Chapter 5

DIMENSIONALITY REDUCTION AND TOPIC MODELING: FROM LATENT SEMANTIC INDEXING TO LATENT DIRICHLET ALLOCATION AND BEYOND

Steven P. Crain
School of Computational Science and Engineering
College of Computing
Georgia Institute of Technology
s.crain@gatech.edu

Ke Zhou
School of Computational Science and Engineering
College of Computing
Georgia Institute of Technology
kzhou@gatech.edu

Shuang-Hong Yang
School of Computational Science and Engineering
College of Computing
Georgia Institute of Technology
shy@gatech.edu

Hongyuan Zha
School of Computational Science and Engineering
College of Computing
Georgia Institute of Technology
zha@cc.gatech.edu

Abstract The bag-of-words representation commonly used in text analysis can be analyzed very efficiently and retains a great deal of useful information, but it is also troublesome because the same thought can be expressed using many different terms or one term can have very different meanings. Dimension reduction can collapse together terms that have the same semantics, to identify and disambiguate terms with multiple meanings and to provide a lower-dimensional representation of documents that reflects concepts instead of raw terms. In this chapter, we survey two influential forms of dimension reduction. Latent semantic indexing uses spectral decomposition to identify a lower-dimensional representation that maintains semantic properties of the documents. Topic modeling, including probabilistic latent semantic indexing and latent Dirichlet allocation, is a form of dimension reduction that uses a probabilistic model to find the co-occurrence patterns of terms that correspond to semantic topics in a collection of documents. We describe the basic technologies in detail and expose the underlying mechanism. We also discuss recent advances that have made it possible to apply these techniques to very large and evolving text collections and to incorporate network structure or other contextual information.

Keywords: Dimension reduction, Latent semantic indexing, Topic modeling, Latent Dirichlet allocation.

1. Introduction

In 1958, Lisowsky completed an index of the Hebrew scriptures to help scholars identify the meanings of terms that had long since become unfamiliar [42]. Through a tedious manual process, he collected together all of the contexts in which every term occurred. As he did this, he needed to suppress differences in word form that were not significant while preserving differences that might affect the semantics. He hoped by this undertaking to enable other researchers to analyze the different passages and understand the semantics of each term in context.

The core task of automated text mining shares many of the same challenges that Lisowsky faced. The same concept can be expressed using any number of different terms (*synonymy*) and conversely the apparently same term can have very different meanings in different contexts (*polysemy*). Automated text mining must leverage clues from the context to identify different ways of expressing the same concept and to identify and disambiguate terms that are polysemous. It must also present the data in a form that enables human analysts to identify the semantics involved when they are not known *a priori*.

It is common to represent documents as a *bag of words* (BOW), accounting for the number of occurrences of each term but ignoring the order. This representation balances computational efficiency with the need to retain the document content. It also results in a vector representation that can be analyzed with techniques from applied mathematics and machine learning, notably dimension reduction, a technique that is used to identify a lower-dimensional representation of a set of vectors that preserves important properties.

BOW vectors have a very high dimensionality — each dimension corresponding to one term from the language. However, for the task of analyzing the concepts present in documents, a lower-dimensional semantic space is ideal — each dimension corresponding to one concept or one topic. Dimension reduction can be applied to find the semantic space and its relationship to the BOW representation. The new representation in semantic space reveals the topical structure of the corpus more clearly than the original representation.

Two of the many dimension reduction techniques that have been applied to text mining stand out. *Latent semantic indexing*, discussed in Section 2, uses a standard matrix factorization technique (singular vector decomposition) to find a latent semantic space. *Topic models*, on the other hand, provide a probabilistic framework for the dimension reduction task. We describe topic modeling in Section 3, including probabilistic latent semantic indexing (PLSI) and latent Dirichlet allocation (LDA). In Section 4, we describe the techniques that are used to interpret and evaluate the latent semantic space that results from dimension reduction. Many recent advances have made it possible to apply dimension reduction and topic modeling to large and dynamic datasets. Other advances incorporate network structures like social networks or other contextual information. We highlight these extensions in Section 5 before concluding in Section 6.

1.1 The Relationship Between Clustering, Dimension Reduction and Topic Modeling

Clustering, dimension reduction and topic modeling have interesting relationships. For text mining, these techniques represent documents in a new way that reveals their internal structure and interrelations, yet there are subtle distinctions. *Clustering* uses information on the similarity (or dissimilarity) between documents to place documents into natural groupings, so that similar documents are in the same cluster. *Soft clustering* associates each document with multiple clusters. By viewing each cluster as a dimension, clustering induces a low-dimensional representation for documents. However, it is often difficult to characterize a cluster

in terms of meaningful features because the clustering is independent of the document representation, given the computed similarity.

On the other hand, dimension reduction starts with a feature representation of documents (typically a BOW model) and looks for a lower-dimensional representation that is faithful to the original representation. Although this close coupling with the original features results in a more coherent representation that maintains more of the original information than clustering, interpretation of the compressed dimensions is still difficult. Specifically, each new dimension is usually a function of all the original features, so that generally a document can only be fully understood by considering all of the dimensions together.

Topic modeling essentially integrates soft clustering with dimension reduction. Documents are associated with a number of latent topics, which correspond to both document clusters and compact representations identified from a corpus. Each document is assigned to the topics with different weights, which specify both the degree of membership in the clusters as well as the coordinates of the document in the reduced dimension space. The original feature representation plays a key role in defining the topics and in identifying which topics are present in each document. The result is an understandable representation of documents that is useful for analyzing the themes in documents.

1.2 Notation and Concepts

Documents. We use the following notation to consistently describe the documents used for training or evaluation. D is a corpus of M documents, indexed by d. There are W distinct terms in the vocabulary, indexed by v. The term-document matrix X is a $W \times M$ matrix encoding the occurrences of each term in each document. The LDA model has K topics, indexed by i. The number of tokens in any set is given by N, with a subscript to specify the set. For example, N_i is the number of tokens assigned to topic i. A bar indicates set complement, as for example $\bar{z}_{dn} \equiv \{z_{d'n'} : d' \neq d \text{ or } n' \neq n\}$.

Multinomial distribution. A commonly used probabilistic model for texts is the multinomial distribution,

$$\mathcal{M}(\boldsymbol{X}|\boldsymbol{\Psi}) \propto \prod_{v=1}^{W} \psi_v^{x_v},$$

which captures the relative frequency of terms in a document and is essentially equivalent to the BOW-vector with ℓ_1-norm standardization as $\sum_{v=1}^{W} \psi_v = 1$.

Dirichlet distribution. Dirichlet distribution is the conjugate distribution to multinomial distribution and therefore commonly used as prior for multinomial models:

$$\mathcal{D}(\boldsymbol{\Psi}|\boldsymbol{\Xi}) = \frac{\Gamma\left(\sum_{i=1}^{K} \xi_i\right)}{\prod_{i=1}^{K} \Gamma(\xi_i)} \prod_{i=1}^{K} \psi_i^{\xi_i - 1}.$$

This distributions favors imbalanced multinomial distributions, where most of the probability mass is concentrated on a small number of values. As a result, it is well suited for models that reflect commonly observed power law distributions in human language.

Generative process. A generative process is an algorithm describing how an outcome was selected. For example, one could describe the generative process of rolling a die: one side is selected from a multinomial distribution with 1/6 probability on each of the six sides. For topic modeling, a random generative process is valuable even though choosing the terms in a document is not random, because they capture real statistical correlations between topics and terms.

2. Latent Semantic Indexing

LSI is an automatic indexing method that projects both documents and terms into a low dimensional space which, by intent, represents the semantic concepts in the document. By projecting documents into the semantic space, LSI enables the analysis of documents at a *conceptual* level, purportedly overcoming the drawbacks of purely term-based analysis. For example, in information retrieval, users may use many different queries to describe the same information need, and likewise, many of the relevant documents may not contain the exact terms used in the particular query. In this case, projecting documents into the semantic space enables the search engine to find documents containing the same concepts but different terms. The projection also helps to resolve terms that are associated with multiple concepts. In this sense, LSI overcomes the issues of *synonymy* and *polysemy* that plague term-based information retrieval.

LSI was applied to text data in the 1980s and later used for indexing in information retrieval systems [23]. It has also been used for a variety of tasks, including assigning papers to reviewers [28] and cross-lingual retrieval.

LSI is based on the singular value decomposition (SVD) of the term-document matrix, which constructs a low rank approximation of the original matrix while preserving the similarity between the documents. LSI is meant to interpret the dimensions of the low-rank approximation as semantic concepts although it is surpassed in this regard by later improvements such as PLSI. We now describe the basic steps for performing LSI. Then, we will discuss the implementation issues and analyze the underlying mechanisms for LSI.

2.1 The Procedure of Latent Semantic Indexing

Given the term-document matrix X of a corpus, the d-th column $\boldsymbol{X}_d$ represents a document d in the corpus and the v-th row of the matrix X, denoted by $\boldsymbol{T}_v$, represents a term v. Several possibilities for the encoding are discussed in the implementation issues section.

Let the singular value decomposition of X be

$$X = U\Sigma V^T,$$

where the matrices U and V are orthonormal and Σ is diagonal—

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{\min\{W,M\}} \end{bmatrix}.$$

The values $\sigma_1, \sigma_2, \ldots, \sigma_{\min\{W,M\}}$ are the singular values of the matrix X. Without loss of generality, we assume that the singular values are arranged in descending order, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{W,M\}}$.

For dimension reduction, we approximate the term-document matrix X by a rank-K approximation $\hat{X}$. This is done with a partial SVD using the singular vectors corresponding to the K largest singular values.

$$\begin{aligned} \hat{X} &= \hat{U}\hat{\Sigma}\hat{V}^T \\ &= [\boldsymbol{U}_1 \quad \ldots \quad \boldsymbol{U}_K] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_K \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_1^T \\ \vdots \\ \boldsymbol{V}_K^T \end{bmatrix}. \end{aligned} \tag{5.1}$$

SVD produces the rank-K matrix $\hat{X}$ that minimizes the distance from X in terms of the spectral norm and the Frobenius norm. Although X is typically sparse, $\hat{X}$ is generally not sparse. Thus, $\hat{X}$ can be viewed as a smoothed version of X, obtained by propagating the co-occurring terms in the document corpus. This smoothing effect is achieved by discovering a latent semantic space formed by the documents. Specifically, we can

observe from Eqn. (5.1) that each document d can be represented by a K-dimensional vector $\hat{\boldsymbol{X}}_d$, which is the d-th row of the matrix $\hat{V}$. The relation between the representation of document d in term space $\boldsymbol{X}_d$ and the latent semantic space $\hat{\boldsymbol{X}}_d$ is given by

$$\boldsymbol{X}_d = \hat{U}\hat{\Sigma}\hat{\boldsymbol{X}}_d.$$

Similarly, each term v can be represented by the K-dimensional vector $\hat{\boldsymbol{T}}_v$ given by

$$\boldsymbol{T}_v = \hat{V}\hat{\Sigma}\hat{\boldsymbol{T}}_v.$$

Thus, LSI projects both terms and documents into a K-dimensional latent semantic space. We can utilize these projections into latent semantic space to perform several tasks.

Information retrieval. In information retrieval, we are given a query $\boldsymbol{q}$ which contains several key terms that describe the information need. The goal is to return documents that are related to the query. In this case, we can view the query as a short document and project it into the latent semantic space using

$$\hat{\boldsymbol{q}} = \hat{\Sigma}^{-1}\hat{U}^T\boldsymbol{q}.$$

Then, the similarity between the query and document can be measured in the latent semantic space. For example, we can use the inner product $\hat{\boldsymbol{V}}_d^T\hat{\boldsymbol{q}}$. By using the smoothed latent semantic space for the comparison, we mitigate the problems with synonymy and polysemy.

Document similarity. The similarity between document d and d' can be measured using their representations in the latent semantic space, for example, using the inner product of $\hat{\boldsymbol{X}}_d$ and $\hat{\boldsymbol{X}}_{d'}$. This can be used to cluster or classify documents. Additional regularization may be necessary to resolve the non-identifiability of the SVD [63].

Term similarity. Analogous to the document similarity, term similarities can be measured in the latent semantic space, so as to identify terms with similar meanings.

2.2 Implementation Issues

2.2.1 Term-Document Matrix Representation. LSI utilizes the term-document matrix X for a document corpus, which represents the occurrences of terms in documents. In practice, the term-document matrix can be constructed in several ways. For example, each

entry x_{vd} can represent the number of times that the term v occurs in document d. However, Zipf's law shows that real documents tend to be *bursty*—a globally uncommon term is likely to occur multiple times in a document if it occurs at all [19]. As a result, simply using the term frequency tends to exaggerate the contribution of the term. This problem can be directly addressed by using a binary representation, which only indicates whether a term occurs in a particular document and ignores its frequency. Global term-weight methods, such as term frequency weighted with inverse document frequency (IDF) [44], provide a good compromise for most document corpora. Besides these BOW representations, the language pyramid model [70] provides a multi-resolution matrix representation for documents, encoding not only the semantic information of term occurrence but also the spatial information such as term proximity, ordering, long distance dependence and so on.

2.2.2 Computation. LSI relies on a partial SVD of the term-document matrix, which can be computed using the Lanczos algorithm [7, 30, 73]. The Lanczos algorithm is an iterative algorithm that computes the eigenvalues and eigenvectors of a large and sparse matrix X using the matrix vector multiplication. This process can be accelerated by exploiting any special structure of the term-document matrix. For example, Zha and Zhang [75] provide an efficient algorithm when the matrix has a low-rank-plus-shift structure, which arises when regularization is added. Numerous implementations that use the Lanczos algorithm are available, including SVDPACK (`http://www.netlib.org/svdpack`).

2.2.3 Handling Changes. In real world applications, the corpus often changes rapidly. As a result, it is impractical to apply LSI to the corpus every time a document is added, removed or changed. There are two strategies for efficiently handling these changes.

Fold-in. One method for updating LSI is called *fold-in*, where we compute the projection of the new documents and terms into the latent semantic space based on the projection for original documents and terms. In order to fold in a document represented by vector $\boldsymbol{d} \in \mathbb{R}^W$ into a existing latent semantic indexing, we can project the document into the latent semantic space based on the SVD decomposition obtained from the original corpus.

$$\hat{\boldsymbol{d}} = \hat{\Sigma}^{-1}\hat{U}^T\boldsymbol{d}.$$

Fold-in is very efficient because the SVD does not need to be recomputed. Because the term vector $\boldsymbol{d}$ is typically sparse, the fold-in process can be computed in $O(KN)$ time, where N is the number of unique terms in $\boldsymbol{d}$.

Updating the semantic space. Although the fold-in process is efficient and maintains a consistent indexing, there is no longer any guarantee that the indexing provides the best rank-K approximation of the modified corpus. Over time, the outdated model becomes increasingly less useful. Several methods for updating the LSI model have been proposed that are both efficient and accurate [8, 52, 74]. For example, Zha and Simon [74] provide an updating algorithm based on performing LSI on $[\hat{X}\ X']$ instead of $[X\ X']$, where X' is the term-document matrix for new documents. Specifically, the low-rank approximate $\hat{X}$ is used to replace the document-term matrix X of the original corpus. Assume that the QR decomposition of the matrix $(I - \hat{U}\hat{U}^T)X'$ is

$$(I - \hat{U}\hat{U})X' = U'R,$$

where R is a triangular matrix and $\hat{X} = \hat{U}\hat{\Sigma}\hat{V}$ is the partial SVD of the matrix X. Then we have

$$[X\ X'] = [\hat{U}\ U']\begin{bmatrix}\hat{\Sigma} & \hat{U}^TX\\ 0 & R\end{bmatrix}\begin{bmatrix}\hat{V}^T & 0\\ 0 & I\end{bmatrix}.$$

Now we can compute the best rank-K approximation of $\begin{bmatrix}\hat{\Sigma} & \hat{U}^TX\\ 0 & R\end{bmatrix}$ by SVD:

$$\begin{bmatrix}\hat{\Sigma} & \hat{U}^TX\\ 0 & R\end{bmatrix} = \hat{P}\Sigma'\hat{Q}^T.$$

Then, the partial SVD for $[\hat{X}\ X']$ can be expressed as

$$\left([\hat{U}\ U']\hat{P}\right)\Sigma'\left(\begin{bmatrix}\hat{V} & 0\\ 0 & I\end{bmatrix}\hat{Q}\right)^T,$$

which provides an approximation of the partial SVD for $[X\ X']$. A theoretical analysis by Zha and Simon shows that this approximation will not introduce unacceptable errors into LSI [74].

2.3 Analysis

Due to the popularity of LSI, there has been considerable research into the underlying mechanism of LSI.

Term context. LSI improves the performance of information retrieval by discovering the latent concepts in a document corpus and thus solving the problems of synonymy and polysemy. Bast and Majumdar [5] demonstrate this point by considering the projections of a query $\boldsymbol{q}$

and document $\boldsymbol{d}$ into the latent semantic space by the mapping

$$\boldsymbol{f}(\boldsymbol{x}) = \hat{U}^T \boldsymbol{x}\,.$$

The cosine similarity of the query and document in the latent semantic space is

$$S_{qd} = \frac{\boldsymbol{q}^T \hat{U}\hat{U}^T \boldsymbol{d}}{\|\hat{U}^T \boldsymbol{q}\|\|\hat{U}^T \boldsymbol{d}\|}\,.$$

Since the factor $\|\hat{U}^T \boldsymbol{q}\|$ does not depend on documents, it can be neglected without affecting the ranking. Note that

$$\|\hat{U}^T \boldsymbol{d}\| = \|\hat{U}\hat{U}^T \boldsymbol{d}\|\,,$$

so the cosine similarity S_{qd} can expressed by

$$S_{qd} = \frac{\boldsymbol{q}^T \hat{U}\hat{U}^T \boldsymbol{d}}{\|\hat{U}\hat{U}^T \boldsymbol{d}\|} = \frac{\boldsymbol{q}^T T \boldsymbol{d}}{\|T\boldsymbol{d}\|},$$

where $T \equiv \hat{U}\hat{U}^T$.

The similarity S_{qd} between query q and document d can be expressed by the cosine similarity of query $\boldsymbol{q}$ and the transformed document $T\boldsymbol{d}$. In term-based information retrieval, the transformation $T = I$, so the original document is used to calculate the similarity. In LSI, however, the transformation T is not the identity matrix. Intuitively, the entry $t_{vv'}$ represents the relationship between terms v and v'. Specifically, the occurrence of term v in document has an equivalent impact on the similarity to $t_{vv'}$ times the occurrence of term v in the same document. In this sense, LSI enriches the document by introducing similar terms that may not occur in the original document.

Bast and Majumdar [5] also analyze LSI from the view of identifying terms that appear in similar contexts in the documents. Consider the sequence of the similarities between a pair of terms with respect to the dimension of latent semantic space, $K_{vv'}(k) = \sum_{i=1}^{k} \boldsymbol{U}_v^{(i)} \boldsymbol{U}_{v'}^{(i)T}$, where $U^{(i)}$ is from the rank-i partial SVD. The trend of the sequence can be categorized into three different types: increasing steadily (A); first increasing and then decreasing (B); or, no clear trend (C). If terms v and v' are related, the sequence is usually of Type A or B. Otherwise, the sequence is of Type C. This result is closely related to global special structures in the term-document matrix X that arise from similar contexts for similar terms. Thus, the sequence $K_{vv'}$ of similar terms have the specific shapes described above.

Since LSI captures the contexts of terms in documents, it is able to deal with the problems of synonymy and polysemy: synonymy can be captured since terms with the same meaning usually occur in similar context; polysemy can be addressed since terms with different meaning can be distinguished by their occurrences in different context. Landauer [40] also provides intuition for LSI by showing that it captures several important aspects of human languages.

Dimension of the latent semantic space. Dupert [29] studies how to determine the optimal number of latent factors for finding the most similar terms of a query. In particular, he shows how LSI can deal with the problem of synonymy in the context of Correlation method. He also provides an upper bound for the dimension of latent semantic space in order to present the corpus correctly.

Probabilistic analysis. Kubato Ando and Lee [38] explore the relationship between the performance of LSI and the uniformity of the underlying distribution. When the topic-documents distribution is quite uniform, LSI can recover the optimal representation precisely. Papadimitriou et al. [53] and Ding [26] analyze LSI from a probabilistic perspective which is related to probabilistic latent semantic indexing [36], which we discuss next.

3. Topic Models and Dimension Reduction

Taboo® (a registered trademark of Hasbro) is a game where one player must help a teammate guess a word from a game card without using any of the taboo words listed on the card. The surprising difficulty of the game highlights that certain terms are very likely to be present based on the topic of a document. *Latent topic models* capture this idea by modeling the conditional probability that an author will use a term given the topic the author is writing about.

LSI reduced the dimensionality of documents by projecting the BOW vectors into a semantic space constructed from the SVD of the term-document matrix. By providing a mechanism to explicitly reason about latent topics, probabilistic topic models can achieve a similar yet more meaningful latent semantic space. The results are presented in familiar probabilistic terms, and thus can be directly incorporated into other probabilistic models and analyzed with standard statistical techniques. Moreover, Bayesian methods can be used to make the models robust to parameter selection. Finally, one of the most useful advantages is that the models can be easily extended by modifying the structure to solve interesting related problems.

3.1 Probabilistic Latent Semantic Indexing

PLSI, proposed by Hofmann [36], provides a crucial step in topic modeling by extending LSI in a probabilistic context. PLSI has seen widespread use in text document retrieval, clustering and related areas; it builds on the same conceptual assumptions as LSI, but uses a radically different *probabilistic* generative process for generating the terms in the documents of a text corpus.

PLSI is based on the following generative process for (w, d), a word w in document d:

- Sample a document d from multinomial distribution $p(d)$.
- Sample a topic $i \in \{1, \ldots, K\}$ based on the topic distribution $\theta_{di} = p(z = i|d)$.
- Sample a term v for token w based on $\Phi_{iv} = p(w = v|z = i)$.

In other words, an unobservable topic variable z is associated with each observation (v, d) in PLSI. The joint probability distribution for $p(v, d)$ can be expressed as

$$p(v, d) = p(d)p(v|d), \qquad \text{where} \qquad p(v|d) = \sum_{i=1}^{K} p(v|z = i)p(z = i|d).$$

This equation has the geometric interpretation that the distribution of terms conditioned on documents $p(z = i|d)$ is a convex combination of the topic-specific term distributions $p(v|z = i)$.

Connection to LSI. An alternative way to express the joint probability is given by

$$p(v, d) = \sum_{i=1}^{K} p(z = i)p(d|z = i)p(v|z = i).$$

This formulation is sometimes called the *symmetric formulation* because it models the documents and terms in a symmetric manner. This formulation has a nice connection to LSI: the probability distributions $p(d|z = i)$ and $p(w|z = i)$ can be viewed as the projections of documents and terms into the latent semantic spaces, just like the matrices $\hat{V}$ and $\hat{U}$ in LSI. Also, the distribution $p(z = i)$ is similar to the diagonal matrix $\hat{\Sigma}$ in LSI. This is the sense in which PLSI is a probabilistic version of LSI.

3.1.1 Algorithms. The maximal likelihood method is used to estimate the parameters $p(d)$, $p(z|d)$ and $p(v|z)$. Given the term-document matrix X, the log-likelihood of observed data can be expressed as

$$\begin{aligned}\mathcal{L} &= \sum_{d=1}^{M}\sum_{v=1}^{W} x_{vd} \log p(w = v, d) \\ &= \sum_{d=1}^{M}\sum_{v=1}^{W} x_{vd} \log \sum_{i=1}^{K} p(w = v|z = i)p(z = i|d)p(d). \end{aligned} \tag{5.2}$$

Maximizing the log-likelihood function is equivalent to minimizing the Kullback-Leibler divergence (KL) [39] between the measured empirical distribution $\hat{p}(v|d)$ and the model distribution $p(w|d) = \sum_{i=1}^{K} p(w|z = i)p(z = i|d)$. Since this is non-convex, expectation-maximization (EM) [24] is used to seek a locally optimal solution. The log-likelihood value (Eqn. (5.2)) increases on each iteration and converges to a local maximum.

Expectation. The E-step computes the posterior of the latent variable z based on the current estimation of the parameters.

$$p'(z = i|d, v) = \frac{p(d)p(z = i|d)p(v|z = i)}{\sum_{i'=1}^{K} p(d)p(z = i'|d)p(v|z = i')},$$

where the prime on p indicates the new estimate of the probability for the next step.

Maximization. The M-step updates the parameters once the latent variables are known using the posterior estimated in the previous E-step:

$$\begin{aligned} p'(w = v|z) &\propto \sum_{d=1}^{M} x_{vd}\, p'(z = i|d, w = v); \\ p'(z = i|d) &\propto \sum_{v=1}^{W} x_{vd}\, p'(z = i|d, w = v); \\ p'(d) &\propto \sum_{v=1}^{W} x_{vd}. \end{aligned}$$

3.1.2 Updating. Given a new document d, the fold-in process can be applied to obtain its representation in the latent semantic space, much like for LSI. Specifically, an EM algorithm similar to parameter estimation can be used to obtain $p(z|d)$ [37]. $p(w|z)$ and $p(z)$ are not updated in the M-step during fold-in.

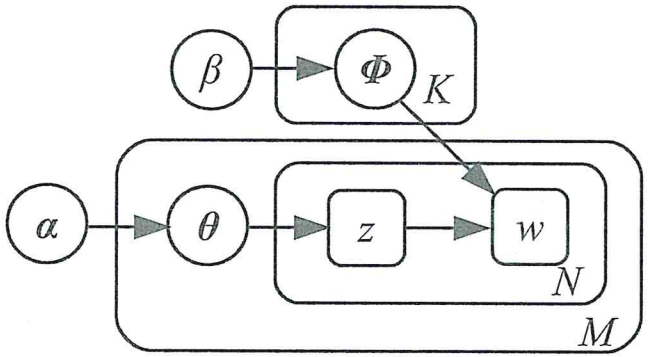

Figure 5.1. Diagram of the LDA graphical model

3.2 Latent Dirichlet Allocation

PLSI provides a good basis for text analysis, but it has two problems. First, it contains a large number of parameters that grows linearly with the number of documents so that it tends to overfit the training data. Second, there is no natural way to compute the probability of a document that was not in the training data. LDA includes a process for generating the topics in each document, thus greatly reducing the number of parameters to be learned and providing a clearly-defined probability for arbitrary documents. Because LDA has a rich generative model, it is also readily adapted to specific application requirements, which we describe in Section 5.

3.2.1 Model. Like PLSI, LDA is based on a hypothetical generative process for a corpus. A diagram of the graphical model showing how the different random variables are related is shown in Fig. 5.1. In the diagram, each random variable is represented by a circle (continuous) or square (discrete). A variable that is *observed* (its outcome is known) is shaded. An arrow is drawn from one random variable to another if the the outcome of the second variable depends on the value of the first variable. A rectangular plate is drawn around a set of variables to show that the set is repeated multiple times, as for example for each document or each token.

- CHOOSE THE TERM PROBABILITIES FOR EACH TOPIC. The distribution of terms for each topic i is represented as a multinomial distribution $\mathbf{\Phi}_i$, which is drawn from a symmetric Dirichlet distribution with parameter β.

$$\mathbf{\Phi}_i \sim \mathcal{D}(\beta)\,; \qquad p(\mathbf{\Phi}_i|\beta) = \frac{\Gamma(W\beta)}{[\Gamma(\beta)]^W} \prod_{v=1}^{W} \phi_{iv}^{\beta-1}\,.$$

- CHOOSE THE TOPICS OF THE DOCUMENT. The topic distribution for document d is represented as a multinomial distribution $\boldsymbol{\theta}_d$, which is drawn from a Dirichlet distribution with parameters $\boldsymbol{\alpha}$. The Dirichlet distribution captures the document-independent popularity and the within-document burstiness of each topic.

$$\boldsymbol{\theta}_d \sim \mathcal{D}(\boldsymbol{\alpha})\,; \qquad p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_{di}^{\alpha_i - 1}\,.$$

- CHOOSE THE TOPIC OF EACH TOKEN. The topic z_{dn} for each token index n is chosen from the document topic distribution.

$$z_{dn} \sim \mathcal{M}(\boldsymbol{\theta}_d)\,; \qquad p(z_{dn} = i|\boldsymbol{\theta}_d) = \theta_{di}\,.$$

- CHOOSE EACH TOKEN. Each token w at each index is chosen from the multinomial distribution associated with the selected topic.

$$w_{dn} \sim \mathcal{M}(\boldsymbol{\phi}_{z_{dn}})\,; \qquad p(w_{dn} = v|z_{dn} = i, \boldsymbol{\phi}_i) = \phi_{iv}\,.$$

Mechanism. LDA provides the mechanism for finding patterns of term co-occurrence and using those patterns to identify coherent topics. Suppose that we have used LDA to learn a topic i and that for term v, $p(w = v|z = i)$ is high. As a result of the LDA generative process, any document d that contains term v has an elevated probability for topic i, that is, $p(z_{dn'} = i|w_{dn} = v) > p(z_{dn'} = i)$. This in turn means that all terms that co-occur with term v are more likely to have been generated by topic i, especially as the number of co-occurrences increases. Thus, LDA results in topics in which the terms that are most probable frequently co-occur with each other in documents.

Moreover, LDA also helps with polysemy. Consider a term v with two distinct meanings in topics i and i'. Considering only this term, the model places equal probability on topics i and i'. However, if the other words in the context place a 90% probability on i and only a 9% probability on i', then LDA will be able to use the context to disambiguate the topic: it is topic i with 90% probability.

Wallach et al. [60] show that the symmetry or asymmetry of the Dirichlet priors strongly influences the mechanism. For the topic-specific term distributions, a symmetric Dirichlet prior provides smoothing so that unseen terms will have non-zero probability. However, an asymmetric prior would equally affect all topics, making them less distinctive.

In contrast, they showed that an asymmetric prior for the document-specific topic distributions made LDA more robust to stop words and less sensitive to the selection of the number of topics. The stop words were mainly relegated to a small number of highly probable topics that influence most documents uniformly. The asymmetric prior also results in more stable topics, which means that additional topics will make small improvements in the model instead of radically altering the topic structure. This is similar to the situation of LSI, where performance is optimal when Σ scales the contribution of each dimension according to its eigenvalue. In the same way, LDA will perform best if α is non-uniform and corresponds to some natural values characteristic of the dataset.

One disadvantage of LDA is that it tends to learn broad topics. Consider the case where a concept has a number of aspects to it. Each of the aspects co-occurs frequently with the main concept, and so LDA will favor a topic that includes the concept and all of its aspects. It will further favor adding other concepts to the same topic if they share the same aspects. As this process continues, the topics become more diffuse. When sharper topics are desired, a hierarchical topic model may be more appropriate.

Likelihood. Training an LDA model involves finding the optimal set of parameters, under which the probability of generating the training documents is maximized. The probability of the training documents under a given LDA model is called the *empirical likelihood* $\mathcal{L}$. It can also be used to identify the optimal model configuration using Bayesian model selection.

$$\mathcal{L} = \prod_{d=1}^{M} \prod_{n=1}^{N} p(w_{dn}|z_{dn}, \mathbf{\Phi}) p(z_{dn}|\boldsymbol{\theta}_d) p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) p(\mathbf{\Phi}|\beta)$$

$$= \phi_{zw}\, \theta_{dz}\, \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_{di}^{\alpha_i - 1} \frac{\Gamma(W\beta)}{[\Gamma(\beta)]^W} \prod_{v=1}^{W} \phi_v^{\beta - 1}.$$

Unfortunately, the direct optimization of the likelihood is problematic because the topic assignments z_{dn} are not directly observed. Even

inference for a single document is intractable. We describe two different approximations for LDA. Collapsed Gibbs sampling samples a value for each z_{dn} in turn, conditioned on the topic assignments for the other tokens. Variational Bayes approximates the model with a series of simpler models that bound the likelihood but neglect the troublesome dependencies.

3.2.2 Collapsed Gibbs. Gibbs sampling is commonly used to estimate the distribution of values for a probability model when exact inference is intractable. First, values are assigned to each variable in the model, either randomly or using a heuristic. Each variable is then sampled in turn, conditioned on the values of the other variables. In the limit of the number of iterations, this process explores all configurations and yields unbiased estimates of the underlying distributions. In practice, Gibbs sampling is implemented by rejecting a large number of samples during an initial burn-in period and then averaging the assignments during an additional large number of samples.

In *collapsed* Gibbs sampling, certain variables are marginalized out of the model. Griffiths and Steyvers [32] propose collapsed Gibbs sampling for LDA, with both $\boldsymbol{\theta}$ and $\mathbf{\Phi}$ marginalized. Only z_{dn} is sampled, and the sampling is done conditioned on $\boldsymbol{\alpha}$, β and the topic assignments of other words $\bar{z}_{dn}$.

$$p(z_{dn}|\bar{z}_{dn}) \propto (N_{dz} + \alpha_z)(N_{zw} + \beta)\,.$$

The N statistics do not include the contribution from the word being sampled, and must be updated after each sampling.

The equation makes intuitive sense. A topic that is used frequently in the document has a higher probability in $\boldsymbol{\theta}$ and so is more likely for the current token also. This characteristic corresponds to the burstiness observed in documents [19]. Similarly, a topic that is frequently assigned for the same term corpus-wide is more likely to be correct here also.

After burn-in, the implementation can keep statistics of the number of times each topic is selected for each word. These statistics can then be aggregated and normalized to estimate the topic distributions for each document or word. To apply a trained model to additional documents, the only change is that the N_{zw} statistic is not updated.

3.2.3 Variational Approximation. Variational approximation provides an alternative algorithm for training an LDA model. We will first consider the case of inferring the topics of a document given an existing LDA model, before we explain how the model is trained. A

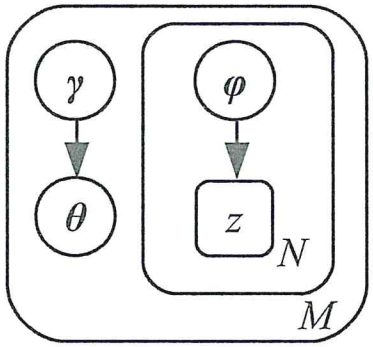

Figure 5.2. Diagram of the LDA variational model

direct approach for topic inference is to apply Bayes' rule:

$$p(\boldsymbol{\theta}|w_d) = \frac{p(\boldsymbol{\theta}, w_d)}{p(w_d)} = \frac{\int_{\boldsymbol{Z}} p(d, \boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\alpha}, \beta) d\boldsymbol{Z}}{\int_{\boldsymbol{Z},\boldsymbol{\theta}} p(d, \boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\alpha}, \beta) d\boldsymbol{Z} d\boldsymbol{\theta}},$$

where $\boldsymbol{Z} = \{z_1, z_2, \ldots, z_N\}$. However, the marginalization in both numerator and denominator is intractable. The *Variational Bayesian* approach provides an approximate solution; instead of inferring the latent variables by directly marginalizing the joint distribution $p(w_d, \boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\alpha}, \beta)$, it uses a much simpler distribution as a proxy and performs the inference through optimization.

Variational inference approximates the true posterior distribution of the latent variables by a fully-factorized distribution—this proxy is usually referred to as the variational model, which assumes all the latent variables are independent of each other. For LDA,

$$q(\boldsymbol{Z}, \boldsymbol{\theta}|\boldsymbol{\gamma}, \boldsymbol{\phi}) = q(\boldsymbol{\theta}|\boldsymbol{\gamma}) \prod_{n=1}^{N} q(z_n|\boldsymbol{\phi}_n) = \mathcal{D}(\boldsymbol{\theta}|\boldsymbol{\gamma}) \prod_{n=1}^{N} \mathcal{M}(z_n|\boldsymbol{\phi}_n).$$

Essentially, this variational distribution is a simplification of the original LDA graphical model by removing the edges between the nodes $\boldsymbol{\theta}$ and $\boldsymbol{Z}$ (Figure 5.2). The optimal approximation is achieved by optimizing the distance (for example, the KL divergence) between the true model and the variational model:

$$\min_{\boldsymbol{\gamma},\boldsymbol{\phi}} \; KL[q(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\gamma}, \boldsymbol{\phi}) \;||\; p(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{\alpha}, \beta)] \,.$$

It can be shown that the above KL-divergence is the discrepancy between the true log-likelihood and its variational lower-bound that is used in the variational EM algorithm (described later in this section) for estimating the LDA hyperparameters $\boldsymbol{\alpha}$ and β.

The optimization has no close-form solution but can be implemented through iterative updates,

$$\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni}, \quad \phi_{ni} \propto \beta_{iw_n} \exp[\Psi(\gamma_i)] \,,$$

where $\Psi(\cdot)$ is the bi-gamma function.

Variational EM for parameter estimation. We can learn a LDA topic model by maximizing the likelihood of the corpus.

$$\begin{aligned} &\max_{\boldsymbol{\alpha},\beta} \sum_{d=1}^{M} \log p(w_d|\boldsymbol{\alpha}, \beta) \\ =&\max_{\boldsymbol{\alpha},\beta} \sum_{d=1}^{M} \log \int_{\boldsymbol{\theta}_d, \boldsymbol{Z}_d} p(w_d, \boldsymbol{\theta}_d, \boldsymbol{Z}_d|\boldsymbol{\alpha}, \beta) d\boldsymbol{\theta}_d d\boldsymbol{Z}_d \,. \end{aligned}$$

Again, it involves intractable computation of the marginal distribution and we therefore resort to variational approximation, which provides a tractable lower bound,

$$\begin{aligned} \mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\phi}) &= \log p(w_d|\boldsymbol{\alpha}, \beta) - KL(q(\boldsymbol{Z}, \boldsymbol{\theta}|\boldsymbol{\gamma}, \boldsymbol{\phi})||p(\boldsymbol{Z}, \boldsymbol{\theta}|\boldsymbol{\alpha}, \beta)) \\ &\leq \log p(w_d|\boldsymbol{\alpha}, \beta) \,, \end{aligned}$$

where $\mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\phi}) = \mathbf{E}_q[\log p(w_d, \boldsymbol{\theta}, \boldsymbol{Z}) - \log q]$ is the variational lower bound for the log-likelihood. The maximum likelihood estimation therefore involves a two-layer optimization,

$$\max_{\boldsymbol{\alpha},\beta} \sum_{d=1}^{M} \max_{\boldsymbol{\gamma}_d, \boldsymbol{\phi}_d} \mathcal{L}(\boldsymbol{\gamma}_d, \boldsymbol{\phi}_d) \,.$$

The inner-loop (the optimization with respect to $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$, referred to as the Variational E-step) goes through the whole corpus and performs variational approximation for each of the documents, which ends up with a tight lower bound for the log-likelihood. Then the M-step updates the model parameters ($\boldsymbol{\alpha}$ and β) by optimizing this lower-bound approximation of the log-likelihood. The E- and M-steps are alternated in an outer loop until convergence.

In the E-step, γ and ϕ are alternately optimized for each document—in practice, 20 iterations is adequate for a good fit. The outer loop may need to be repeated hundreds of times for full convergence. For best results, the likelihood of a separate validation corpus controls early stopping.

3.2.4 Implementations. There have been substantial efforts in developing efficient and effective implementations of LDA, especially for parallel or distributed architectures. In order to provide a quick hands-on experience, we list a few implementations that are open-source or publicly accessible in Table 5.1.

Table 5.1. Publicly-accessible implementations of LDA.

Name	Language	Algorithm	Reference
LDA-C	C	Var. EM	`www.cs.princeton.edu/blei/lda-c`
Mallet	Java	Gibbs	`mallet.cs.umass.edu`
GibbsLDA++	C++	Gibbs	`gibbslda.sourceforge.net`
Gensim	Python	Gibbs	`nlp.fi.muni.cz/projekty/gensim`
Matlab-LDA	Matlab	Gibbs	`psiexp.ss.uci.edu/programs_data`

4. Interpretation and Evaluation

We have looked at three methods for dimension reduction of textual data. These methods have much in common: they identify the relationships of terms and documents to the dimensions of a latent semantic space. Intuitively, the latent dimensions correspond to concepts or topics that are meaningful to the authors. In this section, we discuss how to jump from the mathematical representations to meaningful topics, how to evaluate the resulting models and how to apply them to applications.

4.1 Interpretation

The common way to interpret the topic models that are discovered by dimension reduction is through inspection of the term-topic associations. Typically, practitioners examine the five to twenty terms that are most strongly associated with each topic, and attempt to discern the commonality. For LSI, the terms can be sorted according to the coefficient corresponding to the given feature in the semantic space. For the probabilistic models, the terms are sorted by the probability of generating the term conditioned on the topic. This approach was popularized following Blei et al. [13], and is generally used to report qualitative topic model results even though it has many disadvantages. The chief problem is

that the top terms are often dominated by globally probable terms that may not be representative of the topic. Stop word removal and variations on IDF weighting both help substantially, but the characterization is sensitive to the precise method used to order terms. Mei et al. [47] provide an alternative approach that automatically selects a portion of a document to use as a label for each topic. Buntine and Jakulin [16] provide a more general framework for interpreting topic models.

4.2 Evaluation

There are three main approaches to evaluating the models resulting from dimension reduction. The fit of the models to test data is important for understanding how well the models generalize to new data, but application-driven metrics are also essential if the model is to be useful. When it is necessary for a human to interact with the model, interpretability should also be evaluated.

Fit of test data. A very common approach is to train a model on a portion of the data and to evaluate the fit of the model on another portion of the data. For LSI, the test documents can be projected into the latent semantic space and then the ℓ_2 error introduced by the approximation can be calculated. The probabilistic models can be evaluated by computing the probability of generating the test documents given the model.

Perplexity [4] is the most common way to report this probability. Computed as

$$\exp\left(-\frac{1}{N}\sum_{d=1}^{M}\sum_{n=1}^{N_d}\log p(w_{dn}|\text{model})\right),$$

the perplexity corresponds to the effective size of the vocabulary. For example, a value of 100 indicates that the probabilities resulting from the model are equivalent to randomly picking each word from a vocabulary of 100 words. This means that smaller values indicate that the model fits the test data better.

Wallach et al. [61], evaluate several different ways to compute this probability and recommended the *left-to-right* method, in which the probability of generating each token in a document is conditioned on all previous tokens in the document so that the interaction between the tokens in the document are properly accounted for.

Application performance. Another common approach is to measure the utility of topic models in some application. Whenever the di-

mension reduction is being carried out with a specific application in mind, this in an important evaluation. For example, Wei and Croft [65] discuss the evaluation of LDA models for document search using standard information retrieval metrics.

Interpretability. For text mining, the ability to use the discovered models to better understand the documents is essential. Unfortunately, the fit of test data and application performance metrics completely ignore the topical structure. In fact, models with better perplexity are often harder to interpret [18]. This is not surprising, because the task of finding a meaningful model that fits well is more constrained than the task of finding any model that fits well, so the best fit is likely to be found using a less meaningful model.

Chang et al. [18] propose a new evaluation protocol based on a user study. Starting with a list of top terms for each topic that has been tainted with an additional term, users are asked to identify the spurious term. User performance on this task is higher when the topic is coherent so that the extra term stands out. They also conducted a similar experiment to measure the appropriateness of topic assignments to test documents.

4.3 Parameter Selection

Asuncion et al. [3] compare a variety of different algorithms for the LDA model. They found that with careful selection of the regularization hyperparameters α and β, all of the algorithms had similar perplexity. A grid search over possible values yields the best performance, but interleaving optimization of the hyperparameters with iterations of the algorithm is almost as good with much less computational cost.

4.4 Dimension Reduction

Latent topic models, including LSI, PLSI and LDA, are commonly used as dimension reduction tools for texts. After the training process, the document d can be represented by its topic distribution $p(z|d)$, where z can be viewed as a K-dimensional representation of the original document. The similarity between documents can then be measured by their similarity in the topic space.

$$S_{dd'} = \sum_{z=1}^{K} p(z|d)p(z|d') .$$

Through this equation, documents are projected into a low dimensional space. The terms are projected into a K-dimensional space in the same way. For probabilistic topic models, KL divergence can be used for an alternative comparison.

Handling of synonymy is a natural result of dimension reduction. Multiple terms associated with the same concept are projected into the same place in the latent semantic space. Polysemy presents a more difficult challenge. Griffiths and Steyvers [31] found that LSI was able to detect polysemy: a term that was projected onto multiple latent dimensions generally had multiple meanings. LDA can resolve polysemy provided that one of the topics associated with a polysemous term is associated with additional tokens in the document.

5. Beyond Latent Dirichlet Allocation

LDA has many advantages for topic modeling, including its relative simplicity to implement and the useful topics that it unearths. However, with additional effort topic modeling can be adapted to the characteristics of a particular problem. In this section, we survey recent advances that make it practical to apply topic modeling to very large text corpora, dynamic data, data that is embedded in a network and other problems with special characteristics.

5.1 Scalability

Standard LDA learning algorithms read the documents in the training corpus numerous times and are inherently serial. In practice, this means that LDA models are trained on only a small fraction of the available data. However, recent advances in online and parallel algorithms make it reasonable to train and apply models at very large scale.

Efficient parallel implementations are available based on either collapsed Gibbs sampling or variational approximation. Smola et al. [56] perform Gibbs sampling based on slightly outdated term and topic statistics in parallel with threads that globally update the statistics. Using variational approximation, Asuncion et al. [3] interleave an inference step on all documents with a parallel aggregation of the term and topic statistics. Both of these methods achieve scalability through approximations that have no known convergence guarantee. In contrast, Yan et al. [66] and Liu et al. [43], use careful scheduling to achieve strong parallelization without approximation. However, the approximate methods are easier to implement correctly and work very well in practice.

5.2 Dynamic Data

Numerous approaches are possible when the corpus of documents is changing over time or must be processed as a stream. One common

approach is to augment the corpus with the time of each document and incorporate time into the model. Wang and Agichtein [64] model the revision history of documents by considering the temporal dimension and extend LSI to tensor factorization. PLSI can be similarly augmented to model the temporal patterns of activities in videos [59]. Mølgaard et al. [50] study temporal PLSI for music retrieval, which can be viewed as a probabilistic model for tensor factorization. Blei and Lafferty modeled time evolution of topic models [10] to analyze how the topics used in a corpus changed over time.

For streaming data, Yao et al. [72] present a time and space efficient algorithm for applying an existing topic model to a stream of documents, using a modification of Gibbs sampling. Hoffman et al. [35] developed an online algorithm for LDA that retrains the model for each document in turn. Interestingly, they found that this approach did not sacrifice any quality in the learned model as measured using perplexity. They further show that the online algorithm corresponds to stochastic gradient descent on the variational objective function, and so converges to a stationary point of that function.

The online training process for LDA optimizes each document in turn. First, it uses standard variational approximation to estimate the probability distribution for the topic of each word ϕ_j. Next, topic models $\tilde{\Phi}_i$ are estimated as if the corpus consisted of M copies of this document, based on ϕ_j.

$$\tilde{\Phi}_{ij} = \beta + MN_j\phi_{ij}.$$

The estimate of the topic models Φ_i are then updated to include the contribution from this document by

$$\Phi_{ij} \leftarrow (\Phi_{ij} + \rho\tilde{\Phi}_{ij})/(1+\rho),$$

where $\rho = (t_0+t)^{-\kappa}$ when processing the t-th document. t_0 is a parameter that slows the algorithm during the early iterations and $0.5 < \kappa \leq 1$ is a parameter that controls the rate of learning. This algorithm is essentially the variational algorithm applied to a different single document on each iteration, with appropriate changes to how the topic models are updated. For very large datasets, this is many times faster than other algorithms and yet yields very excellent results.

5.3 Networked Data

Networks play an important role in many text mining problems. Email messages are linked to the senders and recipients. Publications are also linked by citations. Many documents are related to a social network.

The analysis of these documents can reveal more interesting structure if the network graph can be incorporated.

LSI has been applied to analysis network data. Ng et al. analyze the connection between the LSI and HIST [51], which is a widely used algorithm for network data. Other approaches learn low-dimensional representations of documents based on both their contents and the citation graph between them through learning from multiple relationships between different types of entities [69, 76].

PLSI has also been applied to analyze the network data. Cohn and Chang apply PLSI to model the citation graph and identity authoritative document based on the latent factors [21]. Citations between documents can be modeled together with the contents of the documents in a joint probabilistic model [20], through the probability of generating a citation given a latent topic. Guo et al. [34] model the interaction of topics between linked documents. Intuitively, the topics of a document are borrowed from the documents to which it links. Deng et al. [25] propose the two frameworks based on random walk and regularization to propagate the topics of documents according to the links between them.

We describe the work of Mei et al. [46] in detail since it is representative of combining PLSI and network analysis. This work utilizes the network structure as the regularization for PLSI through assuming that the topic distributions are similar for documents connected to each other. The regularization term induced from the structure of the network is optimized together with the log-likelihood function of PLSI. The model is applied to several applications such as author-topic analysis and spatial topic analysis, where network structures are constructed from co-authorships and adjacency of locations, respectively.

Much research has explored various ways to integrate network information into topic models. Rosen-Zvi et al. incorporate authorship information through author-specific topic mixtures [55]. Supervised topic models allow the per-topic term distributions to depend on a document label [12]. Chang and Blei incorporate relational information between documents [17]. It is also possible to integrate general first order logic [2]. McCallum et al. extend LDA so that it can identify topic models that are conditioned on the author and the audience of the communication [45]. This is useful for analyzing the social dynamics of communication in a network.

Relational topic models (RTM) extend LDA to jointly model the generation of documents and the generation of links between documents [17]. The model predicts links based on the similarity of the topic mixture used in two documents, which adds the capability of predicting missing links in the graph structure. Because the links influence the

selection of topics, the model can more accurately predict links than a similar prediction based on topics from LDA.

The generative process of documents is the same for RTM as for LDA. Once the documents are generated, the link λ_{de} between documents d and e is generated from exponential regression on the empirical topic mixtures $\bar{\boldsymbol{Z}}_d$ and $\bar{\boldsymbol{Z}}_e$,

$$\lambda_{de} = \exp\left(\boldsymbol{\eta}^T(\bar{\boldsymbol{Z}}_d \circ \bar{\boldsymbol{Z}}_e) + \boldsymbol{\nu}\right),$$

where $\bar{\boldsymbol{Z}}_d \equiv 1/N_d \sum_{n=1}^{N} \boldsymbol{Z}_{dn}$ and $a \circ b$ is the element-wise product of vectors a and b. Typically the link is taken to be binary, in which case $\boldsymbol{\nu}$ is used to control the threshold. $\boldsymbol{\eta}$ is a parameter that must be learned which controls the importance of each topic in establishing the link. We generally expect it to have positive values, although a negative value in a social network would reflect the adage that opposites attract.

5.4 Adapting Topic Models to Applications

The graphical model of LDA can be easily extended to match the characteristics of a specific application. Here we survey some of the fruitful approaches.

One important class of extensions to LDA has been the introduction of richer priors for document topic and term distributions. Instead of using a fixed, global Dirichlet hyperparameter $\boldsymbol{\alpha}$ for all the documents in a corpus, Mimno and McCallum use regression from document features to establish a document-specific $\boldsymbol{\alpha}$ [48]. This is a valuable enhancement when other meta-features are available that are expected to influence the selected topics, as, for example, the identity of the author, the publication venue and the dates.

The Bayesian hierarchy of LDA provides a useful modeling pipeline for data with complex structure. The hierarchy can model web-like interconnections and uncertain labels [67, 71]. The *mixed membership stochastic block model* coupled two LDA hierarchies to model inter-connected entities [1], which provides a flexible model for network graphs and has proven useful for a variety of applications ranging from role discovery to community detection in social, biological and information networks.

Hierarchical topic models (hLDA) are used to identify subtopics that are increasingly more specific [9]. The hLDA model automatically learns a tree structure hierarchy for topics while they are discovered from the documents. For additional flexibility, hierarchical Dirichlet processes [57] can automatically discover an appropriate number of topics and subtopics. There are also principled ways to learn correlations between topics [11, 41]. Other extensions support richer document representations and contextual information, including bigrams [62], syntactic relationships [15, 33] and product aspects [58].

Multinomial distributions for term occurrences usually have a difficult time modeling the word burstiness in language — if a word appears in a document once, it will likely appear again in the same document. This effect is commonly referred to as Zipf's law, a profound characteristic of language. To discount this impact, Doyle and Elkan replace the per-topic Multinomial distribution with a Dirichlet-Compound Multinomial (also called the multivariate Pòlya) distribution) [27]. Reisinger et al. substitutes spherical admixture models [54], which not only incorporate negative correlations among term occurrence but also admit the natural use of cosine similarity to compare topics or documents.

Standard topic models are not appropriate for identifying consistent topics across multiple languages, because the multiple languages do not co-occur in documents frequently enough to be assigned into the same topics. Mimno et al. developed an extension that works with loosely *aligned* documents [49]—pairs of documents in different languages that have nearly the same mixture of topics. Boyd-Graber and Blei explore various strategies for discovering multilingual topics from unaligned documents [14]. Similar issues arise with documents in multiple dialects. Crain et al. [22] and Yang et al. [68] discuss extensions of LDA that find shared topics between consumer and technical medical documents.

6. Conclusion

Using a BOW representation results in very efficient text mining because more complex factors like grammar and word order can be neglected. However, working directly with individual terms has a number of strong limitations, because multiple documents can discuss the same ideas using very different words, and likewise, the same word can have very different meanings. Dimension reduction is able to lift the BOW representation to a more abstract level that better reflects the needs of a human analyst, where the new dimensions correspond to concepts or topics. In this way, alternative ways of expressing the same content can be reduced to a common representation and terms with multiple meanings can be identified.

LSI is based on a spectral analysis of the term-document matrix. This approach identifies common generalizations that are guaranteed to provide the best lower-dimensional representation of the original data. This representation is not necessarily easy to interpret, but is very useful for performing a conceptual match between two documents that may use different terms for the same concepts.

Probabilistic topic models provide an intuitive, probabilistic foundation for dimension reduction. They allow us to reason about the topics present in a document and expose the probability of seeing each word in any given topic. This makes it much easier to interpret what the topics mean. It also makes it easier to extend the models in interesting ways. Many extensions to PLSI and LDA have been developed, both to allow them to be applied to large scale data and to incorporate special structure for a particular application.

Acknowledgment

Part of the work is supported by NSF grants IIS-1049694, IIS-1116886, a Yahoo! Faculty Research and Engagement Grant and a Department of Homeland Security Career Development Grant.

References

[1] E. Airoldi, D. Blei, S. Fienberg, and E. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, June 2008.

[2] D. Andrzejewski, X. Zhu, M. Craven, and B. Recht. A framework for incorporating general domain knowledge into latent Dirichlet allocation using first-order logic. In *IJCAI*, 2011.

[3] A. Asuncion, M. Welling, P. Smyth, and Y. Teh. On smoothing and inference for topic models. In *UAI*, pages 27–34, 2009.

[4] L. Bahl, J. Baker, E. Jelinek, and R. Mercer. Perplexity—a measure of the difficulty of speech recognition tasks. In *Program, 94th Meeting of the Acoustical Society of America*, volume 62, page S63, 1977.

[5] H. Bast and D. Majumdar. Why spectral retrieval works. In *SIGIR*, page 11, 2005.

[6] J.-P. Benzecri. *L'Analyse des Donnees. Volume II.* 1973.

[7] M. Berry. Large-scale sparse singular value computations. *The International Journal Of Supercomputer Applications*, 6(1):13–49, 1992.

[8] M. Berry, S. Dumais, and G. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM review*, 37(4):573–595, 1995.

[9] D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, 2003.

[10] D. Blei and J. Lafferty. Dynamic topic models. In *ICML*, pages 113–120, 2006.

[11] D. Blei and J. Lafferty. A correlated topic model of science. *AAS*, 1(1):17–35, 2007.

[12] D. Blei and J. McAuliffe. Supervised topic models. In *NIPS*, 2007.

[13] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[14] J. Boyd-Graber and D. Blei. Multilingual topic models for unaligned text. In *UAI*, pages 75–82, 2009.

[15] J. Boyd-Graber and D. Blei. Syntactic topic models. In *NIPS*, pages 185–192. 2009.

[16] W. Buntine and A. Jakulin. Discrete component analysis. In Craig Saunders, Marko Grobelnik, Steve Gunn, and John Shawe-Taylor, editors, *Subspace, Latent Structure and Feature Selection*, volume 3940 of *Lecture Notes in Computer Science*, pages 1–33. Springer Berlin / Heidelberg, 2006.

[17] J. Chang and D. Blei. Relational topic models for document networks. In *AIStats*, 2009.

[18] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, pages 288–296. 2009.

[19] K. Church and W. Gale. Poisson mixtures. *Natural Language Engineering*, 1:163–190, 1995.

[20] D. Cohn. The missing link-a probabilistic model of document content and hypertext connectivity. In *NIPS*, 2001.

[21] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *ICML*, pages 167–174, 2001.

[22] S. Crain, S.-H. Yang, Y. Jiao, and H. Zha. Dialect topic modeling for improved consumer medical search. In *AMIA Annual Symposium*, 2010.

[23] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, September 1990.

[24] A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[25] H. Deng, J. Han, B. Zhao, Y. Yu, and C. Lin. Probabilistic Topic Models with Biased Propagation on Heterogeneous Information Networks. In *KDD*, pages 1271—-1279, San Diego, 2011. ACM.

[26] C. Ding. A similarity-based probability model for latent semantic indexing. In *SIGIR*, pages 58–65, 1999.

[27] G. Doyle and C. Elkan. Accounting for burstiness in topic models. In *ICML*, 2009.

[28] S. Dumais and J. Nielsen. Automating the assignment of submitted manuscripts to reviewers. In *SIGIR*, pages 233–244, 1992.

[29] G. Dupret. Latent concepts and the number orthogonal factors in latent semantic analysis. *SIGIR*, pages 221–226, 2003.

[30] G. Golub and C. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[31] T. Griffiths and M. Steyvers. *Latent Semantic Analysis: A Road to Meaning*, chapter Probabilistic topic models. 2006.

[32] T. Griffiths and M. Steyvers. Finding scientific topics. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 101, pages 5228–5235, 2004.

[33] T. Griffiths, M. Steyvers, D. Blei, and J. Tenenbaum. Integrating topics and syntax. In *NIPS*, pages 537–544, 2005.

[34] Z, Guo, S. Zhu, Y. Chi, Z. Zhang, and Y. Gong. A latent topic model for linked documents. In *SIGIR*, page 720, 2009.

[35] M. Hoffman, D. Blei, and F. Bach. Online learning for latent Dirichlet allocation. In *NIPS*, pages 856–864, 2010.

[36] T. Hofmann. Probabilistic latent semantic analysis. In *UAI*, page 21, 1999.

[37] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57, 1999.

[38] R. Kubota Ando and L. Lee. Iterative residual rescaling: An analysis and generalization of LSI. In *SIGIR*, pages 154–162, 2001.

[39] S. Kullback and R. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, March 1951.

[40] T. Landauer. On the computational basis of learning and cognition: Arguments from LSA. *Psychology of learning and motivation*, (1):1–63, 2002.

[41] W. Li, D. Blei, and A. McCallum. Nonparametric Bayes Pachinko allocation. In *UAI*, 2007.

[42] G. Lisowsky and L. Rost. *Konkordanz zum hebräischen Alten Testament: nach dem von Paul Kahle in der Biblia Hebraica edidit Rudolf Kittel besorgten Masoretischen Text.* Deutsche Bibelgesellschaft, 1958.

[43] Z. Liu, Y. Zhang, E.Y. Chang, and M. Sun. PLDA+: Parallel latent Dirichlet allocation with data placement and pipeline processing. *ACM Trans. Intell. Syst. Technol.*, 2:26:1–26:18, May 2011.

[44] C. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008.

[45] A. McCallum, A. Corrada-Emmanuel, and X. Wang. Topic and role discovery in social networks. In *Proceedings of the 19th international joint conference on Artificial intelligence*, pages 786–791, 2005.

[46] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *WWW*, page 101, 2008.

[47] Q. Mei, X. Shen, and C. Zhai. Automatic labeling of multinomial topic models. In *KDD*, pages 490–499, 2007.

[48] D. Mimno and A. McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *UAI*, 2008.

[49] D. Mimno, H. Wallach, J. Naradowsky, D. Smith, and A. McCallum. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 880–889, 2009.

[50] L. Mølgaard, J. Larsen, and D. Lyngby. Temporal analysis of text data using latent variable models. *2009 IEEE International Workshop on Machine Learning for Signal Processing*, 2009.

[51] A. Ng, A. Zheng, and M. Jordan. Link analysis, eigenvectors and stability. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 903–910, 2001.

[52] G. O'Brien. Information management tools for updating an SVD-encoded indexing scheme. *Master's thesis, The University of Knoxville, Tennessee*, (October), 1994.

[53] C. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 159–168, 1998.

[54] J. Reisinger, A. Waters, B. Silverthorn, and R. Mooney. Spherical topic models. In *ICML*, pages 903–910, 2010.

[55] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI*, 2004.

[56] A. Smola and S. Narayanamurthy. An architecture for parallel topic models. *Proc. VLDB Endow.*, 3:703–710, September 2010.

[57] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical Dirichlet processes. *JASA*, 101, 2006.

[58] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *WWW*, pages 111–120, 2008.

[59] J. Varadarajan, R. Emonet, and J. Odobez. Probabilistic latent sequential motifs: Discovering temporal activity patterns in video scenes. In *BMVC 2010*, volume 42, pages 177–196, 2010.

[60] H. Wallach, D. Mimno, and A. McCallum. Rethinking LDA: Why priors matter. In *NIPS*, pages 1973–1981, 2009.

[61] H. Wallach, I. Murray, R. Salakhutdinov and D. Mimno. Evaluation methods for topic models In *ICML*, pages 1105–1112, 2009.

[62] H. Wallach. Topic modeling: beyond bag-of-words. In *ICML*, 2006.

[63] Q. Wang, J. Xu, and H. Li. Regularized latent semantic indexing. In *SIGIR*, 2011.

[64] Y. Wang and E. Agichtein. Temporal latent semantic analysis for collaboratively generated content: preliminary results. In *SIGIR*, pages 1145—-1146, 2011.

[65] X. Wei and W. Bruce Croft. LDA-based document models for ad-hoc retrieval. In *SIGIR*, pages 178–185, 2006.

[66] F. Yan, N. Xu, and Y. Qi. Parallel inference for latent Dirichlet allocation on graphics processing units. In *NIPS*, pages 2134–2142. 2009.

[67] S. Yang, J. Bian, and H. Zha. Hybrid generative/discriminative learning for automatic image annotation. In *UAI*, 2010.

[68] S. Yang, S. Crain, and H. Zha. Briding the language gap: topic-level adaptation for cross-domain knowledge transfer. In *AIStat*, 2011.

[69] S. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike – joint friendship and interest propagation in social networks. In *WWW*, 2011.

[70] S. Yang and H. Zha. Language pyramid and multi-scale text analysis. In *CIKM*, pages 639–648, 2010.

[71] S. Yang, H. Zha, and B. Hu. Dirichlet-bernoulli alignment: A generative model for multi-class multi-label multi-instance corpora. In *NIPS*, 2009.

[72] L. Yao, D. Mimno, and A. McCallum. Efficient methods for topic model inference on streaming document collections. In *KDD*, pages 937–946, 2009.

[73] Y. Saad. *Numerical Methods for Large Eigenvalue Problems.* Manchester University Press ND, 1992.

[74] H. Zha and H. Simon. On updating problems in latent semantic indexing. *SIAM Journal on Scientific Computing*, 21(2):782, 1999.

[75] H. Zha and Z. Zhang. On matrices with low-rank-plus-shift structures: Partial SVD and latent semantic indexing. *SIAM Journal Matrix Analysis and Applications*, 21:522–536, 1999.

[76] D. Zhou, S. Zhu, K. Yu, X. Song, B. Tseng, H. Zha, and C. Lee Giles. Learning multiple graphs for document recommendations. In *WWW*, page 141, 2008.