness and inconsistency of pairwise judgements. In addition, by employing latent variables, intractable distributions are effectively sampled, and thus efficient computation is accomplished.

Thirdly, I proposed a novel approach, Dual-Clustering Maximum Entropy, which addresses the stability problem of Maximum Entropy when there is an extreme large number of items (classes/words) present. Latent variables are employed for model reduction and facilitate inference. By incorporating the modeling of latent variables, the dual space of the Maximum Entropy problem is explored and a K-means like clustering is conducted over the simplex space. The use of PLVM leads to an efficient algorithm, the complexity of which does not depend on the number of items.

In the end, we propose a novel model, Phrasal Latent Allocation with Negative Sampling (PLANS), to jointly learn the phrasal allocation and the embeddings. PLANS consists of three key components: 1) A transient Chinese

Restaurant Process (tCRP) is proposed to model possibly infinite discrete observations in a stream while maintaining an economic and affordable size of tables and customers by periodical shrinking; 2) Negative sampling (NS) is integrated to efficiently estimate the embedding of phrases; and 3) Simulated Annealing (SA) with geometric cooling stabilizes PLANS by reducing the stochastic behavior towards the end of training. By fitting the unstructured text with underlying phrasal structures, it is demonstrated that both the phrasal allocation and phrase embeddings are effectively computed.

Overall, in this thesis, we have explored a wide range of applications where Probabilistic Latent Variable Models (PLVMs) can efficiently model data of different types or greatly improve the performance in terms of efficiency and scalability. Specifically, we show in this thesis that:

- PLVMs are a very flexible approach for modeling complex observations (such as networks, ranked lists, or sequences) by incorporating latent variables into the generative modeling.
- PLVMs are a powerful tool for knowledge discovery and data mining. By encoding the useful information as latent variables and modeling them with feasible generative process, it can significatnly simplify the computation and achieves good performance.
- PLVMs can also be leveraged for efficient and salable optimization. As one example shown in the thesis, posterior sampling can be leveraged as E-step in the E-M algorithm.

It is our expectation that PLVMs benefit many other research topics in machine learning and data mining. The general methodology is that PLVMs allow us to model complex observations by assuming simpler generation at the cost of incorporating latent variables into modeling. More importantly, those "artificially" added latent variables are in fact statistically meaningful in most applications, as they preserve crucial information of the data. In addition, another merit of PLVMs is that it provides a principle means to develop scalable and efficient algorithms for inference. It would be useful to explore other applications of PLVMs that could benefit from the idea of modeling with latent variables in the future.

Chapter 8

References

E. Aarts and J. Korst. Simulated annealing and boltzmann machines. 1988.

E. M. Airoldi, D. M. Blei, S. E. Fienberg, E. P. Xing, and T. Jaakkola. Mixed membership stochastic block models for relational data with application to protein-protein interactions. In *Proceedings of the international biometrics society annual meeting*, 2006.

J. A. Aslam and M. Montague. Models for metasearch. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 276–284. ACM, 2001.

M. Baroni and R. Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193. Association for Computational Linguistics, 2010.

Y. Bengio and J.-S. Senécal. Adaptive importance sampling to accelerate training of a neural probabilistic language model. *IEEE Transactions on Neural Networks*, 19(4):713–722, 2008.

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *journal of machine learning research*, 3(Feb):1137–1155, 2003.

A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71, 1996.

D. P. Bertsekas, A. Nedi, A. E. Ozdaglar, et al. Convex analysis and optimization. 2003.

D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

D. M. Blei and J. D. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, pages 17–35, 2007.

D. M. Blei and J. D. Lafferty. Topic models. *Text mining: classification, clustering, and applications*, 10(71):34, 2009.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan): 993–1022, 2003.

D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):7, 2010.

U. Böckenholt. Applications of thurstonian models to ranking data. In *Probability models and statistical analyses for ranking data*, pages 157–172. Springer, 1993.

L. Bolelli, S. Ertekin, and C. Giles. Clustering scientific literature using sparse citation graph analysis. *Knowledge Discovery in Databases: PKDD 2006*, pages 30–41, 2006.

L. Bottou and Y. Bengio. Convergence properties of the k-means algorithms. In *Advances in Neural Information Processing Systems*, pages 585–592, 1995.

J. Boyd-Graber, J. Chang, S. Gerrish, C. Wang, and D. Blei. Reading tea leaves: How humans interpret topic models. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, 2009.

S. P. Brooks and B. J. Morgan. Optimization using simulated annealing. *The Statistician*, pages 241–257, 1995.

A. Burgess and J. A. Tully. On the bethe approximation. *Journal of Physics B: Atomic and Molecular Physics*, 11(24): 4271, 1978.

M. A. Carreira-Perpinan and G. Hinton. On contrastive divergence learning. In *AISTATS*, volume 10, pages 33–40. Citeseer, 2005.

J. Chang and D. Blei. Relational topic models for document networks. In *Artificial Intelligence and Statistics*, pages 81–88, 2009.

X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 193–202. ACM, 2013.

M. Chiani, D. Dardari, and M. K. Simon. New exponential bounds and approximations for the computation of error probability in fading channels. *IEEE Transactions on Wireless Communications*, 2(4):840–845, 2003.

N. Chopin. Fast simulation of truncated gaussian distributions. *Statistics and Computing*, 21(2):275–288, 2011.

A. Choromanska, A. Agarwal, and J. Langford. Extreme multi class classification. In *NIPS Workshop: eXtreme Classification, submitted*, 2013.

A. E. Choromanska and J. Langford. Logarithmic time online multiclass prediction. In *Advances in Neural Information Processing Systems*, pages 55–63, 2015.

K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

M. Collins and J. Brooks. Prepositional phrase attachment through a backed-off model. *arXiv preprint cmp-lg/9506021*, 1995.

M. Collins, A. Globerson, T. Koo, X. Carreras, and P. L. Bartlett. Exponentiated gradient algorithms for conditional random fields and max-margin markov networks. *Journal of Machine Learning Research*, 9(Aug):1775–1822, 2008.

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The annals of mathematical statistics*, pages 1470–1480, 1972.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

R. M. Fano and W. Wintringham. Transmission of information. *Physics Today*, 14:56, 1961.

G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis. Graph clustering and minimum cut trees. *Internet Mathematics*, 1 (4):385–408, 2004.

J. Gao, G. Andrew, M. Johnson, and K. Toutanova. A comparative study of parameter estimation methods for statistical natural language processing. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 824, 2007.

E. Garfield. The history and meaning of the journal impact factor. *JAMA: the journal of the American Medical Association*, 295(1):90–93, 2006.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

Z. Ghahramani and T. L. Griffiths. Infinite latent feature models and the indian buffet process. In *Advances in neural information processing systems*, pages 475–482, 2005.

R. Ghosh, T.-T. Kuo, C.-N. Hsu, S.-D. Lin, and K. Lerman. Time-aware ranking in dynamic citation networks. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 373–380. IEEE, 2011.

D. Griffiths and M. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 16:17, 2004.

T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, volume 1, page 6, 2010.

B. Hajek. *Random Processes for Engineers*. Cambridge University Press, 2015.

K. Henderson and T. Eliassi-Rad. Applying latent dirichlet allocation to group discovery in large graphs. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 1456–1461. ACM, 2009.

J.-B. Hiriart-Urruty and C. Lemarechal. Convex analysis and minimization algorithms. ii. 1993.

J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United states of America*, 102(46):16569, 2005.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196, 2001.

F.-L. Huang, C.-J. Hsieh, K.-W. Chang, and C.-J. Lin. Iterative scaling and coordinate descent methods for maximum entropy models. *Journal of Machine Learning Research*, 11(Feb):815–848, 2010.

O. Irsoy and C. Cardie. Deep recursive neural networks for compositionality in language. In *Advances in Neural Information Processing Systems*, pages 2096–2104, 2014.

H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 2011.

A. N. Iusem and M. Teboulle. A primal-dual iterative algorithm for a maximum likelihood estimation problem. *Computational statistics & data analysis*, 14(4):443–456, 1992.

K. Järvelin and J. Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48. ACM, 2000.

Y. Jo, J. E. Hopcroft, and C. Lagoze. The web of topics: discovering the topology of topic evolution in a corpus. In *Proceedings of the 20th international conference on World wide web*, pages 257–266. ACM, 2011.

M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

A. Klementiev, D. Roth, and K. Small. Unsupervised rank aggregation with distance-based models. In *Proceedings of the 25th international conference on Machine learning*, pages 472–479. ACM, 2008.

A. Klementiev, D. Roth, K. Small, and I. Titov. Unsupervised rank aggregation with domain-specific expertise. *Urbana*, 51:61801, 2009.

H. Komiya. Elementary proof for sion's minimax theorem. *Kodai mathematical journal*, 11(1):5–7, 1988.

P. Le and W. Zuidema. Compositional distributional semantics with long short term memory. *arXiv preprint arXiv:1503.02510*, 2015.

O. Levy and Y. Goldberg. Dependency-based word embeddings. In *ACL (2)*, pages 302–308, 2014.

R. V. Lindsey, W. P. Headden III, and M. J. Stipicevic. A phrase-discovering topic model using hierarchical pitman-yor processes. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 214–222. Association for Computational Linguistics, 2012.

R. D. Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2005.

D. M. Magerman and M. P. Marcus. Parsing a natural language using mutual information statistics. In *AAAI*, volume 90, pages 984–989, 1990.

C. L. Mallows. Non-null ranking models. i. *Biometrika*, 44(1/2):114–130, 1957.

R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–7. Association for Computational Linguistics, 2002.

G. McLachlan and D. Peel. *Finite mixture models*. John Wiley & Sons, 2004.

Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 198–207. ACM, 2005.

T. Mikolov and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

T. Minka et al. Divergence measures and message passing. Technical report, Technical report, Microsoft Research, 2005.

T. P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001a.

T. P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001b.

A. Mnih and G. Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pages 641–648. ACM, 2007.

A. Mnih and G. E. Hinton. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088, 2009.

A. Mnih and Y. W. Teh. A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint arXiv:1206.6426*, 2012.

F. Morin and Y. Bengio. Hierarchical probabilistic neural network language model. *AISTATS 2005*, page 246, 2005.

R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 542–550. ACM, 2008.

K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67, 1999.

B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

P. Pantel and D. Lin. An unsupervised approach to prepositional phrase attachment using contextually similar words. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 101–108. Association for Computational Linguistics, 2000.

K. Pearson. Mathematical contributions to the theory of evolution. iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 187:253–318, 1896.

K. Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J. Pfanzagl. *Parametric statistical theory*. Walter de Gruyter, 1994.

J. Pitman. Exchangeable and partially exchangeable random partitions. *Probability theory and related fields*, 102(2): 145–158, 1995.

J. Pitman and M. Yor. The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, pages 855–900, 1997.

J. Pitman et al. Combinatorial stochastic processes. 2002.

R. L. Plackett. The analysis of permutations. *Applied Statistics*, pages 193–202, 1975.

A. Popescul, G. W. Flake, S. Lawrence, L. H. Ungar, and C. L. Giles. Clustering and identifying temporal trends in document databases. In *Advances in Digital Libraries, 2000. ADL 2000. Proceedings. IEEE*, pages 173–182. IEEE, 2000.

V. Qazvinian and D. Radev. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 689–696. Association for Computational Linguistics, 2008.

D. Radev, P. Muthukrishnan, and V. Qazvinian. The acl anthology network corpus. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, pages 54–61. Association for Computational Linguistics, 2009.

C. E. Rasmussen and C. K. Williams. Gaussian processes for machine learning. 2006.

A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. Association for Computational Linguistics, 1996.

B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 693–701, 2011.

H. Sayyadi and L. Getoor. Futurerank: Ranking scientific articles by predicting their future pagerank. In *Proc. of the 9th SIAM International Conference on Data Mining*, pages 533–544, 2009.

M. Sion et al. On general minimax theorems. *Pacific J. Math*, 8(1):171–176, 1958.

P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi. Inferring ground truth from subjective labelling of venus images. 1995.

R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng. Parsing with compositional vector grammars. In *ACL (1)*, pages 455–465, 2013.

M. Steyvers, B. Miller, P. Hemmer, and M. D. Lee. The wisdom of crowds in the recollection of order information. In *Advances in neural information processing systems*, pages 1785–1793, 2009.

A. Stuart, M. G. Kendall, et al. *The advanced theory of statistics*. Charles Griffin, 1968.

Y. W. Teh. Dirichlet process. In *Encyclopedia of machine learning*, pages 280–287. Springer, 2011.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the american statistical association*, 2012.

L. L. Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.

Y. Tsuruoka, J. Tsujii, and S. Ananiadou. Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *Proceedings of the 47th Annual Meeting of the ACL*, pages 477–485. Association for Computational Linguistics, 2009.

H. Valizadegan, R. Jin, R. Zhang, and J. Mao. Learning to rank by optimizing ndcg measure. In *Advances in neural information processing systems*, pages 1883–1891, 2009.

J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, 1928.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

D. Walker, H. Xie, K.-K. Yan, and S. Maslov. Ranking scientific publications using a model of network traffic. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(06):P06010, 2007.

C. Wang and D. M. Blei. Truncation-free online variational inference for bayesian nonparametric models. In *Advances in neural information processing systems*, pages 413–421, 2012.

C. Wang, D. Blei, and D. Heckerman. Continuous time dynamic topic models. *arXiv preprint arXiv:1206.3298*, 2012.

X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433. ACM, 2006.

X. Wang, A. McCallum, and X. Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 697–702. IEEE, 2007.

X. Wang, C. Zhai, and D. Roth. Understanding evolution of research themes: a probabilistic generative model for citations. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1115–1123. ACM, 2013.

X. Wang, J. Wang, J. Luo, C. Zhai, and Y. Chang. Elephant and blind men: Thurstonian pairwise preference for ranking in crowdsourcing. In *ICDM*, 2016a.

X. Wang, J. Wang, and C. Zhai. Dual-clustering maximum entropy with application to classification and word embeding. In *under review*, 2016b.

W. F. R. Weldon. On certain correlated variations in carcinus maenas. *Proceedings of the Royal Society of London*, 54 (326-330):318–329, 1893.

P. Welinder, S. Branson, P. Perona, and S. J. Belongie. The multidimensional wisdom of crowds. In *Advances in neural information processing systems*, pages 2424–2432, 2010.

J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, pages 2035–2043, 2009.

I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, pages 254–255. ACM, 1999.

Y. Yan, R. Rosales, G. Fung, M. W. Schmidt, G. H. Valadez, L. Bogoni, L. Moy, and J. G. Dy. Modeling annotator expertise: Learning when everybody knows a bit of something. In *AISTATS*, pages 932–939, 2010.

G. Yao and U. Böckenholt. Bayesian estimation of thurstonian ranking models based on the gibbs sampler. *British Journal of Mathematical and Statistical Psychology*, 52(1):79–92, 1999.

J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, 2005.

I. E. Yen, X. Huang, P. Ravikumar, K. Zhong, and I. S. Dhillon. Pd-sparse: A primal and dual sparse approach to extreme multiclass and multilabel classification. In *Proceedings of the 33nd International Conference on Machine Learning*, pages 3069–3077, 2016.

W. Yin and H. Schütze. An exploration of embeddings for generalized phrases. In *ACL (Student Research Workshop)*, pages 41–47, 2014.

H.-F. Yu, F.-L. Huang, and C.-J. Lin. Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 85(1-2):41–75, 2011.

M. Yu and M. Dredze. Learning composition models for phrase embeddings. *Transactions of the Association for Computational Linguistics*, 3:227–242, 2015.

C. Yuan, T.-C. Lu, and M. J. Druzdzel. Annealed map. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 628–635. AUAI Press, 2004.

G.-X. Yuan, C.-H. Ho, and C.-J. Lin. Recent advances of large-scale linear classification. *Proceedings of the IEEE*, 100 (9):2584–2603, 2012.

Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 271–278. ACM, 2007.

H. Zhang, B. Qiu, C. L. Giles, H. C. Foley, and J. Yen. An lda-based community structure discovery approach for large-scale social networks. In *Intelligence and Security Informatics, 2007 IEEE*, pages 200–207. IEEE, 2007.

Y. Zhao, Z. Liu, and M. Sun. Phrase type sensitive tensor indexing model for semantic composition. In *AAAI*, pages 2195–2202, 2015.

Appendix A

Supplementary results on Thurstonian Pairwise Preference

A.1 Model Updating

By zeroing the derivatives of $\mathcal{Q}(\mathbf{\Theta}^{(t+1)}; \mathbf{\Theta}^{(t)})$ with respect to $\mathbf{\Theta}^{(t+1)}$, the following closed forms are obtained for the update of $\{s_{l,i}^{(t+1)}\}$, $\{\delta_l^{2(t+1)}\}$ and $\{\theta_m^{(t+1)}\}$:

$$\begin{cases} s_{l^*,i^*}^{(t+1)} = \dfrac{\sum\limits_{k \in \mathcal{W}_{l^*,i^*}} \mathbf{E}_{\mathbf{Z}|\mathbf{D},\mathbf{\Theta}^{(t)}}[s_{l^*,i^*}^{(k)}]}{\sum\limits_{k \in \mathcal{W}_{l^*,i^*}} 1} & \text{(A.1)} \\ \delta_{l^*}^{2(t+1)} = \dfrac{\sum\limits_{(k,i) \in \mathcal{P}_{l^*}} \mathbf{E}_{\mathbf{Z}|\mathbf{D},\mathbf{\Theta}^{(t)}}[(s_{l^*,i}^{(k)} - s_{l^*,i}^{(t+1)})^2]}{\sum\limits_{(k,i) \in \mathcal{P}_{l^*}} 1} & \text{(A.2)} \\ \theta_{m^*}^{(t+1)} \propto \sum\limits_{l} \mathbf{E}_{\mathbf{Z}|\mathbf{D},\mathbf{\Theta}^{(t)}}[\mathbf{1}(m_l = m^*)] & \text{(A.3)} \end{cases}$$

where $\mathcal{W}_{l^*,i^*}$ denotes the set of workers who have judged d_{i^*} for q_{l^*}, and $\mathcal{P}_{l^*}$ denotes the set of $\langle$worker, document$\rangle$ pairs involved in the annotation for q_{l^*}, i.e.,

$$\mathcal{W}_{l^*,i^*} = \{k \mid \exists i,\ \langle k, l^*, i^*, i\rangle \in \mathbf{D} \text{ or } \langle k, l^*, i, i^*\rangle \in \mathbf{D}\}$$
$$\mathcal{P}_{l^*} = \{(k,i) \mid \exists \tilde{i},\ \langle k, l^*, i, \tilde{i}\rangle \in \mathbf{D} \text{ or } \langle k, l^*, \tilde{i}, i\rangle \in \mathbf{D}\}$$

Unfortunately, $\{\tau_{k,m}^{(t+1)}\}$ do not have a closed-form analytic solution, where we employ Newton's method. The partial derivatives *w.r.t.* $\tau_{k^*,m^*}^{(t+1)}$ are given by:

$$\frac{\partial \mathcal{Q}}{\partial \tau_{k^*,m^*}^{(t+1)}} = \sum_{\substack{l,i_1,i_2 \\ \langle k^*,l,i_1,i_2\rangle \in \mathbf{D}}} \mathbf{E}_{\mathbf{Z}|\mathbf{D},\mathbf{\Theta}^{(t)}} \Bigg[\mathbf{1}(m_l = m^*) \frac{s_{l,i_1}^{(k^*)} - s_{l,i_2}^{(k^*)}}{\sqrt{2}} f\Big(- \frac{\tau_{k^*,m^*}^{(t+1)}}{\sqrt{2}} (s_{l,i_1}^{(k^*)} - s_{l,i_2}^{(k^*)})\Big) \Bigg] \tag{A.4}$$

$$\frac{\partial^2 \mathcal{Q}}{\partial {\tau_{k^*,m^*}^{(t+1)}}^2} = - \sum_{\substack{l,i_1,i_2 \\ \langle k^*,l,i_1,i_2\rangle \in \mathbf{D}}} \mathbf{E}_{\mathbf{Z}|\mathbf{D},\mathbf{\Theta}^{(t)}} \Bigg[\mathbf{1}(m_l = m^*) \frac{(s_{l,i_1}^{(k^*)} - s_{l,i_2}^{(k^*)})^2}{2} \Big\{ f\Big(- \frac{\tau_{k^*,m^*}^{(t+1)}}{\sqrt{2}} (s_{l,i_1}^{(k^*)} - s_{l,i_2}^{(k^*)})\Big) \frac{\tau_{k^*,m^*}^{(t+1)}}{\sqrt{2}} (s_{l,i_1}^{(k^*)} - s_{l,i_2}^{(k^*)}) + f^2\Big(- \frac{\tau_{k^*,m^*}^{(t+1)}}{\sqrt{2}} (s_{l,i_1}^{(k^*)} - s_{l,i_2}^{(k^*)})\Big) \Big\} \Bigg] \tag{A.5}$$

where we used the fact that

$$\partial \mathrm{Q}(x)/\partial x = -\mathrm{P}_{\mathtt{N}}(x|0,1)$$

And $f(\cdot)$ is defined as

$$f(x) = \frac{\mathrm{P}_{\mathtt{N}}(x|0,1)}{\mathrm{Q}(x)} \tag{A.6}$$

whose derivative is calculated as:

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = -xf(x) + f^2(x) \tag{A.7}$$

An important implementation issue of Newton's method is numeric stability. For large $x > 0$, computing $f(x)$ using Equation (A.6) is not advised as both $\mathrm{P}_{\mathtt{N}}(x|0,1)$ and $\mathrm{Q}(x)$ approach zero fast. To address this issue, we use the following approximation (Chiani et al., 2003):

$$\mathrm{Q}(x) \approx \frac{1}{12} e^{-\frac{x^2}{2}} + \frac{1}{4} e^{-\frac{2}{3}x^2} \tag{A.8}$$

Using this result, $f(x) \approx \frac{12}{\sqrt{2\pi}}$ can be found to be a good approximation for $x > 8$.

A.2 Inference of TRM

TRM is the building block of the proposed TPP and is investigated as a baseline in the experiment. We present the maximum likelihood estimation (MLE) using the Expectation-Maximization (E-M) algorithm.

The joint likelihood is given by

$$\begin{aligned}&\mathrm{P}\big(\{\sigma_l^{(k)}\},\{s_{l,i}^{(k)}\}|\{s_{l,i}\},\{\delta_l^2\}\big)\\=&\prod_{l,i,k}\mathrm{P}_N(s_{l,i}^{(k)}|s_{l,i},\delta_l^2)\cdot\mathbb{1}(\sigma_l^{(k)},\{s_{l,i}^{(k)}\})\end{aligned}\tag{A.9}$$

where $\mathbb{1}(\sigma_l^{(k)},\{s_{l,i}^{(k)}\})=1$ if the ranking derived from the order of $\{s_{l,i}^{(k)}\}$ is consistent with $\sigma_l^{(k)}$ and 0 otherwise. In addition, like TPP, the posterior distribution is approximated by Gibbs sampling,

$$\begin{aligned}&\mathrm{P}\big(s_{l^*,i^*}^{(k^*)}|\{s_{l,i}\},\{\delta_l^2\},\{\sigma_l^{(k)}\},\{s_{l,i}^{(k)}\}\ \{s_{l^*,i^*}^{(k^*)}\}\big)\\=&\mathrm{P}_N(s_{l^*,i^*}^{(k^*)}|s_{l^*,i^*},\delta_{l^*}^2)\cdot\mathbb{1}(s_-\le s_{l^*,i^*}^{(k^*)}\le s_+\})\end{aligned}\tag{A.10}$$

where s_+ (or s_-) denotes the worker's perceived score $s_{l^*,i}^{(k^*)}$ of the document d_i which immediately precedes (or follows) d_{i^*} as ranked by $\sigma_{l^*}^{(k^*)}$ if such d_i exists or otherwise evaluated as $+\infty$ (or $-\infty$). Consequently, we samples $s_{l^*,i^*}^{(k^*)}$ by

$$s_{l^*,i^*}^{(k^*)}\sim\mathrm{TN}_{s_-}^{s_+}(s_{l^*,i^*},\delta_{l^*}^2)$$

Lastly, we update the parameters by optimizing the expected joint log likelihood, which yields the same updating rules as in Equation (A.1), Equation (A.2) with the only difference being that k is ranged over all workers that rank for q_{l^*} in Equation (A.1) and (k,i) over all workers that judge q_{l^*} and documents in the ranking list of q_{l^*} in Equation (A.2).

A final note of TRM is about its identifiability: It requires rescaling in the same manner as in Equation (4.16) and Equation (4.17) to cancel extra freedom in order to prevent the model from undesired drifting and scaling.