Embedding Semantics in LDA Topic Models

Chapter · March 2010

DOI: 10.1002/9780470689646.ch10

4 authors, including:

Loulwah Alsumait
Kuwait University
10 PUBLICATIONS **638** CITATIONS

SEE PROFILE

Carlotta Domeniconi
George Mason University
216 PUBLICATIONS **4,750** CITATIONS

SEE PROFILE

Daniel Barbara
George Mason University
168 PUBLICATIONS **7,358** CITATIONS

SEE PROFILE

10

Embedding semantics in LDA topic models

Loulwah AlSumait, Pu Wang, Carlotta Domeniconi and Daniel Barbará

10.1 Introduction

The huge advancement in databases and the explosion of the Internet, intranets, and digital libraries have resulted in giant text databases. It is estimated that approximately 85% of worldwide data is held in unstructured formats with an increasing rate of roughly 7 million digital pages per day (White 2005). Such huge document collections hold useful yet implicit and nontrivial knowledge about the domain. Text mining (TM) is an integral part of data mining that is aimed at automatically extracting such knowledge from the unstructured textual data. The main tasks of TM include text classification, text summarization, document and/or word clustering, in addition to classical natural language processing tasks such as machine translation and question-answering. The learning tasks are more complex when processing text documents that arrive in discrete or continuous streams over time.

Topic modeling is a newly emerging approach to analyze large volumes of unlabeled text (Steyvers and Griffiths 2005). It specifies a statistical sampling technique to describe how words in documents are generated based on (a small set of) hidden topics. In this chapter, we investigate the role of prior knowledge semantics in estimating the topical structure of large text data in both batch and online modes under the framework of latent Dirichlet alglocation (LDA) topic

Text Mining: Applications and Theory Edited by Michael W. Berry and Jacob Kogan
© 2010, John Wiley & Sons, Ltd. ISB N: 978-0-470-74982-1

modeling (Blei et al. 2003). The objective is to enhance the descriptive and/or predictive model of the data's thematic structure based on the embedded prior knowledge about the domain's semantics.

The prior knowledge can be either external semantics from prior-knowledge sources, such as ontologies and large universal datasets, or a data-driven semantics which is a domain knowledge that is extracted from the data itself. This chapter investigates the role of semantic embedding in two main directions. The first is to embed semantics from an external prior-knowledge source to enhance the generative process of the model parameters. The second direction which suits the online knowledge discovery problem is to embed data-driven semantics. The idea is to construct the current LDA model based on information propagated from topic models that were learned from previously seen documents of the domain.

10.2 Background

Given the unstructured nature of text databases, many challenges face TM algorithms. First, there are a very high number of possible features to represent a document. Such features can be derived from all the words and/or phrase types in the language. Furthermore, in order to unify the data structure of documents, it is necessary to use a dictionary of all the words to represent a document, which results in a very sparse representation. Another critical challenge stems from the complex relationships between concepts and from the ambiguity and context sensitivity of words in text. Thus, a good TM algorithm must be efficient to process such large and challenging data so that the documents are represented in short descriptions in which only the essential and most discriminative information is preserved. The rest of this section is focused on three major advancements to solve this problem, then the LDA topic models will be introduced in Section 10.3.

10.2.1 Vector space modeling

The first major progress in text processing was due to the vector space model (Salton 1983), in which a document is represented as a vector of dimension W, $\mathbf{w}_d = (w_{1d}, \ldots, w_{Wd})$, where each dimension is associated with one term of the dictionary. Each entry w_{id} is the *term frequency – inverse document frequency* (tf-idf) of the term i in document d given by $w_{id} = n_{id} \times \log(D/n_i)$. The local frequency of the term (n_{id}) is weighted by its global frequency in the whole corpus to reduce the importance of common words that appear in many documents since they are naturally bad discriminators. To represent the whole corpus, the term – document matrix, X, is constructed. X is a $W \times D$ matrix whose rows are indexed by the terms of the dictionary and whose columns are indexed by the documents.

Although the VSM has empirically shown its effectiveness and is widely used, it suffers from a number of inherent shortages to capture inter- and intra-document statistical structure and provides a small reduction only in the description of the corpus.

10.2.2 Latent semantic analysis

To address the shortages of the VSM, researchers in information retrieval (IR) have introduced latent semantic analysis (LSA) (Deerwester et al. 1990), which is a factor analysis that reduces the term – document matrix to a K-dimensional subspace that captures most of the variance in the corpus. By computing the singular value decomposition (SVD), the term – document matrix X is decomposed into three matrices $X = U\Sigma V^T$. The rows in U give the occurrence of the original words which correspond to the K *concepts* of the new factor space, while the columns in V give the relation between the documents and each of the K concepts.

Although LSA overcomes some of the drawbacks of the VSM, it suffers from a number of limitations. First, given the high-dimensionality nature of text data, computation of the SVD is expensive. In addition, the new feature space is very difficult to interpret since each dimension is a linear combination of a set of words from the original space. LSA is also not generalizable to incorporate other side information such as time and author.

10.2.3 Probabilistic latent semantic analysis

Researchers have proposed statistical approaches to understand LSA, some of whom have discussed its relationship to Bayesian methods (Story 1996) and generative probabilistic models (Papadimitriou et al. 2000). As a major advance in the application of Bayesian methods to document modeling, Hofmann (1999) introduced probabilistic latent semantic analysis (pLSA), also called the *aspect model*, as an alternative to LSA. It is a latent variable model that associates an unobserved class (aspect) variable z_k with each document d and represents each aspect by a distribution over words $p(\mathbf{w}|\mathbf{z})$. The pLSA model is parameterized by the joint distribution of a document d and a word w_{di} that appears in it, $p(d, w_{di}) = p(d)\sum_{z=1}^{K} p(w_{di}|z)p(z|d)$.

A graphical model of pLSA is shown in Figure 10.1. Given the hidden aspects, the documents and words are conditionally independent. In addition, pLSA allows the documents to be associated with a mixture of topics weighted by the posterior $p(\mathbf{z}|d)$.

The generative process of a model specifies a probabilistic sampling procedure that describe how words in documents can be generated based on the hidden topics. Thus, the generative process of the pLSA is as follows:

1. Draw a document with probability $p(d)$.
2. For each word i in document d:
   (a) Draw a latent aspect z_i with probability $p(z_i|d)$.
   (b) Draw a word w_{di} with probability $p(w_{di}|z_i)$.

Nonetheless, this is not a true generative model as the variable d is a dummy random variable that is indexed by the documents in a training set (Blei et al.

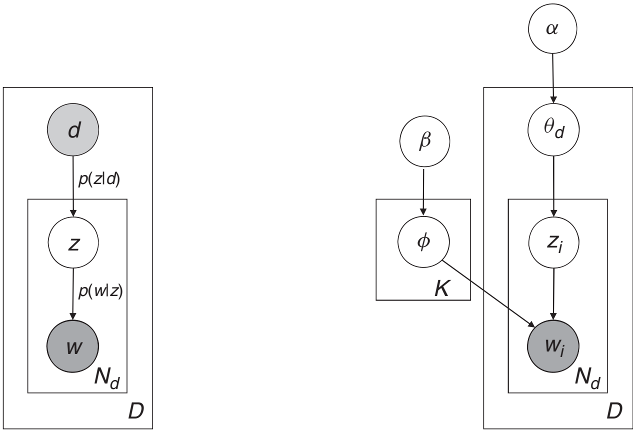

Figure 10.1 A graphical model of pLSA (left) and LDA (right).

2003). As a consequence, pLSA is inclined to overfit the training data, which harms its ability to generalize the inferred aspect model to generate previously unseen documents.

Despite its limitation, pLSA has influenced a huge amount of work in statistical machine learning and TM. As a result, a class of statistical models, named probabilistic topic models (PTMs), have been created to uncover the underlying structure of large collections of discrete data, such as text. PTMs are generative models of documents that assume the existence of hidden variables, representing topics associated with the observed text documents which are responsible for the patterns of word use. Topic models are aimed at discovering these hidden variables based on hierarchical Bayesian analysis. Among the variety of topic models proposed, LDA (Blei et al. 2003) is a truly generative model that is capable of generalizing the topic distributions so that it can be used to generate unseen documents as well.

10.3 Latent Dirichlet allocation

The LDA PTM is a three-level hierarchical Bayesian network that represents the generative probabilistic model of a corpus of documents. The basic idea is that documents are represented by a mixture of topics where each topic is a latent multinomial variable characterized by a distribution over a fixed vocabulary of words. The completeness of the LDA's generative process for documents is achieved by considering Dirichlet priors on the document distributions over topics and on the topic distributions over words. This emerging approach has been successfully applied to find useful structures in many kinds of documents, including emails, the scientific literature (Griffiths and Steyvers 2004), libraries of digital books (Mimno and McCallum 2007), and news archives (Wei and Croft 2006).

This section introduces the LDA topic model with a brief description of its graphical model and generative process (Section 10.3.1) and the posterior inference (Section 10.3.2). The section concludes with a brief review of an online version of LDA, namely OLDA.

10.3.1 Graphical model and generative process

LDA relates words and documents through latent topics based on the bag-of-words assumption, i.e. the *exchangeability*, for the words in a document and for the documents in a corpus. The graphical model of LDA is given in Figure 10.1. The documents $\boldsymbol{\theta}$ are not directly linked to the words $\mathbf{w}$. Rather, this relationship is governed by additional latent variables, z, introduced to represent the responsibility of a particular topic in using that word in the document, i.e. the topic(s) that the document is focused on. By introducing the Dirichlet priors α and β over the document and topic distributions, respectively, the generative model of LDA is complete and is capable of processing unseen documents.

So, the structure of the LDA model allows the interaction of the observed words in documents with structured distributions of a *hidden variable model* (Blei et al. 2003). Learning the structure of the hidden variable model can be achieved by inferring the posterior probability distribution of the hidden variables, i.e. the topical structure of the collection, given the observed documents. This interaction can be viewed in the generative process of LDA:

1. Draw K multinomials ϕ_k from a Dirichlet prior β, one for each topic k.
2. Draw D multinomials θ_d from a Dirichlet prior α, one for each document d.
3. For each document d in the corpus, and for each word w_{di} in the document:
    (a) Draw a topic z_i from multinomial θ_d; $(p(z_i|\alpha))$.
    (b) Draw a word w_i from multinomial ϕ_z; $(p(w_i|z_i, \beta))$.

Inverting the generative process, i.e. fitting the hidden variable model to the observed data (words in documents), corresponds to inferring the latent variables and, hence, learning the distributions of underlying topics. The hidden structure of topics in the LDA model is described by the posterior distribution of the hidden variables given the D documents

$$p(\Theta, \mathbf{z}, \Phi|\mathbf{w}, \alpha, \beta) = \frac{p(\mathbf{w}, \Theta, \mathbf{z}, \Phi|\alpha, \beta)}{\int_{\phi_{1:K}} \int_{\theta_{1:D}} p(\mathbf{w}|\alpha, \beta)}. \tag{10.1}$$

10.3.2 Posterior inference

In LDA, exploring the data and extracting the topics correspond to computing the posterior expectations. These are the topic probability over terms $(\mathrm{E}(\Phi|\mathbf{w}))$, the document proportions over topics (E($\Theta|\mathbf{w}$)), and the topic assignments of words (E($\mathbf{z}|\mathbf{w}$)). Although the LDA model is relatively simple, exact inference of the posterior distribution in Equation (10.1) is intractable (Blei et al. 2003). The solution is to use sophisticated approximations such as variational expectation maximization (Blei et al. 2003) and expectation propagation (Minka and Lafferty 2002).

Griffiths and Steyvers (2004) proposed a simple and effective strategy for estimating ϕ and θ. It is an approximate iterative technique that is a special form of Markov chain Monte Carlo (MCMC) methods. Gibbs sampling is able to simulate a high-dimensional probability distribution $p(\mathbf{x})$ by iteratively sampling one dimension x_i at a time, conditioned on the values of all other dimensions, which is usually denoted $\mathbf{x}_{\neg i}$.

Under Gibbs sampling, ϕ and θ are not explicitly estimated. Instead, the posterior distribution over the assignments of words to topics, $P(\mathbf{z}|\mathbf{w})$, is approximated by means of the Monte Carlo algorithm, see Heinrich (2005) for a detailed derivation of the algorithm. Gibbs sampling iterates over each word token in the text collection in a random order and estimates the probability of assigning the current word token to each topic ($P(z_i = j)$), conditioned on the topic assignments to all other word tokens ($\mathbf{z}_{\neg i}$) as (Griffiths and Steyvers 2004)

$$P(z_i = j|\mathbf{z}_{\neg i}, w_i, \boldsymbol{\alpha}, \boldsymbol{\beta}) \propto \frac{C^{KW}_{w_{\neg i},j} + \beta_{w_{di},j}}{\sum_{v=1}^{W}(C^{KW}_{v,j} + \beta_{v,j})} \times \frac{C^{KD}_{d_{\neg i},j} + \alpha_{d,j}}{\sum_{k=1}^{K}(C^{KD}_{d,k} + \alpha_{d,k})}, \quad (10.2)$$

where $C^{KW}_{w_{\neg i},j}$ is the number of times word w is assigned to topic j, not including the current token instance i; and $C^{KD}_{d_{\neg i},j}$ is the number of times topic j is assigned to some word token in document d, not including the current instance i. From this distribution, i.e. $p(z_i|\mathbf{z}_{\neg i}, \mathbf{w})$, a topic is sampled and stored as the new topic assignment for this word token. After a sufficient number of sampling iterations, the approximated posterior can be used to get estimates of ϕ and θ by examining the counts of word assignments to topics and topic occurrences in documents.

Given the direct estimate of topic assignments z for every word, it is important to obtain its relation to the required parameters Θ and Φ. This is achieved by sampling new observations based on the current state of the Markov chain (Steyvers and Griffiths 2005). Thus, estimates $\acute{\Theta}$ and $\acute{\Phi}$ of the word – topic and topic – document distributions can be obtained from the count matrices

$$\acute{\phi}_{ik} = \frac{C^{WK}_{i,k} + \beta_{i,k}}{\sum_{v=1}^{W}(C^{WK}_{v,k} + \beta_{v,k})}, \qquad \acute{\theta}_{dk} = \frac{C^{DK}_{d,k} + \alpha_{d,k}}{\sum_{j=1}^{K}(C^{DK}_{d,j} + \alpha_{d,j})}. \quad (10.3)$$

Gibbs sampling has been empirically tested to determine the required length of the burn-in phase, the way to collect samples, and the stability of inferred topics (Griffiths and Steyvers 2004; Heinrich 2005; Steyvers and Griffiths 2005).

10.3.3 Online latent Dirichlet allocation (OLDA)

OLDA is an online version of the LDA model that is able to process text streams (AlSumait et al. 2008). The OLDA model considers the temporal ordering information and assumes that the documents arrive in discrete time slices. At each time slice t of a predetermined size ε, e.g. an hour, a day, or a year, a stream of documents, $S^t = \{d_1, \ldots, d_{D^t}\}$, of variable size, D^t, is received and ready to be processed. A document d received at time t is represented as a vector of word tokens, $\mathbf{w}_d^t = \{w_{d1}^t, \ldots, w_{dN_d}^t\}$. Then, an LDA topic model with K components is used to model the newly arrived documents. The generated model, at a given time, is used as a prior for LDA at the successive time slice, when a new data stream is available for processing (see Figure 10.2 for an illustration). The hyperparameters β can be interpreted as the prior observation counts on the number of times words are sampled from a topic before any word from the corpus is observed (Steyvers and Griffiths 2005), bishop. So, the count of words in topics, resulting from running LDA on documents received at time t, can be used as the priors for the $t+1$ stream.

Thus, the per-topic distribution over words at time t, $\Phi_k^{(t)}$, is drawn from a Dirichlet distribution governed by the inferred topic structure at time $t-1$ as follows:

$$\begin{aligned} \Phi_k^{(t)} | \boldsymbol{\beta}_k^{(t)} &\sim Dirichlet(\boldsymbol{\beta}_k^{(t)}) \\ &\sim Dirichlet(\omega \hat{\Phi}_k^{(t-1)}), \end{aligned} \qquad (10.4)$$

where $\hat{\Phi}_k^{(t-1)}$ is the frequency distribution of a topic k over words at time $t-1$ and $0 < \omega \leq 1$ is an *evolution tuning parameter* that is introduced to control the evolution rate of the model. Since the Dirichlet hyperparameters determine the smoothness degree of the priors, it is important to control its effect and to balance

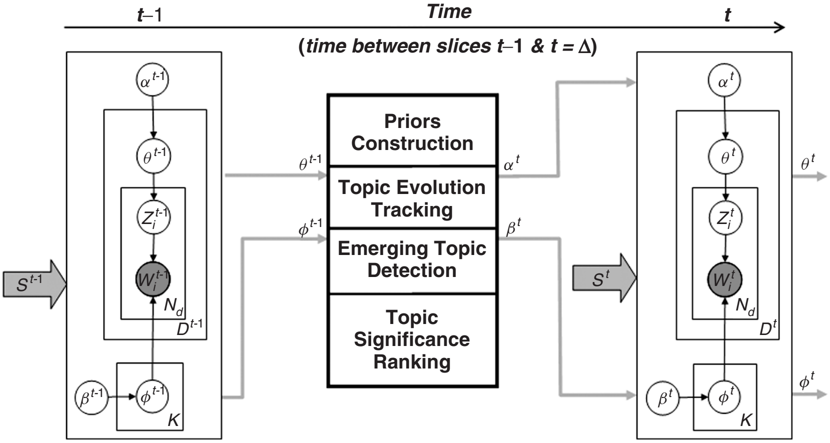

Figure 10.2 A flowchart of OLDA.

between the weight of the past and current semantics in the inference process according to the homogeneity and the evolution rate of the domain's thematic structure. This sequential model is expanded in Section 10.5 to allow data-driven semantic embedding from a wider range of previous models.

Given the definition of $\boldsymbol{\beta}^{(t)}$ in expression (10.4), the topic distributions in consecutive models are aligned so that the evolution of topics in a sequential corpus is captured. For example, if a topic distribution at time t corresponds to a particular theme, then the distribution that has the same ID number in the consecutive models will relate to the same theme, assuming that it appears consistently over time. Thus, the inferred word distribution of topic k at time t can be considered a drifted description of the latent variable k at time $t-1$. The drift is driven by the natural evolution of the topic which includes the changes that occur in the terminology and/or in the interactions with other topics. To model this evolution, an *evolutionary matrix*, $\mathbf{B}_k^{(t)}$, is constructed to capture the evolution of each topic k at each time epoch t within a *sliding history window*, δ. This is given as follows:

$$\mathbf{B}_k = \begin{pmatrix} \phi_1^{t-\delta} & \dots & \phi_1^{(t-1)} & \phi_1^{(t)} \\ \phi_2^{t-\delta} & \dots & \phi_2^{(t-1)} & \phi_2^{(t)} \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{W^{(t)}}^{t-\delta} & \dots & \phi_{W^{(t)}}^{(t-1)} & \phi_{W^{(t)}}^{(t)} \end{pmatrix}, \tag{10.5}$$

where each entry $B_k(v,t)$ is the weight of word v under topic k at time t.[1] Thus, working with the evolutionary matrix will allow for tracking the drifts of existing topics, detection of emerging topics, and visualizing the data in general.

Thus, the generative model for time slice t of the proposed OLDA model can be summarized as follows:

1. For each topic $k = 1, \dots, K$:
   (a) Compute $\boldsymbol{\beta}_k^{(t)} = \omega\hat{\Phi}_k^{(t-1)}$.

   (b) Generate a topic $\Phi_k^{(t)} \sim \mathit{Dirichlet}(\cdot|\boldsymbol{\beta}_k^{(t)})$.
2. For each document, $d = 1, \dots, D^{(t)}$:
   (a) Draw $\Theta_d^{(t)} \sim \mathit{Dirichlet}(\cdot|\boldsymbol{\alpha}^{(t)})$.

   (b) For each word token, w_{di}, in document d:
      i. Draw $z_i^{(t)}$ from multinomial $\Theta_d^{(t)}$; $(p(z_i^{(t)}|\boldsymbol{\alpha}_d^{(t)}))$.

      ii. Draw $w_{di}^{(t)}$ from multinomial $\Phi_{z_i}^{(t)}$; $p(w_{di}^{(t)}|z_i^{(t)}, \boldsymbol{\beta}_{z_i}^{(t)})$.

[1] New observed terms at time t are assumed to have 0 count in ϕ for all topics in previous streams.

Maintaining the models' priors as Dirichlet is essential to simplify the inference problem by making use of the conjugacy property of Dirichlet and multinomial distributions. In fact, by tracking the history as prior patterns, the data likelihood and, hence, the posterior inference of LDA are left the same. Thus, implementing Gibbs sampling in Equation (10.2) in OLDA is straightforward. The main difference of the online approach is that the sampling is performed over the current stream only. This makes the time complexity and memory usage of OLDA efficient and practical. In addition, the β under OLDA are constructed from historic observations rather than fixed values.

10.3.4 Illustrative example

The LDA and OLDA models can be illustrated by generating artificial data from a known topic model and applying the topic models to check whether the data is able to infer the original generative structure. To illustrate the LDA model, six sets of documents are generated from three topic distributions that are equally weighted. Table 10.1 shows the dictionary and topic distributions of the data.

For each set, 16 documents of size 16 word tokens, on average, are generated. After the word assignment vector, **z**, is randomly initialized, LDA is trained over the documents with the number of components K equal to the true number of components, i.e. K is set to 3. Table 10.2 gives the word – topic correlation counts of LDA averaged over the six sets of documents after 50 iterations of Gibbs sampling. It can be seen that the LDA model is able to correctly estimate the density of each topic.

Table 10.1 Topic distributions of simulated data. Each column is a multinomial distribution of a topic over the dictionary.

Topic	k_1 33%	k_2 34%	k_3 33%
Dictionary↓	$p(w_i\|k_1)$	$p(w_i\|k_2)$	$p(w_i\|k_3)$
river	0.37	0	0
stream	0.41	0	0
bank	0.22	0.28	0
money	0	0.3	0.07
loan	0	0.2	0
debt	0	0.12	0
factory	0	0	0.33
product	0	0	0.25
labor	0	0	0.25
news	0.05	0.05	0.05
reporter	0.05	0.05	0.05

Table 10.2 The frequency distributions of topics discovered by LDA from the static simulated data with K equal to 3.

Topic	T_1	T_2	T_3
	29.8%	35.5%	34.7
Dictionary	$f(w_i \mid T_1)$	$f(w_i \mid T_2)$	$f(w_i \mid T_3)$
river	0	0	78
stream	0	0	93
bank	0	56	71
money	0	103	0
loan	0	56	0
debt	0	28	0
factory	85	0	0
production	73	0	0
labor	61	0	0
news	3	19	15
reporter	10	15	14

Table 10.3 Topic distributions of dynamic simulated data over three streams. The rule (—) indicates that the corresponding word or topic has not yet emerged.

Stream	$t = 1$			$t = 2$			$t = 3$		
Topic	k_1	k_2	k_3	k_1	k_2	k_3	k_1	k_2	k_3
	40%	60%	0%	40%	50%	10%	30%	40%	30%
Dictionary↓	$p(w_i \mid k_j)$			$p(w_i \mid k_j)$			$p(w_i \mid k_j)$		
river	0.2	0	–	0.4	0	0	0.37	0	0
stream	0.4	0	–	0.2	0	0	0.41	0	0
bank	0.3	0.35	–	0.25	0.36	0.1	0.22	0.28	0
money	0	0.3	–	0	0.24	0	0	0.3	0.07
loan	0	0.25	–	0.05	0.22	0.1	0	0.2	0
debt	–	–	–	0	0.08	0	0	0.12	0
factory	–	–	–	0	0	0.37	0	0	0.33
product	–	–	–	0	0	0.33	0	0	0.25
labor	–	–	–	–	–	–	0	0	0.25
news	0.05	0.05	–	0.05	0.05	0.05	0.05	0.05	0.05
reporter	0.05	0.05	–	0.05	0.05	0.05	0.05	0.05	0.05

Given the same dictionary, three streams of documents are generated from evolving descriptions of topics to demonstrate the OLDA model. Table 10.3 shows the distributions of topics in the three time epochs. Topic 3 emerges as a new topic at the second time epoch. In addition to the new terms introduced by

Table 10.4 Topics discovered by OLDA from dynamic simulated data.

$t = 1$		$t = 2$		$t = 3$	
ID	Topic distribution	ID	Topic distribution	ID	Topic distribution
1	news reporter	1	news reporter	1	reporter news
2	bank	2	bank	2	bank
3	money loan	3	money loan debt	3	money loan debt
4	stream river	4	river stream	4	river stream
5	bank news	5	bank factory production	5	production factory labor

topic 3, a number of terms such as debt and labor gradually emerge. The weight (importance) of topics also varies between the streams. The OLDA topic model is trained on the corresponding documents of each stream with K set to 5. At each time epoch, OLDA is trained on the currently generated documents only. Table 10.4 lists the highest important words under each topic of the evolving simulated data that were discovered by OLDA with K set to 5 at each time epoch. After 50 iterations of Gibbs sampling on each stream, OLDA converged to aligned topic models that correspond to the true topic densities and evolution.

Another observation stems from the setting of K, i.e. the number of components. When K is set to the true number of topics, the topic distributions included some common words in addition to the semantically descriptive ones, see for example the words *news* and *reporter* in topics T_1, T_2, and T_3 in Table 10.2. When K is increased to 5, the topics became more focused as the common words are mapped into individual topics, see topics 1 and 2 in Table 10.3.

10.4 Embedding external semantics from Wikipedia

This section investigates the role of embedding semantics from a source by enhancing the generative process of the model parameters. Such human-defined concept databases provide a natural source of semantics that can provide useful knowledge regarding the hidden thematic structure of the data. We model external knowledge using Wikipedia (Wikipedia 2009). Wikipedia is currently considered the richest online encyclopedia, which consists of a huge number of categorized and consistently structured documents. After the identification of related Wikipedia concepts, LDA is applied to learn a model of the topics discussed in the corresponding Wikipedia articles. The learned topics represent priors about the available knowledge that will be embedded in the inference process of the LDA model to enhance the discovered topics from the text data, which will be referred to hereafter as the test documents.

10.4.1 Related Wikipedia articles

In this work, each Wikipedia article is represented by its title and considered as a single concept. Since Wikipedia includes a large variety of concepts and domains, it is important to use the most related articles to the test documents in order to ensure semantic relatedness and, hence, enhance the inferred model. The related Wikipedia articles are defined to be all Wikipedia concepts that are mentioned in a preset number of test documents, ρ. This is done by searching for the title of the Wikipedia article in the test documents. The threshold value ρ controls the number of Wikipedia articles, $\mathcal{D}$, to be retrieved and, hence, the amount of noise that is allowed to be included in the generative model.

10.4.2 Wikipedia-influenced topic model

After the identification of related Wikipedia concepts, LDA is applied to learn the topics that are discussed in the corresponding Wikipedia articles. In particular, LDA learns two Wikipedia distributions, the topic – word distribution $\boldsymbol{\phi}$ and the topic – document distribution $\boldsymbol{\theta}$, from

$$\phi_{ik} = \frac{C^{WK}_{w_i,k} + \beta_i}{\sum_{v=1}^{W} C^{WK}_{v,k} + \beta_v}, \qquad \theta_{mk} = \frac{C^{DK}_{m,k} + \alpha_k}{\sum_{j=1}^{K} C^{DK}_{m,j} + \alpha_j}, \tag{10.6}$$

where m is the index of the Wikipedia article. Within the related Wikipedia articles, $C^{WK}_{i,k}$ is the number of times word i is assigned to topic k and $C^{DK}_{m,k}$ is the number of times topic k is assigned to some word token in Wikipedia article m.

The prior distributions $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are then updated into posteriors using the test documents. Specifically, the topic – word distribution $\boldsymbol{\phi}$ is updated to a new $\hat{\boldsymbol{\phi}}$, and a new topic – document distribution $\hat{\boldsymbol{\theta}}$ is learned from scratch using the test documents

$$\hat{\phi}_{ik} = \frac{C^{WK}_{w_i,k} + \underline{C}^{WK}_{w_i,k} + \beta_i}{\sum_{v=1}^{V} C^{WK}_{v,k} + \underline{C}^{WK}_{v,k} + \beta_v}, \quad \hat{\theta}_{dk} = \frac{\underline{C}^{DK}_{d,k} + \alpha_k}{\sum_{j=1}^{K} \underline{C}^{DK}_{d,j} + \alpha_j}, \tag{10.7}$$

where d is the index of the test document, $\underline{C}^{WK}_{v,k}$ is the number of times word v is assigned to topic k, and $\underline{C}^{DK}_{d,k}$ is the number of times topic k is assigned to some word in test document d. Hence, the generative process of the test documents is influenced by the Wikipedia topic model.

10.5 Data-driven semantic embedding

When a topic is observed at a certain time, it is more likely to appear in the future with a similar distribution over words. Unlike general data mining techniques, such an assumption is trivial in the area of TM. It is widely acceptable, for instance, to consider the documents and the words in the documents to be statistically dependent. Once a word occurs in a document, it is likely to occur again. Consequently, a similar implication can be made about the topic distribution over time. Despite their natural drifts, the underlying themes of any domain are, in general, consistent. Hence, incorporating prior knowledge about the underlying semantics would eventually enhance the identification and description of topics in the future. In this section, the role of previously discovered topics in inferring future semantics in text streams is investigated under the framework of OLDA topic modeling. A detailed version of the proposed approach can be found in AlSumait et al. (2009).

OLDA is extended to enable semantic embedding in three major directions. First, instead of generating the topic parameters based on the most recently estimated model, the history window is set to incorporate more models in the parameter generation process. Second, the contribution of the semantic history in the inference process is controlled by assigning different weights to different time epochs. Lastly, given the evolutionary matrices of topics defined in Equation (10.5), the priors can be generated using a weighted linear combination of the semantics extracted from all the models that fall within the history window. These three factors are further explained in the following subsections.

10.5.1 Generative process with data-driven semantic embedding

To incorporate inferred semantics from past data, the proposed approach considers all the topic – word distributions learned within a sliding history window, δ, when constructing the current priors. As a result, OLDA can provide alternatives for full, short, or intermediate memory of history.

Given the sliding history window of size c, $1 < c \leq t$, the weight of past models in the prior construction can be controlled by defining a vector of evolution tuning parameters $\boldsymbol{\omega}$, instead of the single parameter in expression (10.4). The evolution tuning vector can be used to control the weights of individual models as well as the total weight of history with respect to new semantics. The setting depends mainly on the homogeneity of the data and on the evolution rate of the domain.

The overall influence of history in topic estimation is an important factor that can effect the semantic description of the data. For example, some text repositories, like the scientific literature, persistently introduce novel ideas and, as a consequence, topic distributions change faster compared to other datasets. On the other hand, a great part of the news in news feeds, like sports, stock markets, and weather, are steady over time. Thus, for such consistent topic structures, assigning a higher weight for historic information, compared to the weight of current observations, would improve topic prediction, while the settings should be reversed in fast evolving datasets.

By adjusting the total weight of history, i.e. $\sum_{c=1}^{\delta} \omega_c$, the OLDA model provides a direct way to deploy and tune the influence of history in the inference process. If the total history weight is equal to one, this would (relatively) balance the weights of historic and current observations. When the total weight of history is less (greater) than one, the historic semantic has less (more) influence than the semantic of the current stream.

Thus, given the sliding window δ, the history weight vector $\boldsymbol{\omega}$, and the evolutionary matrix of topic $k\mathbf{B}_k^{(t)}$, as defined in Equation (10.5), the parameters of topic k at time t can be determined by a weighted mixture of the topic's past distributions

$$\begin{aligned}\boldsymbol{\beta}_k^{(t)} &= \mathbf{B}_k^{(t-1)}\boldsymbol{\omega} && (10.8)\\ &= \hat{\Phi}_k^{(t-\delta)}\omega_1 + \cdots + \hat{\Phi}_k^{(t-2)}\omega_{\delta-1} + \hat{\Phi}_k^{(t-1)}\omega_\delta. && (10.9)\end{aligned}$$

Given the equality in Equation (10.8), the per-topic distribution over words at time t, $\Phi_k^{(t)}$, is drawn from a Dirichlet distribution governed by the evolutionary matrix of the topic as follows:

$$\begin{aligned}\Phi_k^{(t)}|\boldsymbol{\beta}_k^{(t)} &\sim Dirichlet(\boldsymbol{\beta}_k^{(t)})\\ &\sim Dirichlet(\mathbf{B}_k^{(t-1)}\boldsymbol{\omega}). && (10.10)\end{aligned}$$

By updating the priors as described above, the structure of the model is kept simple, as all the historic knowledge patterns are printed in the priors rather than in the structure of the graphical model itself. In addition, the learning process on the new stream of data starts from what has been learned so far, rather than starting from arbitrary settings that do not relate to the underlying distributions.

10.5.2 OLDA algorithm with data-driven semantic embedding

An overview of the proposed OLDA algorithm with semantic embedding is shown in Algorithm 8. In addition to the text streams, $S^{(t)}$, the algorithm takes as input the sliding history window size δ, weight vector $\boldsymbol{\omega}$, and fixed Dirichlet values, a and b, for initializing the priors α and β, respectively, at time slice 1. Note that b is also used to set the priors of new words that appear for the first time in any time slice. The output of the algorithm is the generative models and the evolution matrices $\mathbf{B}_k$ for all topics.

Algorithm 8 – OLDA with semantic embedding

```
INPUT: $b$; $a$; $\delta$; $\boldsymbol{\omega}$; $\Delta$; $S^{(t)}$, $t = \{1, 2, 3 \ldots\}$
$t = 1$
loop
    New text stream $S^{(t)}$ is received after time delay equal to $\Delta$
    if $t = 1$ then
        $\boldsymbol{\beta}_k^{(t)} = b$, $k \in \{1, \ldots, K\}$
    else
```

```
            $\boldsymbol{\beta}_k^t = \mathbf{B}_k^{t-1}\boldsymbol{\omega}$, $k \in \{1, \ldots, K\}$
        end if
        $\boldsymbol{\alpha}_d^{(t)} = a$, $d = 1, \ldots, D^{(t)}$
        initialize $\Phi^{(t)}$ and $\theta^{(t)}$ to zeros
        initialize topic assignment, $\mathbf{z}^{(t)}$, randomly for all word tokens in $S^{(t)}$
        $[\Phi^{(t)}, \Theta^{(t)}, \mathbf{z}^{(t)}] = \text{GibbsSampling}(S^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\alpha}^{(t)})$
        if $t < \delta$ then
            $\mathbf{B}_k^t = \mathbf{B}_k^{(t-1)} \cup \hat{\Phi}_k^{(t)}$, $k \in \{1, \ldots, K\}$
        else
            $\mathbf{B}_k^t = \mathbf{B}_k^{(t-1)}(1 : W^{(t)}, 2 : \delta) \cup \hat{\Phi}_k^{(t)}$, $k \in \{1, \ldots, K\}$
        end if
    end loop
```

10.5.3 Experimental design

LDA with semantic embedding is evaluated in the problem domain of document modeling. *Perplexity* is a canonical measure of goodness that is used in language modeling. It evaluates the generalization performance of the model on previously unseen documents. Lower perplexity means a better generalization performance and, hence, a better estimation of density. Formally, for a test set of M documents, the perplexity is (Blei et al. 2003)

$$perplexity(D_{test}) = \exp\left\{-\frac{\sum_{d=1}^{M} \log p(\mathbf{w}_d)}{\sum_{d=1}^{M} N_d}\right\}. \tag{10.11}$$

We tested OLDA under different configurations of historic semantic embedding. A summary of the conducted models and their parameter settings are listed in Table 10.5. The window size, δ, was set to values from 0 to 5. The OLDA model with history window of size 0 ignores the history and processes the text stream using a fixed symmetric Dirichlet prior. Under such a model, the estimation is influenced by the semantics of the current stream only. This model, named OLDAFixed, and the OLDA model with $\delta = 1$ are considered as baselines to which the rest of the tested models are compared. To compute the perplexity at every time instance, the documents of the next stream are used as the test set of the model currently generated.

All models were run for 500 iterations and the last sample of the Gibbs sampler was used for evaluation. The number of topics, K, is fixed across all the streams. K, a, and b are set to 50, $50/K$, and 0.01, respectively. All experiments are run on a 2 GHz Pentium M-processor laptop using the MATLAB Topic Modeling Toolbox, authored by Mark Steyvers and Tom Griffiths.[2]. The two datasets used in our experiments for the OLDA model with historic semantic embedding are described below.

[2] The Topic Modeling Toolbox is available at: http://psiexp.ss.uci.edu/research/programs data/toolbox.htm

Table 10.5 Name and parameter settings of OLDA models. The * indicates that the model was applied on the data.

Reuters	NIPS	Model name	δ	ω
*	*	OLDAFixed	0	NA($\beta = 0.05$)
*	*	$1/\omega(1)$	1	1
*	*	$2/\omega(1)$	2	1, 1
*		$2/\omega(0.8)$	2	0.2, 0.8
*	*	$2/\omega(0.7)$	2	0.3, 0.7
*	*	$2/\omega(0.6)$	2	0.4, 0.6
*	*	$2/\omega(0.5)$	2	0.5, 0.5
*	*	$3/\omega(1)$	3	1, 1, 1
*	*	$3/\omega(0.8)$	3	0.05, 0.15, 0.8
*	*	$3/\omega(0.7)$	3	0.1, 0.2, 0.7
*		$3/\omega(0.6)$	3	0.15, 0.25, 0.6
*	*	$3/\omega(0.33)$	3	0.33, 0.33, 0.34
*	*	$4/\omega(1)$	4	1, 1, 1, 1
	*	$4/\omega(0.9)$	4	0.01, 0.03, 0.06, 0.9
*		$4/\omega(0.8)$	4	0.03, 0.07, 0.1, 0.8
*	*	$4/\omega(0.7)$	4	0.05, 0.1, 0.15, 0.7
*		$4/\omega(0.6)$	4	0.05, 0.15, 0.2, 0.6
*	*	$4/\omega(0.25)$	4	0.25, 0.25, 0.25, 0.25
	*	$5/\omega(1)$	5	1, 1, 1, 1, 1
	*	$5/\omega(0.7)$	5	0.05, 0.05, 0.1, 0.15, 0.7
	*	$5/\omega(0.6)$	5	0.05, 0.1, 0.15, 0.2, 0.6
*	*	$5/\omega(0.2)$	5	0.2, 0.2, 0.2, 0.2, 0.2

Reuters-21578.[3] The corpus consists of newswire articles classified by topic and ordered by their date of issue. There are 90 categories with some articles classified in multiple topics. For our experiments, only articles with at least one topic were kept for processing. For data preprocessing, stop words were removed while the remaining words were down-cased and stemmed to their root source. The resulting dataset consists of 10 337 documents, 12 112 unique words, and a total of 793 936 word tokens. For simplicity, we partitioned the data into 30 slices and considered each slice as a stream.

NIPS dataset.[4] The NIPS set consists of the full text of 13 years of the proceedings from 1988 to 2000 of the Neural Information Processing Systems (NIPS) Conference. The data was preprocessed for down-casing, removing stop words and numbers, and removing those words appearing less than five times in the corpus. The dataset contains 1740 research papers, 13 649 unique words, and 2 301 375 word tokens in total. The set is divided into 13 streams based on the year of publication.

[3] The original dataset is available to download from the UCI Knowledge Discovery in Databases Archive: http://archive.ics.uci.edu/ml/.

[4] The original dataset is available at the NIPS Online Repository: http://nips.djvuzone.org/txt.html.

10.5.4 Experimental results

Wikipedia-influenced LDA was run on nine subsets of the Reuters dataset which correspond to the first nine streams. The perplexity of a model was computed using the successive stream as the test set. Figure 10.3 shows the perplexity of Wikipedia-influenced LDA compared to the corresponding models that were trained on the Reuters documents only. It can be seen that the perplexity of LDA with Wikipedia articles is lower in five out of the nine models. We believe that the higher perplexity in some cases with Wikipedia is due to the unstructured approach used to partition the data, which does not guarantee the representation of all the classes in each stream. Thus, any document in the test set that belongs to a new class would eventually increase the perplexity. However, when this factor is neutralized, incorporating external knowledge from Wikipedia does improve the performance.

To test the data-driven semantic embedding, OLDA was first run on the Reuters dataset. It was found that by increasing the window size, δ, OLDA resulted in lower perplexity than the baselines. Figure 10.4 plots the perplexity of OLDA and OLDAFixed at every stream of Reuters under different settings of window size, δ, and the weight vector, ω, was fixed on $1/\delta$. The figure clearly shows that embedding semantics enhanced the document modeling performance. In addition, incorporating semantics from more models, i.e. using a window size greater than 1, further improves the perplexity with respect to OLDA with short memory ($\delta = 1$).

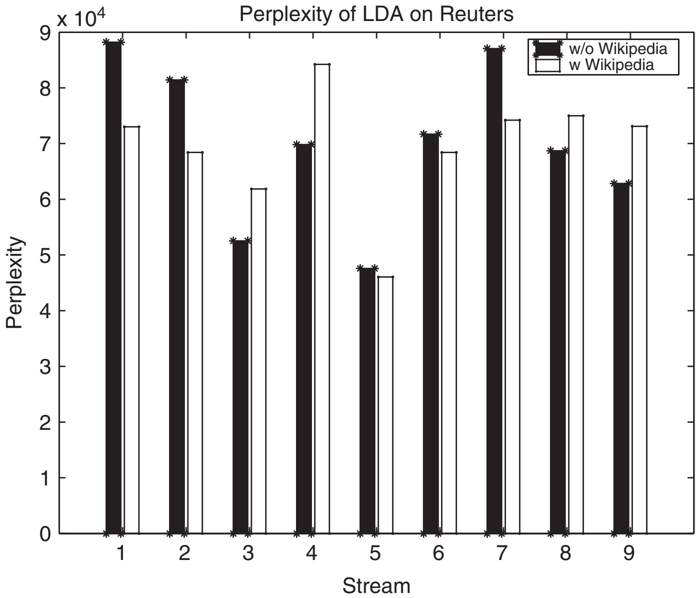

Figure 10.3 Perplexity of OLDA on Reuters with and without Wikipedia articles.

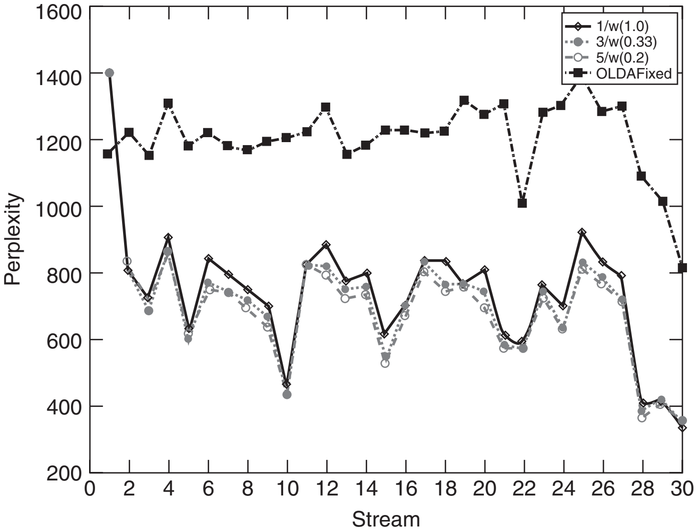

Figure 10.4 Perplexity of OLDA on Reuters for various window sizes compared to OLDAFixed.

Testing with NIPS resulted in a slightly different behavior. When ω was fixed, increasing the window size did show a reduction in the model's perplexity, compared to OLDA with short memory. This is illustrated in Figure 10.5. The larger the window, the lower the perplexity of the model. Nonetheless, the OLDA model only showed improvements with respect to OLDAFixed when the window size was larger than 3. In addition to the window size, previous experiments on NIPS suggested the effect of the total weight of history in estimating the topical semantics of heterogeneous and fast evolving domains like scientific research (AlSumait et al. 2008). The experiments explained next provide evidence of such a justification. Nonetheless, it is worth mentioning here that the OLDA model outperforms OLDAFixed in its ability to automatically detect and track the underlying topics.

To investigate the role of the total history weight, we tested OLDA on NIPS and Reuters under a variety of ω settings. Figure 10.6 shows the average perplexity of OLDA with δ fixed at 2 and the total sum of ω set to 0.05, 0.1, 0.15, 0.2, and 1 for both datasets. Both baselines, OLDAFixed and OLDA with short memory, are also shown. We found that the contribution of history in NIPS is completely opposite to that in Reuters. While increasing the weight for history resulted in a better topical description of Reuters news, lower perplexities were reported with NIPS only for topic models that assign a lower weight for history. In fact, the history weight and perplexity in NIPS (Reuters) are negatively (positively) correlated.

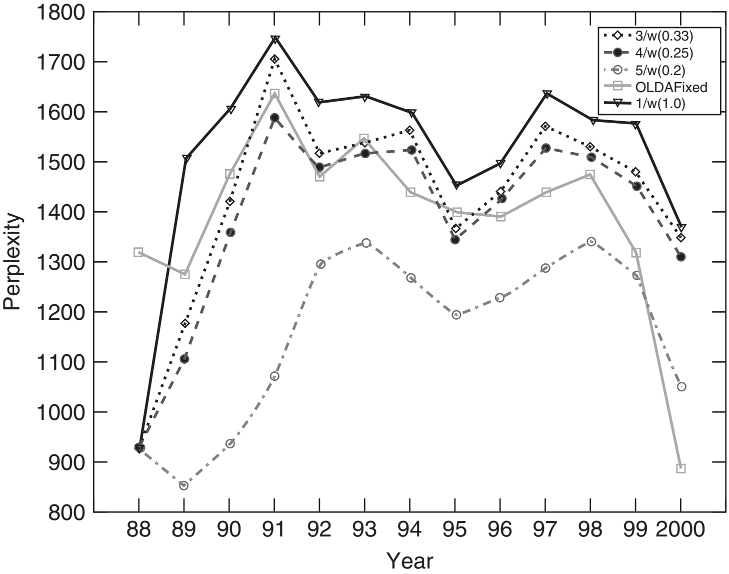

Figure 10.5 Perplexity of OLDA on NIPS for various window sizes compared to OLDAFixed.

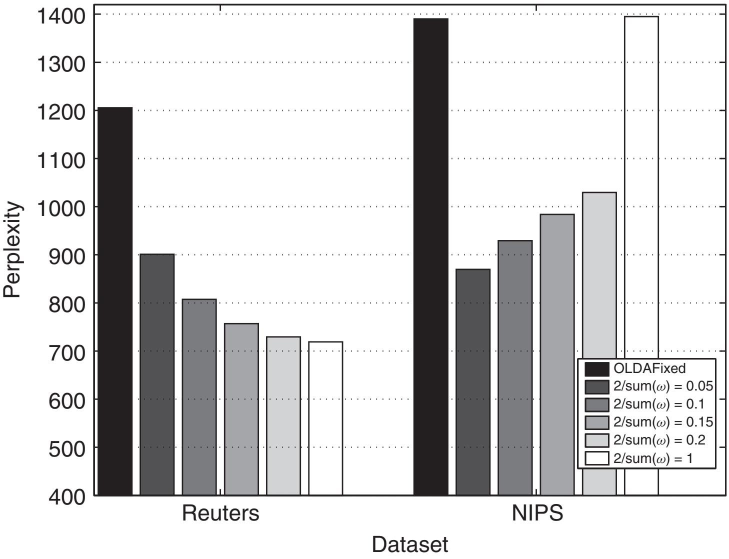

Figure 10.6 Average perplexity of OLDA on Reuters and NIPS under different weights of history contribution compared to OLDA with fixed β.

Reuters' documents span a short period of time while the streams of NIPS are yearly based. As a result, the Reuters' topics are homogeneous and more stable. So, letting the current generative model be heavily influenced by the past topical structure will eventually result in a better description of the data. On the other hand, although there is a set of predefined publication domains in NIPS, like algorithms, applications, and visual processing, these topics are very broad and interrelated. Furthermore, research papers usually cover more topics and continuously introduce novel ideas and topics. Hence, the influence of previous semantics should not exceed the topical structure of the present.

10.6 Related work

The problem of embedding semantic information within the document representation and/or distance metrics has recently been investigated intensively in the domain of text classification and clustering (e.g. AlSumait and Domeniconi (2008), Cristianini et al. (2002)). However, the problem of embedding semantic information within the generative model and the inference process of LDA topic modeling is a new research area. Very recently (Andrzejewski et al. 2009), domain knowledge has been implemented in the form of must-link and cannot-link primitives about the word compositions that should have high or low probability in the topics. These primitives are incorporated in LDA using a mixture of Dirichlet tree priors.

A number of papers in the literature have used LDA topic modeling to represent some kind of semantic embedding. In the domain of text segmentation, the work in Sun et al. (2008) used an LDA-based Fisher kernel to measure text semantic similarity between blocks of documents in the form of latent semantic topics that were previously inferred using LDA. The kernel is controlled by the number of shared semantics and word co-occurrences. Phrase discovery is another area that aims at identifying phrases (n-grams) in text. Wang et al. (2007) presented a topical n-gram model that automatically identified feasible n-grams based on the context that surround it. Moreover, there are some research efforts to incorporate prior knowledge from large universal datasets, like Wikipedia. Phan et al. (2008) built a classifier on both a small set of labeled documents and an LDA topic model estimated from Wikipedia.

10.7 Conclusion and future work

In this chapter, the effect of embedding semantic information in the framework of probabilistic topic modeling is investigated. In particular, static and online LDA topic models are first introduced and two directions to embed semantics within their inference process are defined. The first direction updates the topical structure based on prior knowledge that is learned from Wikipedia. The second approach constructs the parameters based on the topical semantics that have been inferred by the past generated models.

This work can be extended in many directions. LDA with external semantic embedding can be used to build an unsupervised classifier that can effectively group documents based on their content with no need for labeled training documents. In addition, it can be extended to work online on text streams and using an evolving external knowledge. The effect of the embedded historic semantics on detecting emerging and/or periodic topics constitutes future work.

References

AlSumait L and Domeniconi C 2008 Text clustering with local semantic kernels. In *Survey of Text Mining: Clustering, Classification, and Retrieval* (ed. Berry M and Castellanos M) 2nd edn Springer.

AlSumait L, Barbará D and Domeniconi C 2008 Online LDA: Adaptive topic model for mining text streams with application on topic detection and tracking. *Proceedings of the IEEE International Conference on Data Mining*.

AlSumait L, Barbará D and Domeniconi C 2009 The role of semantic history on online generative topic modeling. *Proceedings of the Workshop on Text Mining, held in conjunction with the SIAM International Conference on Data Mining*.

Andrzejewski D, Zhu X and Craven M 2009 Incorporating domain knowledge into topic modeling via Dirichlet forest priors *Proceedings of the International Conference on Machine Learning*.

Blei D, Ng A and Jordan M 2003 Latent Dirichlet allocation. *Journal of Machine Learning Research* **3**, 993–1022.

Cristianini N, Shawe-Taylor J and Lodhi H 2002 Latent semantic kernels. *Journal of Intelligent Information Systems* **18**(2–3), 127–152.

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

Griffiths T and Steyvers M 2004 Finding scientific topics. *Proceedings of the National Academy of Sciences*, pp. 5228–5235.

Heinrich G 2005 *Parameter Estimation for Text Analysis*. Springer.

Hofmann T 1999 Probabilistic latent semantic indexing. *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*.

Mimno D and McCallum A 2007 Organizing the OCA: Learning faceted subjects from a library of digital books. *Proceedings of the Joint Conference on Digital Libraries*.

Minka T and Lafferty J 2002 Expectation-propagation for the generative aspect model. *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*.

Papadimitriou C, Tamaki H, Raghavan P and Vempala S 2000 Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences* **61**(2), 217–235.

Phan X, Nguyen L and Horiguchi S 2008 Learning to classify short and sparse text and web with hidden topics from large-scale data collections. *International WWW Conference Committee*.

Salton G 1983 *Introduction to Modern Information Retrieval*. McGraw-Hill.

Steyvers M and Griffiths T 2005 Probabilistic topic models. In *Latent Semantic Analysis: A Road to Meaning* (ed. Landauer T, McNamara D, Dennis S and Kintsch W) Lawrence Erlbaum Associates.

Story R 1996 An explanation of the effectiveness of latent semantic indexing by means of a Bayesian regression model. *Information Processing and Management* **32**(3), 329–344.

Sun Q, Li R, Luo D and Wu X 2008 Text segmentation with LDA-based Fisher kernels. *Proceedings of the Association for Computational Linguistics*.

Wang X, McCallum A and Wei X 2007 Topical n-grams: Phrase and topic discovery, with an application to information retrieval. *Proceedings of the 7th IEEE International Conference on Data Mining*.

Wei X and Croft B 2006 LDA-based document models for ad-hoc retrieval. *Proceedings of the Conference on Research and Development in Information Retrieval*.

White C 2005 Consolidating, accessing and analyzing unstructured data.

Wikipedia 2009 Wikipedia: The free encyclopedia.