NAIST-IS-DD0061024

Doctoral Dissertation

Distributional Approaches to Natural Language Processing

Daichi Mochihashi

March 24, 2005

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of SCIENCE

Daichi Mochihashi

Thesis Committee:

Professor Yuji Matsumoto	(Supervisor)
Professor Shin Ishii	(Co-supervisor)
Professor Kiyohiro Shikano	(Co-supervisor)
Associate Professor Kentaro Inui	(Co-supervisor)

Distributional Approaches to Natural Language Processing *

Daichi Mochihashi

Abstract

Natural language is a complex and compound organization that structures basic linguistic elements to represent various meanings. Therefore, to understand the nature of natural language, we need a sophisticated treatment of the basic elements as well as the insights about how these elements will be structured.

In the words of statistical natural language processing, we need a sophisticated statistical model of the basic elements, such as words or phrases, to be combined with the structural modeling such as syntactic parsing or dependency analysis. Since the basic property of these elements is considered common over the whole corpora, we need to model their distributional property over the corpora by a statistical learning approach.

In this dissertation, I focus on the statistical learning approach to the distributional modeling of basic elements by considering two kinds of distributional units, that is, static units and dynamic units. Along these two units, I propose novel treatments of distributional units that are different from the methods used in natural language processing so far.

In Chapter 2, I consider static units. Static units are the units we know *a priori* to be effective for a specific kind of task in hand; for example, documents, paragraphs, or sentences.

For the treatment of these units, specifically, measuring a distance between two units, kernel methods have been adopted recently. However, not all the

*Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0061024, March 24, 2005.

natural language processing tasks fall easily into kernelization. In these cases, an Euclidean distance, known also as a cosine distance, has been used quite often for many natural language processing tasks, such as information retrieval, text segmentation, or candidate enumeration in Question Answering, with the tf.idf feature weighting in a preprocessing stage.

In contrast, this chapter proposes an optimal metric distance in place of the Euclidean distance from the cluster information we know beforehand as "similar", by a semi-supervised learning approach. This metric is computed analytically as a solution to the quadratic optimization problem that minimizes the distortion of data distribution in the clusters when measured by the metric distance we wish to obtain.

Experimental results on the retrieval and clustering task for documents and sentences showed consistent performance improvements over the Euclidean distance that has been used so far. As opposed to the similar method recently proposed in machine learning, this method has an advantage in that it computes the optimal metric using the whole data at once without any iterative optimization.

In Chapter 3, I consider dynamic units. Lately, some probabilistic text models for words and documents have been proposed. However, all of them model a text as a bag-of-words, that is, assume a complete exchangeability of words in a document. While these models have been also applied to context modeling in a statistical long-distance language modeling recently, that assumption does not hold true for dynamic context modeling or document modeling with long and heterogeneous semantic content often met in the actual situation.

This chapter focuses on the context modeling to propose a novel Bayesian long-distance statistical language model that makes an optimal prediction of next word based on an online probabilistic inference of appropriate dynamic units of semantic homogeneity, by an unsupervised learning approach.

First, we view context shift as a latent stochastic process to apply a Mean shift model known in statistics. This model is essentially a nonlinear HMM that cannot be decoded by traditional Baum-Welch algorithm or Kalman Filters. For this purpose, we used a multinomial Particle Filter, a sequential Monte Carlo method that has been used mainly in signal processing or robotics research, to

estimate its states and parameters sequentially.

While this model is an extension to a DNA sequence modeling recently proposed by Chen and Lai (2003) in statistics, naïve application to natural language raises problems as to the extremely large number of symbols (words) and strong semantic correlations between them. Therefore, we extended a multinomial Particle Filter by both LDA and DM, Bayesian text models recently proposed, to incorporate semantic relationships between words and updates hyperparameters that were assumed to be known and fixed in the original modeling. As a result, we give two models, MSM-LDA and MSM-DM: the former tracks changes of mixing distribution of a mixture model in a multinomial topic simplex, and the latter tracks a unigram distribution directly in a word simplex. They recognize topic shifts and their rate sequentially in a Bayesian fashion, to make an optimal prediction of next word by a mixture of different lengths of context sampled by each particle individually.

Experiments on the standard British National Corpus showed consistent perplexity improvements on simple context models that have been used thus far, to give a Bayesian context model with the lowest perplexity in the current state of art.

Though this model is a forward predictive language model, it can be extended in principle using a Monte Carlo forward-backward or to a collection of documents, obviating a unit of "document" that has been assumed for a naïve unit of semantic modeling in natural language processing.

Through the proposed methodological sophistications along the two kinds of units, we expect a natural language processing that deal with distributional and semantic aspects of language more naturally and flexibly.

Keywords:

Metric Learning, Language Model, Bayesian Learning, Unsupervised Learning, Sequential Monte Carlo, Time Series Analysis

Contents

List of Figures

List of Tables

Acknowledgements

I am really indebted to many people for finishing my graduate study in natural language processing at the Nara Institute of Science and Technology.

First of all, I thank to Professor Yuji Matsumoto who accommodated me in the Computational Linguistics laboratory to persue my own interest freely and constantly supported me in various aspects.

Second, I wish to thank for the two professors, Professor Shin Ishii and Professor Naonori Ueda, who led me into a probabilistic modeling and Bayesian methods by only a few words that influenced me greatly. Without these words, I may not have the view that I possess now in natural language processing. I also thank for the co-supervisors, Professor Kiyohiro Shikano and Associate Professor Kentaro Inui, who gave me valuable comments and criticisms about this dissertation.

Prior to coming to NAIST, I received many kindnesses from many people at the University of Tokyo. While I had been exploring extensive range of topics during my undergraduate terms, I am also indebted to many teachers, especially noting Tsuneko Nakazawa at Department of Language and Information Sciences, Kentaro Torisawa at Department of Information Science, and Takashi Ikegami, my former advisor at Department of System Sciences.

Since I came to NAIST, I have been influenced by many active colleagues in the Computational Linguistics laboratory and Theoretical Life-science Laboratory; Satoru Takabayashi, Taku Kudo, Hiroki Tamakoshi, and Shigeyuki Oba, to name a few. Not only are they active on research, their passion on everyday problems enlightened me and made me feel satisfied with being involved in NAIST.

At the same time, privately I have been greatly helped and inspired by the members of the KTYY research group at the University of Tokyo, Komaba,

through the *italk* chat system with which I have been involved. Conversation there covers many topics, sometimes quite academic, that often helped about my technical problems and facilicated my research very much as well as relieving me of the solitudes that I often faced.

After coming to ATR, I have been in a completely satisfactory research environment that has been distilled by ATR people and its tradition. I deeply express my gratitude to the members of Spoken Language Translation Research Laboratories and the director Seiichi Yamamoto, especially for the department head Genichiro Kikui who picked me up to join into SLT and consistently helped and discussed with me about various topics. Without these warm and friendly environment, I could not continue my research and deepen my interest and knowledge as I come to possess now.

Finally, I would like to thank for my friends and my family who constantly supported me. Since I entered the university especially, I strongly feel their friendships and hospitalities, without which I cannot survive alone. Wish they may be pleased with my finishing this dissertation.

Chapter 1

Introduction

Natural language is a compound and complex organization that human have been developing for communication.

We notice here that the communication through natural language involves a vast amount of hidden *background knowledge* behind a specific linguistic expression just uttered. Because natural language has huge amount of words or phrasal chunks that have strong semantic correlations but also have subtle important differences, selecting one of them automatically implies many information about the other words that were not in use but influenced the selection of that word. We generally understand linguistic expressions considering whole such information: thus the description of these background knowledge is indispensable to understand the nature of natural language.

However, enumerating whole such information by hand is practically impossible because human intervention inevitably leads to subjective decisions and limited coverage, owing to the large cost of its construction. Moreover, informative background knowledge is often needed about the new and specific words that cannot be covered with the human work and static information.

Instead, we can approach to the background knowledge by a statistical inference from corpora, because texts in the corpora will surely reflect these background knowledge in their generation. In fact, information extracted only from the statistics of corpora is shown to match well with our intuition of semantic knowledge (Griffiths and Steyvers, 2002; Landauer and Dumais, 1997). Therefore, the objective of this research in natural language processing is to make a

sophisticated statistical inference of linguistic background knowledge that can be integrated with the surface analyzes.

From the perspective of learning from corpora, these background knowledge of words or phrasal chunks, henceforth referred simply as "words," will be extracted from their global distribution, that is, usage patterns of words over the whole corpus because the background knowledge is considered the same and ubiquitous to some extent.[1] Since its property is determined through the global distribution over the whole data, we call this kind of model of background knowledge a *distributional model*. On the other hand, hierarchical or sequential syntactic structure builds on the local scope of language, such as sentences or phrases, to organize the information from each word to constitute a complex meaning. Since its property determines the structure of local scope of language, we call this kind of model a *structural model*.

Generally, structural model of sentences or documents builds on the distributional model of words, though the connection still remains weak in the current state of art. While "distribution" is partly determined by the units constrained by the structural model thus the connection is reciprocal, we still note that the distributional model is more fundamental to natural language and requires careful modeling efforts. For example, an utterance "Tiger!" or "No." has no syntactic structure but has apparent meanings; moreover, complex structural meanings are sometimes crystallized into a single word such as "unthinkable" (Sapir, 1921).

Therefore, distributional models must be explored first to be combined with the structural models to make themselves adequate to represent meanings. The relationship between the two models are shown graphically in Figure 1.1. Two models are complementary devices to represent meanings.

Learning in Distributional models The advent of extensive computational resources in 1990s opened up a statistical learning approach that enabled a robust and complex modeling of natural language phenomena.

Particularly, we saw intensive explorations of structural models by statistical

[1]Of course, these background knowledge is different from person to person because of his subjective view of language; however, this difference will never be complete that prevents effective communication. Here, we assume that such personal differences are concentrated on the difference of the corpus that he will encounter.

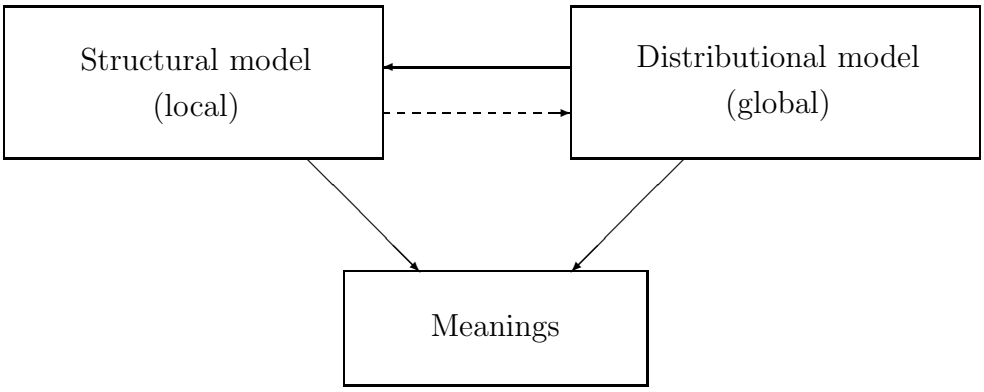


Figure 1.1. Distributional and Structural Models to Represent Meanings.

approaches, such as syntactic parsing, dependency analysis, phrase chunking, and labeling problems like part-of-speech tagging as a fundamental technique on which structural models build themselves.

However, statistical modeling of distributional models is yet in its dawn. In fact, structural modelings mentioned above often use a deterministic approximations of distributional properties using such as dictionaries, thesauri, or even a mere exact matching of words. Besides the problem of ingenious combination of two models, we notice that the distributional models still have naïve and intuitive assumptions that require more sophistication.

For example, LSI (Deerwester et al., 1990), PLSI (Hofmann, 1999), and LDA (Blei et al., 2001) are considered such distributional approaches. However, they all use a simple unit of "document" and make an assumption the the words were generated independent and identically distributed (i.i.d.) given a document; for long enough documents that we often encounter, apparently this assumption is not always true. Moreover, the space where documents and words are embedded in these models is often assumed to be isotropic and orthogonal. Of course this is, again, a simplifying assumption that is to be investigated.

Because of such deficiencies of current distributional models, their approximation error will influence structural models when combined and the other upper-layered natural language processing tasks such as text classification, information retrieval, or candidates generation in Question Answering, that utilize

distributional models often bypassing computationally intensive structural models. Therefore, as well as structural models, distributional models require more elaboration that includes less assumptions and reflect data distributions more precisely.

1.1. Goal of this dissertation

For this purpose, this dissertation first notices that there are two kinds of semantic units in distributional models: (a) static units and (b) dynamic units; and propose refinements along these units.

Static units are the units for a specified task where i.i.d. property within the unit is given *a priori*; for example, documents, paragraphs, or sentences suitable for the task at hand. Since the units are already given, here we will find a more appropriate semantic treatment of these units: specifically, an effective comparison method of two units beyond the conventional one combining with our prior knowledge of "similarity" as a *semi-supervised learning*.

Dynamic units are the units that are recognized *a posteriori* as semantically coherent through a strict statistical inference. This kind of units are considered to be implicitly embedded in static units described above, though they are difficult to find by a merely superficial analysis and requires intensive modeling efforts. I attack this problem by introducing an explicit statistical model of natural language that generates heterogeneous semantic content. I solve an online inference problem to find the hidden units of partial exchangeability in the long-distance language modeling framework to make an optimal prediction based on these units by an *unsupervised learning*.

1.2. Technical Contributions and Outline of Dissertation

Contributions and the outline of this dissertation are therefore as follows.

In Chapter 2, some preliminary constructions are discussed that are necessary to understand the proposed method in Chapter 4.

In Chapter 3, static units are considered. Static units are the units we know *a priori* to be effective for a specific kind of task in hand; for example, documents,

paragraphs, or sentences.

For the treatment of these units, specifically, measuring a distance between two units, kernel methods have been adopted recently. However, not all the natural language processing tasks fall easily into kernelization. In these cases, an Euclidean distance, known also as a cosine distance, has been used quite often for many natural language processing tasks, such as information retrieval, text segmentation, or candidate enumeration in Question Answering, with the tf.idf feature weighting in a preprocessing stage.

In contrast, this chapter proposes an optimal metric distance in place of the Euclidean distance from the cluster information we know beforehand as "similar", by a semi-supervised learning approach. This metric is computed analytically as a solution to the quadratic optimization problem that minimizes the distortion of data distribution in the clusters when measured by the metric distance we wish to obtain.

Experimental results on the retrieval and clustering task for documents and sentences showed consistent performance improvements over the Euclidean distance that has been used so far. As opposed to the similar method recently proposed in machine learning, this method has an advantage in that it computes the optimal metric using the whole data at once without any iterative optimization.

In Chapter 4, dynamic units are considered. Lately, some probabilistic text models for words and documents have been proposed. However, all of them model a text as a bag-of-words, that is, assume a complete exchangeability of words in a document. While these models have been also applied to context modeling in a statistical long-distance language modeling recently, that assumption does not hold true for dynamic context modeling or document modeling with long and heterogeneous semantic content often met in the actual situation.

This chapter focuses on the context modeling to propose a novel Bayesian long-distance statistical language model that makes an optimal prediction of next word based on an online probabilistic inference of appropriate dynamic units of semantic homogeneity, by an unsupervised learning approach.

First, we view context shift as a latent stochastic process to apply a Mean shift model known in statistics. This model is essentially a nonlinear HMM that

cannot be decoded by traditional Baum-Welch algorithm or Kalman Filters. For this purpose, we used a multinomial Particle Filter, a sequential Monte Carlo method that has been used mainly in signal processing or robotics research, to estimate its states and parameters sequentially.

While this model is an extension to a DNA sequence modeling recently proposed by Chen and Lai (2003) in statistics, naïve application to natural language raises problems as to the extremely large number of symbols (words) and strong semantic correlations between them. Therefore, we extended a multinomial Particle Filter by both LDA and DM, Bayesian text models recently proposed, to incorporate semantic relationships between words and updates hyperparameters that were assumed to be known and fixed in the original modeling. As a result, we give two models, MSM-LDA and MSM-DM: the former tracks changes of mixing distribution of a mixture model in a multinomial topic simplex, and the latter tracks a unigram distribution directly in a word simplex. They recognize topic shifts and their rate sequentially in a Bayesian fashion, to make an optimal prediction of next word by a mixture of different lengths of context sampled by each particle individually.

Experiments on the standard British National Corpus showed consistent perplexity improvements on simple context models that have been used thus far, to give a Bayesian context model with the lowest perplexity in the current state of art.

Though this model is a forward predictive language model, it can be extended in principle using a Monte Carlo forward-backward or to a collection of documents, obviating a unit of "document" that has been assumed for a naïve unit of semantic modeling in natural language processing.

Chapter 2

Preliminaries

Before proceeding to the actual approaches, this chapter provides some preliminary theoretical constructions that are closely connected to the contents of Chapter 4. This chapter is irrelevant for understanding Chapter 3; thus readers in haste may skip this chapter and return here again when necessary. The outline of this chapter is as follows.

First, we introduce the statistical language modeling framework in natural language processing, with a special emphasis on the Bayesian n-gram smoothing method that is closely related to the proposed method of context estimation in Chapter 4.

Second, we introduce three probabilistic text modelings, PLSI, LDA, and DM, and their parameter estimation schemes that lay the groundwork for our extension of multinomial filtering with semantic correlations and also used by the online hyperparameter updates of the proposed filtering algorithm.

Third, we briefly describe the Particle Filter, a sequential Monte Carlo method for nonlinear Bayesian estimation. Using the general theory of Particle Filter, we derive an online inference algorithm of context modeling in Chapter 4.

2.1. Statistical Language Modeling

Statistical language model in natural language processing is a general model that gives a joint probability $p(\mathbf{w})$ of the word sequence $\mathbf{w} = w_1 w_2 \cdots w_T$.

Here, we note that the actual definition of $\mathbf{w}$ is arbitrary: $\mathbf{w}$ may be a phrase, sentence, documents, or even all of the huge sequence of training text stream.

Statistical language model is useful many situation; for example, speech recognition, machine translation and information retrieval all utilize statistical language models. Moreover, it is also necessary for human interfaces or OCR, and has a close relationship to Shannon games, information theory and text compression. Therefore, statistical language model constitutes a basic foundation in natural language processing.

Approaches to statistical language model is practically tackled by two approaches: conditional modelings and "whole sentence" approaches.

Conditional modeling decomposes $p(\mathbf{w}) = p(w_1 \cdots w_T)$ using Bayes' formula:

$$p(w_1 \cdots w_t) = \prod_{t=1}^{T} p(w_t | w_{t-1} w_{t-2} \cdots w_1) \tag{2.1}$$

$$\simeq \prod_{t=1}^{T} p(w_t | \underbrace{w_{t-1} \cdots w_{t-(n-1)}}_{(n-1) \text{ words}}) \tag{2.2}$$

$$= \begin{cases} \prod_{t=1}^{T} p(w_t | w_{t-1}, w_{t-2}) & : \text{Trigram } (n=3) \\ \prod_{t=1}^{T} p(w_t | w_{t-1}) & : \text{Bigram } (n=2) \\ \prod_{t=1}^{T} p(w_t) & : \text{Unigram } (n=1). \end{cases} \tag{2.3}$$

In expression (2.2), we make a simplifying assumption that the appearance of a word only depends on its $(n-1)$ precedents: with $n = 3, 2, 1$ frequently adopted, this model is called a trigram, bigram, and unigram models, respectively. On the other hand, "whole sentence" methods model $p(\mathbf{w})$ directly often by a maximum entropy (log-linear) models (Rosenfeld, 1996; Rosenfeld, 1997). However, these methods usually have dependence of conditional models, namely, n-grams, that provides a basic model to be further improved by the maximum entropy approach.

When we constrain ourselves to the n-gram models that is related to Chapter 4, the main difficulty of n-gram models is widely acknowledged to lie in the data sparseness problem to be explained below.

2.1.1 N-gram Model Smoothing

Let us consider only the bigrams for simplicity in the following arguments. When we estimate the probability $p(w_t|w_{t-1})$, the maximum likelihood estimate

$$\hat{p}(w_t|w_{t-1}) = \frac{n(w_{t-1}w_t)}{n(w_{t-1})} \tag{2.4}$$

where $n(x)$ means the count of occurrence of a sequence x, makes most of the bigram probabilities zero or unreliable estimates. Since the possible space of bigrams has a quadratic order of the number of the lexicon, that usually amounts to some hundreds of millions at least, hat often prevents the naïve use of maximum likelihood estimate. Therefore, some ingenious ways to smoothing is necessary for n-gram modeling.

This problem of n-gram smoothing is widely known and has been extensively investigated in natural language processing and speech recognition research. Currently, two kinds of smoothing methods are widely accepted and used: Good-Turing smoothing (Good, 1953) and Kneser-Ney smoothing (Kneser and Ney, 1995). They both use some binning of n-grams based on its frequency to provide an intricate but better smoothing estimate than the maximum likelihood estimate. However, the binning of n-grams used by these algorithms inevitably discards some individual information of n-grams except for the binned frequencies to leave some room for improvement.

2.1.2 Dirichlet Smoothing of N-gram

As opposed to these methods of binning, MacKay (1994) proposed a more theoretically sound smoothing formula by a hierarchical Bayesian method. Contrary to the other smoothing algorithms that make a point estimate of n-gram probabilities, MacKay introduced a probabilistic estimate of the n-gram as a posterior Dirichlet distribution

$$p(\,\cdot\,|w_{t-1}) \sim \mathrm{Dir}(\boldsymbol{\alpha} + n(w_{t-1}\,\cdot\,))\,, \tag{2.5}$$

where $\boldsymbol{\alpha}$ is a hyperparameter of the prior Dirichlet distribution that works as a smoothing term for bigrams and can be optimized by a Newton-Raphson iteration.

Under this model, posterior estimate of the bigram probability is obtained by taking an expectation of (2.5) that yields

$$E[p(w_t|w_{t-1})] = \frac{\alpha(w_t) + n(w_{t-1}w_t)}{\sum_w(\alpha(w) + n(w_{t-1}w))} \tag{2.6}$$

$$= \frac{n(w_{t-1})}{\alpha_0 + n(w_{t-1})}\,\hat{p}(w_t|w_{t-1}) + \frac{\alpha_0}{\alpha_0 + n(w_{t-1})}\,\alpha(w_t) \tag{2.7}$$

$$(\alpha_0 = \sum_w \alpha(w))\,.$$

Equation (2.7) shows that this Bayesian estimate is also considered an adaptive linear interpolation between the maximum likelihood estimate and the corresponding hyperparameter element that means a 'prior count' of Dirichlet distribution.

Although this estimate has not been widely used because of its theoretical complexity of derivation of (2.6) and computational requirements of hyperparameter optimization, some recent research show that this estimate really improves former non-Bayesian modelings that have been adopted so far (Watanabe and Hori, 2003; Chen and Goodman, 1996).

Recently, Dirichlet Mixture (Yamamoto et al., 2003) has been proposed as a text modeling and it can be regarded as a natural extension of Dirichlet smoothing using multiple hyperparameters to weigh them adaptively, as we will show in the next section of text modelings.

2.2. Latent Variable Text Modeling

Lately, some probabilistic models for words and documents have been proposed. While they all use a simple unigram assumption on word appearances, rather they focus on a *semantic* aspects of words and documents, that is, modeling probabilistic correlations between words that appeared in the same document. Since we combine these models with dynamic methods in this dissertation, below we describe three models, PLSI, LDA, and DM in this order, and their parameter estimation algorithms that will be exploited in Chapter 4.

Before proceeding to actual modelings and parameter estimation formulae, we introduce some common notations here to facilitate easy understanding because

they have a common foundation (and in fact, a common training data structure in practice) of "bag of words" assumption.

Notation

$w,\ v$
: A word in the lexicon $1 \dots V$.

$z,\ t,\ m$
: Latent topic variables in $1 \dots M$.

d
: Document index (in PLSI). Range is $1 \dots D$.

$\boldsymbol{p}$
: Multinomial unigram distribution which is itself considered as a random variable of V dimensions.

$\boldsymbol{\lambda}, \theta$
: Multinomial mixture distribution on M-dimensional latent topic variables,

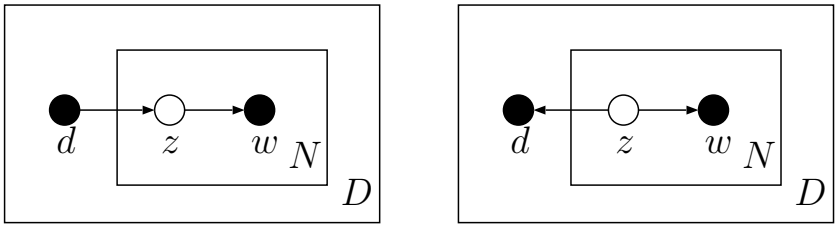
2.2.1 Probabilistic Latent Semantic Indexing (PLSI)

Probabilistic Latent Semantic Indexing (PLSI) (Hofmann, 1999) is a innovative probabilistic text model proposed in Information Retrieval to overcome inherent vector space assumption (Papadimitriou et al., 1998) of Latent Semantic Indexing (Deerwester et al., 1990) to render it into a solid statistical model of texts.

In fact, PLSI is not a strict Bayesian text model because it is based on a maximum likelihood estimate, and lacks a generative property as to new texts. However, it is important because it laid a probabilistic foundation for more strict Bayesian text models as LDA and DM, which we will describe next.

PLSI assumes a following "generative" model for a document $\mathbf{w} = w_1 \dots w_N$ and pseudo document index d (although these notations somewhat differs from original introduction):

1. Draw $d \sim p(d)$.
2. For $n = 1 \dots N$,
    1. Draw $z \sim p(z|d)$.
    2. Draw $w_n \sim p(w|z)$.



(a) Original model (b) Equivalent model

Figure 2.1. Graphical Model of PLSI.

This process corresponds to a graphical model shown in Figure 2.1(a).

Mathematically, this process amounts to a pair of equations

$$p(d, \mathbf{w}) = p(d)p(\mathbf{w}|d) \tag{2.8}$$
$$= p(d) \prod_n p(w_n|d) \tag{2.9}$$
$$= p(d) \prod_n \sum_z p(w_n|d)p(z|d), \tag{2.10}$$

where the number of latent variables M is assumed known.

When we are given a corpus $\mathbf{W} = \mathbf{w}_1 \ldots \mathbf{w}_D$ and an index set $\mathcal{D} = 1 \ldots D$ of pseudo documents. the probability of $\mathbf{W}$ and $\mathcal{D}$ under PLSI is given as

$$p(\mathbf{W}, \mathcal{D}) = \prod_d p(d, \mathbf{w}_d) \tag{2.11}$$
$$= \prod_d p(d) \prod_n \sum_z p(w_n|z)p(z|d). \tag{2.12}$$

In practice, to avoid ancillary index probability $p(d)$, (2.10) is rewritten into equivalent form

$$p(d, w) = \sum_z p(z)p(d|z)p(w|z) \ , \tag{2.13}$$

which has a modified graphical model in Figure 2.1(b).

Under this equivalent form, the probability of corpus $\mathbf{W}$ and ancillary index

set $\mathcal{D}$ is

$$p(\mathbf{W}, \mathcal{D}) = \prod_d p(d, \mathbf{w}_d) \tag{2.14}$$

$$= \prod_d \prod_n p(d, w_n) \tag{2.15}$$

$$= \prod_d \prod_n \sum_z p(z)p(d|z)p(w_n|z) \ . \tag{2.16}$$

That is, in log-likelihood form

$$\log p(\mathbf{W}, \mathcal{D}) = \sum_d \sum_n \log \left(\sum_z p(z)p(d|z)p(w_n|z) \right) \ . \tag{2.17}$$

2.2.1.1 Parameter Estimation of PLSI

Maximization of $\log p(\mathbf{W}, \mathcal{D})$ with respect to PLSI parameters $p(z),\ p(d|z),\ p(w|z)$ can be carried out by a standard procedure of EM algorithm.

Let us define a set of true latent variables Z for each word in each document, and define a free energy F as

$$F = \Big\langle \log p(\mathbf{W}, \mathcal{D}, Z) \Big\rangle_{p(Z|\mathbf{W},\mathcal{D})} \tag{2.18}$$

$$= \sum_d \sum_n \left[\sum_z p(z_{dn}|d, w_{dn}) \left[\log p(z_{dn}) + \log p(d|z_{dn}) + \log p(w_{dn}|z_{dn}) \right] \right] \ . \tag{2.19}$$

Then, by introducing Lagrangian λ to solve for $p(z)$, for example,

$$\frac{\delta}{\delta p(z)} \left[F + \lambda \left(\sum_z p(z) - 1 \right) \right] = 0, \tag{2.20}$$

we get

$$p(z) \propto \sum_d \sum_n p(z_{dn}|d, w_{dn}) = \sum_d \sum_w n(d, w) p(z|d, w) \tag{2.21}$$

where $n(d, w)$ denotes the number of occurrences word w in a document d.

For other parameters $p(d|z)$, $p(w|z)$ we can conduct the same procedure to get an EM algorithm in Figure 2.2 to estimate $p(z)$, $p(d|z)$ and $p(w|z)$ iteratively.

However, PLSI Language Model has a severe overfitting problem especially when applied to a small training data, that is, naïve use of the EM algorithm in Figure 2.2 usually falls into a overfitting of its parameters. To avoid this, Hofmann (1999) proposed a Tempering EM approach where an inverse temparature β is

E step

$$p(z|d,w) \propto p(z)p(d|z)p(w|z) \tag{2.22}$$

M step

$$p(w|z) \propto \sum_d n(d,w)p(z|d,w) \tag{2.23}$$

$$p(d|z) \propto \sum_z \sum_w n(d,w)p(z|d,w) \tag{2.24}$$

$$p(z) \propto \sum_d \sum_w n(d,w)p(z|d,w) \tag{2.25}$$

Figure 2.2. EM algorithm for PLSI.

introduced and set it gradually smaller (higher temparature) at a performance degradation on a held out set, and new β is used thereafter to avoid overfitting. Nevertheless, this procedure requires additional held out set besides training set, and moreover, there is no theoretically verified way as to the size of held out data, and a strategy to decrease β gradually.

Latent Dirichlet Allocation is such a probablistic text model that avoids the overfitting problems mentioned above by a natural Bayesian regularization, and has a further strict generative property as opposed to PLSI which requires a pseudo index set $\mathcal{D}$.

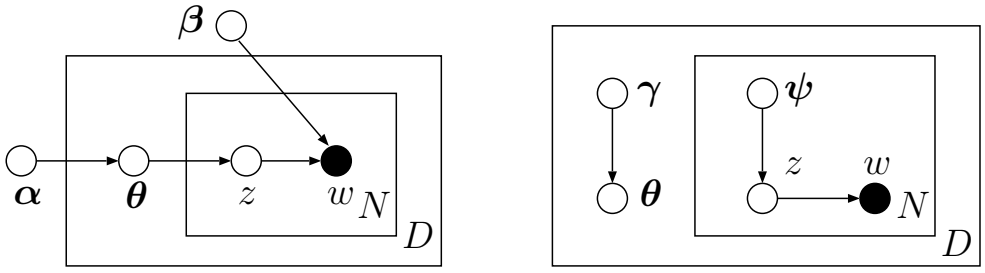
2.2.2 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (Blei et al., 2001; Blei et al., 2003) is a Bayesian extension of PLSI, and can be viewed as a strict generative model of corpora.

LDA also assumes latent topic variables $z \in Z = \{1 \dots M\}$ to hypothesize that the words $\mathbf{w} = w_1 \dots w_N$ in a document have been generated by a mixture of unigrams $\boldsymbol{\beta} = \{p(w|z)\}$ $(z \in Z)$ with a multinomial mixture distribution θ on Z, that is,

$$p(\mathbf{w}|\boldsymbol{\beta},\theta) = \prod_n \sum_m p(w_n|z_m)\theta_m = \prod_n \sum_m \beta_{mn}\theta_m \tag{2.26}$$

where $\beta_{nm} = p(w_n|m)$.



(a) Original (b) Variational Approximation

Figure 2.3. Graphical Models of LDA.

In LDA, multinomial θ itself is regarded as a random variable and endowed a prior Dirichlet distribution $p(\theta) = \mathrm{Dir}(\theta|\boldsymbol{\alpha})$.

Therefore, the probability of $\mathbf{w}$ given LDA parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}$ is

$$p(\mathbf{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int p(\theta|\boldsymbol{\alpha}) p(\mathbf{w}|\theta, \boldsymbol{\beta}) d\theta \tag{2.27}$$

$$= \int p(\theta|\boldsymbol{\alpha}) \prod_n \sum_m p(w_n|z_m)\theta_m d\theta \tag{2.28}$$

$$= \frac{\Gamma(\sum_m \alpha_m)}{\prod_z \Gamma(\alpha_m)} \int \left(\prod_m \theta_m^{\alpha_m - 1} \right) \left(\prod_n \sum_m \prod_v (\theta_v \beta_{mv})^{w_n^v} \right) d\theta. \tag{2.29}$$

where w_n^v is an index variable that takes a value 1 when $w_n = v$ else 0. In other words, LDA assumes a following process to generate $\mathbf{w} = w_1 \dots w_N$:

1. Draw $\theta \sim \mathrm{Dir}(\boldsymbol{\alpha})$.
2. For $n = 1 \dots N$,
    1. Draw $z \sim \mathrm{Mult}(\theta)$.
    2. Draw $w_n \sim \mathrm{Mult}(\beta_z)$.

This process has a graphical model shown in Figure 2.3(a).

Because for each document $\mathbf{w}$ there is a latent multinomial θ that is estimated stochastically as a Dirichlet distribution $p(\theta|\mathbf{w}) = \mathrm{Dir}(\theta|\boldsymbol{\alpha})$, as a posterior view LDA in fact "allocates" a Dirichlet distribution $\mathrm{Dir}(\theta|\boldsymbol{\alpha})$ to explain its generative process from the latent topics.[1]

[1]This "allocation" is actually executed through a VB-EM algorithm which will be explained in section 2.2.2.1.

2.2.2.1 Parameter Estimation of LDA

For the corpus $W = \mathbf{w}_1, \ldots, \mathbf{w}_D$, LDA assumes exchangeability of each document $\mathbf{w}_d$ $(d = 1 \ldots D)$ [2] to give a joint probability

$$p(W|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{d=1}^{D} p(\mathbf{w}_d|\boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{2.30}$$

or equivalently,

$$\log p(W|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{d=1}^{D} \log p(\mathbf{w}_d|\boldsymbol{\alpha}, \boldsymbol{\beta}) \,. \tag{2.31}$$

Thus we want to maximize the likelihood (2.31) with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, where $p(\mathbf{w}_d|\boldsymbol{\alpha}, \boldsymbol{\beta})$ is given by (2.29).

However, exact estimation of the parameter $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ according to (2.29) and (2.31) is intractable as opposed to PLSI because of the coupling of the latent variables θ and z as shown in Figure 2.3(a).

One of the strategy to alleviate this problem is to use a variational approximation (Jordan et al., 1999). Using Jensen's inequality, the likelihood (2.31) is lower bounded by a variational distribution q as follows.

$$\log p(W|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{d} \log p(\mathbf{w}_d|\alpha, \beta) \,, \tag{2.32}$$

$$\log p(\mathbf{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int \sum_{z} \log p(\mathbf{w}, z, \theta|\boldsymbol{\alpha}, \boldsymbol{\beta}) \, d\theta \tag{2.33}$$

$$= \int \sum_{z} \log q(z, \theta|\mathbf{w}, \boldsymbol{\gamma}, \boldsymbol{\psi}) \frac{p(\mathbf{w}, z, \theta|\boldsymbol{\alpha}, \boldsymbol{\beta})}{q(z, \theta|\mathbf{w}, \boldsymbol{\gamma}, \boldsymbol{\psi})} \, d\theta \tag{2.34}$$

$$\geq \int \sum_{z} q(z, \theta|\mathbf{w}, \boldsymbol{\gamma}, \boldsymbol{\psi}) \log \frac{p(\mathbf{w}, z, \theta|\boldsymbol{\alpha}, \boldsymbol{\beta})}{q(z, \theta|\mathbf{w}, \boldsymbol{\gamma}, \boldsymbol{\psi})} \, d\theta \tag{2.35}$$

$$= \int \sum_{z} q(z, \theta|\mathbf{w}, \boldsymbol{\gamma}, \boldsymbol{\psi}) \left[\log p(\theta|\boldsymbol{\alpha}) + \sum_{n} \log p(z_n|\theta) \right. \tag{2.36}$$

$$\left. + \sum_{n} \log p(w_n|z_n, \boldsymbol{\beta}) \right] d\theta \,, \tag{2.37}$$

[2] This i.i.d. assumption has been relaxed recently through an introduction of hyperprior to hyperparameter $\boldsymbol{\alpha}$ (Yu et al., 2005); however, this extension is not our current concern, and we do not persue it here.

where $q = q(z, \theta|\mathbf{w}, \boldsymbol{\gamma}, \boldsymbol{\psi})$ is a variational approximation of $p(\mathbf{w}, z, \theta|\boldsymbol{\alpha}, \boldsymbol{\beta})$.[3] When we assume a factorial decomposition of q, dropping a connection between θ and z as graphically shown in Figure 2.3(b), as

$$q(z, \theta|\mathbf{w}, \boldsymbol{\gamma}, \boldsymbol{\psi}) = q(\theta|\boldsymbol{\gamma}) \prod_n q(z_n|w_n, \boldsymbol{\psi}) \,, \tag{2.38}$$

the variational likelihood (2.37) becomes

$$\begin{aligned} \log p(\mathbf{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}) \geq & \Big\langle \log p(\theta|\boldsymbol{\alpha}) \Big\rangle_{q(\theta|\boldsymbol{\gamma})} + \sum_n \Big\langle \log p(z_n|\theta) \Big\rangle_{q(\theta|\boldsymbol{\gamma}),\, q(z_n|w_n,\boldsymbol{\psi})} \\ & + \sum_n \Big\langle \log p(w_n|z_n, \beta) \Big\rangle_{q(z_n|w_n,\boldsymbol{\psi})} \\ & + \Big\langle \log q(\theta|\boldsymbol{\gamma}) \Big\rangle_{q(\theta|\boldsymbol{\gamma})} + \sum_n \Big\langle \log q(z_n|w_n, \boldsymbol{\psi}) \Big\rangle_{q(z_n|w_n,\boldsymbol{\psi})} \end{aligned} \tag{2.39}$$

Thus we come to maximize the variational lower bound (2.39) with respect to the parameters $\boldsymbol{\gamma}, \boldsymbol{\psi}$ and afterwards $\boldsymbol{\alpha}, \boldsymbol{\beta}$. Below, we describe the parameter reestimation formulae of $\boldsymbol{\gamma}, \boldsymbol{\psi}, \boldsymbol{\beta}$, and $\boldsymbol{\alpha}$, in this order.

(a) Maximization w.r.t. $\boldsymbol{\gamma}$ For a document $\mathbf{w}$, we collect the terms containing γ_i from the variational likelihood:

$$\begin{aligned} L[\gamma_i] = & \sum_i (\alpha_i - 1)\Big[\Psi(\gamma_i) - \Psi(\sum_i \gamma_i)\Big] + \sum_n \sum_i \psi_{ni}\Big[\Psi(\gamma_i) - \Psi(\sum_i \gamma_i)\Big] \\ & - \log\Gamma(\sum_i \gamma_i) + \log\Gamma(\gamma_i) - \sum_i (\gamma_i - 1)\Big[\Psi(\gamma_i) - \Psi(\sum_i \gamma_i)\Big] \qquad (2.40) \\ = & \sum_i \Big[\Psi(\gamma_i) - \Psi(\sum_i \gamma_i)\Big]\Big[\alpha_i - \gamma_i + \sum_n \psi_{ni}\Big] - \log\Gamma(\sum_i \gamma_i) + \log\Gamma(\gamma_i) \,. \qquad (2.41) \end{aligned}$$

Therefore, by differenciation

$$\begin{aligned} & \frac{\partial L}{\partial \gamma_i} = 0 \\ & \quad \Longleftrightarrow \Psi'(\gamma_i)\Big[\alpha_i - \gamma_i + \sum_n \psi_{ni}\Big] = \Psi'(\sum_i \gamma_i) \sum_i \Big[\alpha_i - \gamma_i + \sum_n \psi_{ni}\Big] \qquad (2.42) \\ & \therefore \gamma_i = \alpha_i + \sum_n \psi_{ni} \,. \quad \square \qquad (2.43) \end{aligned}$$

[3]Note that $\mathbf{w}$ is given for both p and q: therefore, both functions are the distributions over the latent variables z and θ, given the respective parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}$ and $\boldsymbol{\gamma}, \boldsymbol{\psi}$.

VB-E step:

For a document $d = 1 \dots D$,
For a word $n = 1 \dots N_d$,
For a latent class $i = 1 \dots M$,
$$\psi_{dni} \propto p(w_{dn}|z_i) \exp\Big(\Psi(\gamma_i)\Big) .$$

VB-M step:

$$p(v|z_i) \propto \sum_d \sum_n w_{dn}^v \psi_{dni} .$$

Figure 2.4. VB-EM algorithm for a document in LDA.

(b) Maximization w.r.t. $\boldsymbol{\psi}$ Similarly, collecting terms containing ψ_{ni} and adding a Lagrange multiplier λ gives

$$L[\psi_{ni}] = \psi_{ni}\Big[\Psi(\gamma_i) - \Psi(\sum_i \gamma_i)\Big] + \psi_{ni} \log \beta_{ni} - \psi_{ni} \log \psi_{ni} + \lambda \left(\sum_n \psi_{ni} - 1\right) \quad (2.44)$$

By differenciation,

$$\frac{\partial L}{\partial \psi_{ni}} = \Psi(\gamma_i) - \Psi(\sum_i \gamma_i) + \log \beta_{ni} - (\log \psi_{ni} + 1) + \lambda = 0 \quad (2.45)$$

$$\therefore \ \psi_{ni} \propto \beta_{ni} \exp\Big[\Psi(\gamma_i) - \Psi(\sum_i \gamma_i)\Big] \quad (2.46)$$

$$\propto \beta_{ni} \exp\Big[\Psi(\gamma_i)\Big] . \quad \square \quad (2.47)$$

Since the parameter estimation formulae for $\boldsymbol{\gamma}$ and $\boldsymbol{\psi}$ are intertwined, actual maximization is conducted via a Variational Bayes EM (VB-EM) algorithm of Figure 2.4.

(c) Maximization w.r.t. $\boldsymbol{\beta}$ For a single document $\mathbf{w}$, the likelihood term that contains $\boldsymbol{\beta}$ is

$$L[\beta_{vi}](\mathbf{w}) = \sum_n \sum_v \sum_i w_n^v \psi_{vi} \log \beta_{vi} . \quad (2.48)$$

Therefore, joint likelihood term containing $\boldsymbol{\beta}$ is

$$L[\beta_{vi}] = \sum_d \sum_v \sum_i w_{dn}^v \psi_{dni} \log \beta_{vi} + \lambda\Big(\sum_v \beta_{vi} - 1\Big) , \quad (2.49)$$

giving

$$\frac{\partial L}{\partial \beta_{vi}} = \sum_d \sum_n \sum_v w_{dn}^v \frac{\psi_{dni}}{\beta_{vi}} + \lambda = 0 \tag{2.50}$$

$$\therefore\ \beta_{vi} \propto \sum_d \sum_n \sum_v w_{dn}^v \psi_{dni}\,. \quad \square \tag{2.51}$$

(d) Maximization w.r.t. $\boldsymbol{\alpha}$

$$\langle \log p(\theta|\boldsymbol{\alpha}) \rangle_{q(\theta|\mathbf{w},\boldsymbol{\gamma})} \tag{2.52}$$

$$= \int \left(\log \frac{\Gamma(\alpha_0)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta_i^{\alpha_i - 1} \right) \cdot \frac{\Gamma(\gamma_0)}{\prod_i \Gamma(\gamma_i)} \prod_i \theta_i^{\gamma_i - 1} d\theta \tag{2.53}$$

$$= \log \Gamma(\alpha_0) - \sum_i \log \Gamma(\alpha_i) + \int \sum_i (\alpha_i - 1) \log \theta_i \cdot \frac{\Gamma(\gamma_0)}{\prod_i \Gamma(\gamma_i)} \prod_i \theta_i^{\gamma_i - 1} d\theta \tag{2.54}$$

$$= \log \Gamma(\alpha_0) - \sum_i \log \Gamma(\alpha_i) + \sum_i (\alpha_i - 1)\{\Psi(\gamma_i) - \Psi(\gamma_0)\} \tag{2.55}$$

$$\equiv L_{\alpha_d}\,. \tag{2.56}$$

Likelihood term w.r.t. $\boldsymbol{\alpha}$ to maximize is

$$L_\alpha = \sum_d L_{\alpha_d} \tag{2.57}$$

$$= \sum_d \left[\log \Gamma(\alpha_0) - \sum_i \log \Gamma(\alpha_i) + \sum_i (\alpha_i - 1)\{\Psi(\gamma_{di}) - \Psi(\gamma_{d0})\} \right] \tag{2.58}$$

to get

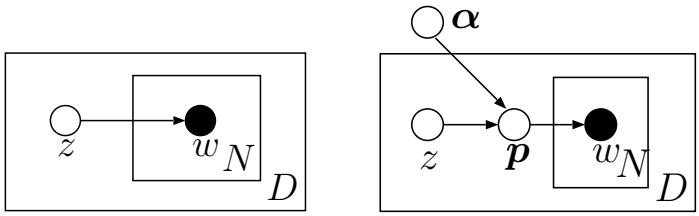
$$\frac{\partial L_\alpha}{\partial L_{\alpha_k}} = M\left(\Psi(\alpha_0) - \Psi(\alpha_k)\right) + \sum_d \left(\Psi(\gamma_{dk}) - \Psi(\gamma_{d0})\right) \tag{2.59}$$

$$\frac{\partial^2 L_\alpha}{\partial \alpha_k \partial \alpha_l} = \begin{cases} M(\Psi'(\alpha_0) - \Psi'(\alpha_k)) & (k = l) \\ M\Psi'(\alpha_0) & (k \neq l) \end{cases} \tag{2.60}$$

Therefore, the Hessian H about $\boldsymbol{\alpha}$ is written as

$$H = M\left[\mathrm{diag}(-\Psi'(\boldsymbol{\alpha})) + \mathbf{1}\Psi'(\sum_i \alpha_i)\mathbf{1}^T \right]. \tag{2.61}$$

Because this Hessian H has a special form $H = \mathrm{diag}(\mathbf{h}) + \mathbf{1}z\mathbf{1}^T$, linear order Newton-Raphson iteration exists to estimate H efficiently (Blei et al., 2003)(Minka, 2000a).



(a) Unigram Mixture (UM) (b) Dirichlet Mixture (DM)

Figure 2.5. Graphical model of UM and DM.

2.2.3 Dirichlet Mixture (DM)

Dirichlet Mixture (Yamamoto et al., 2003) is a novel probabilistic text model that is reported to have the lowest perplexity when being used as a context modeling.

DM is also a nonparametric mixture model that have 'topic' components; however, as opposed to LDA and PLSI, each mixture component covers all the word simplex as a prior Dirichlet distribution of multinomial unigram distribution of words. Therefore, DM is able to model all region of the word simplex (i.e. multinomial manifold) which is necessary for a flexible modeling of contextual distribution.

Specifically, DM assumes a following generative model for a document $\mathbf{w} = w_1 \dots w_N$:

1. Draw $m \sim \text{Mult}(\boldsymbol{\lambda})$.
2. For $n = 1 \dots N$,
    1. Draw $\boldsymbol{p} \sim \text{Dir}(\boldsymbol{\alpha}_m)$
    2. Draw $w_n \sim \text{Mult}(\boldsymbol{p})$.

where $\boldsymbol{p}$ is a V-dimensional unigram distribution and $\boldsymbol{\lambda}$, $\boldsymbol{\alpha}_1 \dots \boldsymbol{\alpha}_M = \boldsymbol{\alpha}$ are M-dimensional multinomial and V-dimensional Dirichlet parameters, respectively. This model is considered as a Bayesian extension of the Unigram Mixture (Nigam et al., 2000) and has a graphical model shown in Figure 2.5.

Mathematically, probability of the corpus $W = \mathbf{w}_1 \dots \mathbf{w}_D$ given the DM pa-

rameters $\boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M$ is

$$\begin{aligned} p(\mathbf{W}|\boldsymbol{\lambda}, \boldsymbol{\alpha}) &= \prod_{d=1}^{D} p(\mathbf{w}_d|\boldsymbol{\lambda}, \boldsymbol{\alpha}) \\ &= \prod_{d=1}^{D} \int p(\mathbf{w}_d|\boldsymbol{p}) \sum_{m=1}^{M} \lambda_m \mathrm{Dir}(\boldsymbol{p}|\boldsymbol{\alpha}_m) d\boldsymbol{p} \\ &= \prod_{d=1}^{D} \left[\sum_{m=1}^{M} \lambda_m \frac{\Gamma(\sum_v \alpha_{mv})}{\Gamma(\sum_v \alpha_{mv} + \sum_v n_{dv})} \prod_{v=1}^{V} \frac{\Gamma(\alpha_{mv} + n_{dv})}{\Gamma(\alpha_{mv})} \right], \end{aligned} \tag{2.62}$$

where n_{dv} is the number of occurrences of word v in a document d. Parameters $\boldsymbol{\lambda}$, $\boldsymbol{\alpha}_1 \ldots \boldsymbol{\alpha}_M$ can be iteratively estimated by a combination of EM algorithm and a Newton-Raphson method, which is a straight extension to the estimation of Polya mixture (Minka, 2000b).

2.2.3.1 Parameter Estimation of Dirichlet Mixture

As we derived above, the probability of $\mathbf{w}_i$ given DM parameters $\boldsymbol{\lambda}, \boldsymbol{\alpha}$ is

$$p(\mathbf{w}_i|\boldsymbol{\lambda}, \boldsymbol{\alpha}) = \sum_{m=1}^{M} \lambda_m \frac{\Gamma(\alpha_{m0})}{\Gamma(\alpha_{m0} + n_i)} \prod_{v=1}^{V} \frac{\Gamma(\alpha_{mv} + n_{iv})}{\Gamma(\alpha_{mv})}. \tag{2.63}$$

Therefore, the whole probability that we want to maximize is

$$L(\mathbf{W}|\boldsymbol{\lambda}, \boldsymbol{\alpha}) = \log p(\mathbf{W}|\boldsymbol{\lambda}, \boldsymbol{\alpha}) = \sum_d \log p(\mathbf{w}_d|\boldsymbol{\lambda}, \boldsymbol{\alpha}). \tag{2.64}$$

This maximization can be done through an EM algorithm. Let the true that generated $\mathbf{w}_i$ be z_i, and the set of z_i $(i = 1..D)$ be Z. We let $p_{im} = p(z_i = m|\mathbf{w}_i, \boldsymbol{\lambda}, \boldsymbol{\alpha})$. Then, by following a standard EM procedure, we must maximize

$$F = \sum_i \Big\langle \log p(\mathbf{W}, Z|\boldsymbol{\lambda}, \boldsymbol{\alpha}) \Big\rangle_{p(Z|\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\alpha})} \tag{2.65}$$

$$= \sum_i \sum_m p_{im} \log p(\mathbf{w}_i, z_i = m|\boldsymbol{\lambda}, \boldsymbol{\alpha}) \tag{2.66}$$

$$= \sum_i \sum_m p_{im} \log \left[\lambda_m \frac{\Gamma(\alpha_{m0})}{\Gamma(\alpha_{m0} + n_i)} \prod_v \frac{\Gamma(\alpha_{mv} + n_{iv})}{\Gamma(\alpha_{mv})} \right] \tag{2.67}$$

$$= \underbrace{\sum_i \sum_m p_{im} \log \lambda_m}_{(a)} + \underbrace{\sum_i \sum_m p_{im} \log \left[\frac{\Gamma(\alpha_{m0})}{\Gamma(\alpha_{m0} + n_i)} \prod_v \frac{\Gamma(\alpha_{mv} + n_{iv})}{\Gamma(\alpha_{mv})} \right]}_{(b)} \tag{2.68}$$

Let the first and second term of (2.68) be (a) and (b), respectively. Therefore, it is sufficient to maximize (a) and (b) iteratively in turn.

(a) Maximization w.r.t. $\boldsymbol{\lambda}$ To maximize F with respect to $\boldsymbol{\lambda}$, we introduce a Lagrangian μ to differenciate it as

$$\frac{\partial}{\partial \lambda_m}\left[(a)+\mu\left(\sum_m \lambda_m - 1\right)\right] = 0 \tag{2.69}$$

to get

$$\sum_i \frac{p_{im}}{\lambda_m} + \mu = 0. \tag{2.70}$$

Therefore,

$$\lambda_m \propto \sum_i p_{im} = \sum_i p(z_i{=}m|\mathbf{w}_i, \boldsymbol{\lambda}, \boldsymbol{\alpha}) \tag{2.71}$$

(b) Maximization w.r.t. $\boldsymbol{\alpha}$ Let (b) be $L(\boldsymbol{\alpha})$. Here, for efficiency we approximate $L(\boldsymbol{\alpha})$ by a LOO (leave-one-out) approximation (Appendix A) as

$$L(\boldsymbol{\alpha}) = \sum_i \sum_m p_{im} \log\left[\frac{\Gamma(\alpha_{m0})}{\Gamma(\alpha_{m0}{+}n_i)}\prod_v \frac{\Gamma(\alpha_{mv}{+}n_{iv})}{\Gamma(\alpha_{mv})}\right] \tag{2.72}$$

$$\simeq \sum_i \sum_m p_{im} \sum_v n_{iv} \log\left(\frac{\alpha_{mv}+n_{iv}-1}{\alpha_{m0}+n_i-1}\right) \tag{2.73}$$

$$= \sum_i \sum_m p_{im} \sum_v n_{iv} \log(\alpha_{mv}+n_{iv}-1) - \sum_i \sum_m p_{im} n_i \log(\alpha_{m0}+n_i-1) \tag{2.74}$$

Here, we introduce two bounds about $\log(x+n)$ and $\log(x)$:

$$\log(x+n) = \log(q\cdot\frac{x}{q} + (1-q)\cdot\frac{n}{1-q}) \tag{2.75}$$

$$\geq q\log x + (1-q)\log n - H(q) \tag{2.76}$$

$$\log x = \log(x_0) + \frac{x-x_0}{x_0} - O(x_0^2) \tag{2.77}$$

$$\leq ax - 1 - \log a\,, \tag{2.78}$$

where $H(x)$ is an entropy function $H(x) = x\log x + (1-x)\log(1-x)$, and $q = x/(x+n)$, $a = 1/x$. Proof is found in Appendix B.

Using these bounds, we get

$$\sum_i \sum_m p_{im} \sum_v n_{iv} \log(\alpha_{mv} + n_{iv} - 1) - \sum_i \sum_m p_{im} n_i \log(\alpha_{m0} + n_i - 1) \tag{2.79}$$

$$\geq \sum_i \sum_m p_{im} \sum_v n_{iv} \Big[q_{iv} \log \alpha_{mv} + (1 - q_{iv}) \log(n_{iv} - 1) - H(q_{iv}) \Big]$$

$$- \sum_i \sum_m p_{im} n_i \Big[a_i(\alpha_{m0} + n_i - 1) - 1 - \log a_i \Big] \tag{2.80}$$

$$= \sum_i \sum_m p_{im} \sum_v n_{iv} q_{iv} \log \alpha_{mv} - \sum_i \sum_m p_{im} n_i a_i \left(\sum_v \alpha_{mv} + n_i - 1 \right) + (\text{const.})$$

$$\equiv f(\boldsymbol{\alpha}). \tag{2.81}$$

Therefore, by differenciating $f(\boldsymbol{\alpha})$ w.r.t. α_{mv} as

$$\frac{\partial f(\boldsymbol{\alpha})}{\partial \alpha_{mv}} = \frac{\sum_i p_{im} n_{iv} q_{iv}}{\alpha_{mv}} - \sum_i p_{im} n_i a_i = 0, \tag{2.82}$$

we get

$$\alpha_{mv} = \frac{\sum_i p_{im} n_{iv} \frac{\alpha_{mv}}{\alpha_{mv} + n_{iv} - 1}}{\sum_i p_{im} \frac{n_i}{\alpha_{m0} + n_i - 1}} \tag{2.83}$$

$$= \frac{\sum_i p_{im} \frac{n_i v}{\alpha_{mv} + n_{iv} - 1}}{\sum_i p_{im} \frac{n_i}{\alpha_{m0} + n_i - 1}} \cdot \alpha_{mv}. \tag{2.84}$$

EM-Newton algorithm of DM

Finally, we need a calculation of $p_{im} = p(z_i = m | \mathbf{w}_i, \boldsymbol{\lambda}, \boldsymbol{\alpha})$.
In fact,

$$p_{im} = p(z_i = m | \mathbf{w}_i, \boldsymbol{\lambda}, \boldsymbol{\alpha}) \tag{2.85}$$

$$\propto p(z_i = m, \mathbf{w}_i | \boldsymbol{\lambda}, \boldsymbol{\alpha}) \tag{2.86}$$

$$= p(\mathbf{w}_i | z_i = m, \boldsymbol{\lambda}, \boldsymbol{\alpha}) p(z_i = m | \boldsymbol{\lambda}, \boldsymbol{\alpha}) \tag{2.87}$$

$$= \lambda_m \cdot \frac{\Gamma(\alpha_{m0})}{\Gamma(\alpha_{m0} + n_i)} \prod_v \frac{\Gamma(\alpha_{mv} + n_{iv})}{\Gamma(\alpha_{mv})} \tag{2.88}$$

E step: Calculate p_{im} by (2.89).
M step: $\lambda_m \propto \sum_i p_{im}$.
For $m = 1 \dots M$,
Maximize $\boldsymbol{\alpha}_m$ given $\boldsymbol{\lambda}$ and p_{im} by (2.84).

Figure 2.6. EM-Newton algorithm of DM.

Therefore, we get

$$p_{im} = \left(\lambda_m \cdot \frac{\Gamma(\alpha_{m0})}{\Gamma(\alpha_{m0}+n_i)} \prod_v \frac{\Gamma(\alpha_{mv}+n_{iv})}{\Gamma(\alpha_{mv})}\right) \Big/ \sum_m \left(\lambda_m \cdot \frac{\Gamma(\alpha_{m0})}{\Gamma(\alpha_{m0}+n_i)} \prod_v \frac{\Gamma(\alpha_{mv}+n_{iv})}{\Gamma(\alpha_{mv})}\right). \tag{2.89}$$

With this formula, we obtain a final EM-Newton algorithm in Figure 2.6.

2.3. Particle Filter

Particle Filter (also known as a sequential Monte Carlo method (SMC)) (Doucet et al., 2001) is a Bayesian filtering algorithm to conduct a Monte Carlo method sequentially.

The main advantage of SMC is that it can accurately track any nonlinear dynamics beyond Normal distributions and discrete distributions where traditional Kalman filters and discrete HMMs have been applied respectively. Although it has been used mainly in signal processing or robotics research lately with an increasing computational resources, thanks to its nonparametric structure it can be applied in principle to the natural language and Dirichlet distributions that we treat in this dissertation.

Below, we briefly prepare the general theory of SMC that will be used in Chapter 4.

Importance Sampling and Sequential Importance Sampling SMC can be considered as an extension of Importance Sampling (for example, (Gilks et al., 1996)), one of the basic algorithm of Monte Carlo methods. Generally, Importance Sampling is used to approximate an intractable integral often met in the

Bayesian expectation $E[f(\mathbf{x})] = \int p(\mathbf{x})f(\mathbf{x})d\mathbf{x}$ of the function $f(\mathbf{x})$ of a random variable $\mathbf{x}$, by a sampling as

$$E[f(\mathbf{x})] = \int p(\mathbf{x})f(\mathbf{x})d\mathbf{x} \tag{2.90}$$

$$= \int q(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}f(\mathbf{x})d\mathbf{x} \tag{2.91}$$

$$\simeq \frac{1}{N}\sum_{i=1}^{N}\frac{p(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})}f(\mathbf{x}^{(i)}) \qquad (\mathbf{x}^{(i)} \sim q(\mathbf{x})) \tag{2.92}$$

$$= \sum_{i=1}^{N} w(\mathbf{x}^{(i)})f(\mathbf{x}^{(i)})\,, \qquad \left(w(\mathbf{x}^{(i)}) = \frac{1}{N}\frac{p(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})}\right) \tag{2.93}$$

where $q(\mathbf{x})$ is called a *proposal distribution* that samples easier than $p(\mathbf{x})$.

Equation (2.93) means that we can obtain $E[f(\mathbf{x})]$ from the N Monte Carlo samples $\mathbf{x}^{(i)} \sim q(\mathbf{x})$ $(i = 1 \dots N)$, weighted accordingly by $w(\mathbf{x}^{(i)})$.

SMC is an extension of Importance Sampling to the time series of random variables $\mathbf{X}_T = \mathbf{x}_1 \cdots \mathbf{x}_T$.

SMC assumes the following state space model for $\mathbf{X}_T$ and observations $\mathbf{Y}_T = \mathbf{y}_1 \cdots \mathbf{y}_T$ according to $\mathbf{X}_T$:

$$\begin{cases} \mathbf{x}_t \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}) & \text{(transition equation)} \\ \mathbf{y}_t \sim p(\mathbf{y}_t|\mathbf{x}_t) & \text{(observation equation)} \end{cases} \tag{2.94}$$

Equations in (2.94) are called a transition equation and an observation equation, respectively.

Generic Particle Filter algorithm Assuming this general model, SMC estimates the function $f(\mathbf{x}_t)$ of the current state $\mathbf{x}_t$ of the system[4] from the observation $\mathbf{Y}_t = \mathbf{y}_1 \cdots \mathbf{y}_t$ up to time t, using N Monte Carlo samples called *particles* $\mathbf{x}_t^{(1)} \cdots \mathbf{x}_t^{(N)}$:

$$E[f(\mathbf{x}_t)|\mathbf{Y}_t] = \int f(\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_t)d\mathbf{x}_t \tag{2.95}$$

$$\simeq \sum_{i=1}^{N} f(\mathbf{x}_t^{(i)})w(\mathbf{x}_t^{(i)})\,. \tag{2.96}$$

[4]For simplification, we concentrate here only the *filtering* estimates of SMC that will be treated in this dissertation, leaving other kind of estimates that can also be treated by SMC.

Here, the following two properties are assumed:

- $\mathbf{x}_t^{(i)}$ is sampled from arbitrary proposal distribution $q(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_t)$ that satisfies some weak conditions:

$$\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_t). \tag{2.97}$$

- Weight of the i'th particle $\mathbf{x}^{(i)}$, $w(\mathbf{x}_t^{(i)})$, is initialized at time 1 by $1/N$, and updated by the following recursive formula (Doucet et al., 2001) :

$$w(\mathbf{x}_t^{(i)}) \propto w(\mathbf{x}_{t-1}^{(i)}) \, \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)}|\mathbf{X}_{t-1}, \mathbf{Y}_t)}, \tag{2.98}$$

  where $\propto$ means a normalization to sum to 1 over $i = 1 \ldots N$.

If we know the predictive distribution $p(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_t)$ exactly and use it as the proposal distribution $q(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_t)$, (2.98) reduces simply (Doucet, 1998) :

$$w(\mathbf{x}_t^{(i)}) \propto w(\mathbf{x}_{t-1}^{(i)}) \, p(\mathbf{y}_t|\mathbf{x}_{t-1}^{(i)}) \, . \tag{2.99}$$

As we will see in Chapter 4, this is the case in our study because once we know the previous hidden states by sampling, the predictive distribution of the next word is obtained in a closed form.

Chapter 3

Metric Learning Problem in Vector Space Models

In this chapter, we treat a metric learning problem that describes the heterogeneity of the latent space where linguistic expressions reside.

Natural language processing involves many kinds of linguistic expressions such as phrases, sentences, documents, and a collection of documents. Comparing these expressions based on semantic proximity is a fundamental task and has many application.

Generally, there are two basic approaches to compare two linguistic expressions: (a) structural and (b) non-structural. Structural approaches make use of syntactic parsing or similar method like dependency analysis to conduct a rigorous comparison of expressions, while nonstructural approaches use a vector representation and provide a rough but fast comparison that is needed for search or retrieval from vast amount of corpora.

Although structural approaches have become recently available in the kernel-based method (Collins and Duffy, 2001; Suzuki et al., 2003), here we concentrate on nonstructural comparison. This is not only because the nonstructural comparison constitutes an integral part of structural methods as its atomic comparison method in the leaves, but because it is frequently embedded in many application where structural parsings are not available or computationally too expensive to conduct for the whole data to compare.

For example, information retrieval usually deal with huge amount of data

and a "bag-of-words" vectorial description has been used, owing to the data size and also to a lack of scalable text segmentation algorithms. Meanwhile, text segmentation algorithms themselves, such as TextTiling (Hearst, 1994) and its recent successors using the inter-paragraph similarity matrix (Choi, 2000), all use the nonstructural cosine similarity as a measure of semantic proximity between the paragraphs.

However, the distance functions have been largely defined and used *ad hoc*, usually a tf.idf weighting scheme and a simple cosine similarity that is equivalent to an Euclidean dot product. This kind of distance functions have two severe problems that have been ignored when applied in natural language processing.

3.1. Problems with Euclidean distances

When we address nonstructural comparison, linguistic expressions are often modeled by a feature vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$, where each element u_i corresponds to the number of occurrences of i'th feature. When the features are simply words, this representation is called a "bag-of-words"; however, in general the features are not restricted to words and thus we use a general term "feature" throughout this chapter.

To measure a distance between the two vectors $\mathbf{x}, \mathbf{y}$, a dot product or Euclidean distance[1]

$$d(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T(\mathbf{x} - \mathbf{y}) \tag{3.1}$$

$$= \sum_{i=1}^{n}(x_i - y_i)^2 \tag{3.2}$$

(where ^{T} denotes a transposition) has been employed so far with a heuristic feature weighting such as tf.idf in the preprocessing stage.

There are two main problems as to this distance:

(i) The correlations between the features are ignored.

[1]When we normalize the length of the vectors $|\mathbf{x}| = |\mathbf{y}| = 1$ as commonly adopted, $(\mathbf{x} - \mathbf{y})^T(\mathbf{x}-\mathbf{y}) = |\mathbf{x}|^2 + |\mathbf{y}|^2 - 2\mathbf{x}\cdot\mathbf{y} \propto -\mathbf{x}\cdot\mathbf{y} = -\cos(\mathbf{x}, \mathbf{y})$; therefore, this includes a cosine similarity (Manning and Schütze, 1999).

(ii) Feature weightings are not optimal.

Problem (i) is especially important in natural language because linguistic features (e.g. words) usually have strong correlations between them, such as collocations, syntactic constructions, and so on, as well as a prior semantic correlations we may assume.

However, these correlations cannot be considered in equation (3.1). While it is possible to address this by a specific kernel function like polynomials (Müller et al., 2001), kernel-based approach is not available for many problems such as information retrieval or question answering that do not fit for the classification, the main outlet of kernel methods in current natural language processing.

Problem (ii) is more subtle but an inherent one: although tf.idf often works well in practice, it has several free options, especially in *tf* such as using logs or square roots. Nevertheless, we have no principle from which to choose because the Euclidean distance with tf.idf has no theoretical basis that gives any optimality as a distance function.

3.2. Related Work

The above problems of feature correlations and feature weightings can be summarized as a problem of defining an appropriate metric in the feature space based on the distribution of data. This problem has recently been highlighted in the field of machine learning research. (Xing et al., 2002) has an objective that is quite similar to that of this chapter and gives a metric matrix that resembles ours using sample pairs of "similar points" as training data. (Bach and Jordan, 2003) and (Schultz and Joachims, 2003) seek to answer the same problem with an additional scenario of spectral clustering and relative comparisons in Support Vector Machines, respectively. In this aspect, our work is a straight successor of (Xing et al., 2002) where its general usage in vector space is preserved. We offer a discussion on the similarity to our method and our advantages in section 3.5.

We note that the Fisher kernel of (Jaakkola and Haussler, 1999) has the same concept in that it gives an appropriate similarity of two data through the Fisher information matrix obtained from the empirical distribution of data. However,

the Fisher matrix is often approximated by a unit matrix because of its heavy computational demand.

In the field of information retrieval, (Jiang and Berry, 1998) proposes a Riemannian SVD (R-SVD) from the viewpoint of relevance feedback. This work is close in spirit to our work, but is not aimed at defining a permanent distance function and does not utilize cluster structures existent in the training data.

3.3. Defining an Optimal Metric

To solve the problems in section 3.1, we note the function that synonymous clusters play. There are many levels of (more or less) synonymous clusters in linguistic data: phrases, sentences, paragraphs, documents, and, in a web environment, the site that contains the document. These kinds of clusters often can be attributed to linguistic expressions because they nest in general so that each expression has a parent cluster.

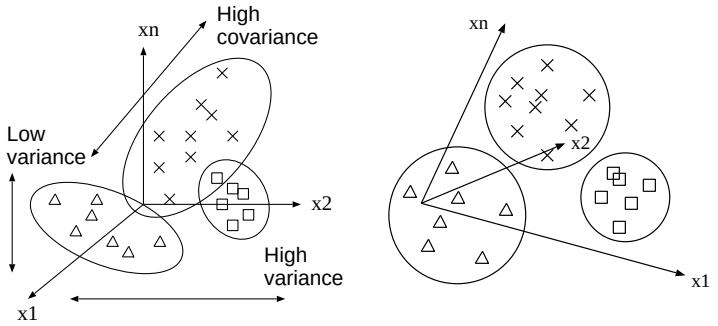
Since these clusters are synonymous, we can expect the vectors in each cluster to concentrate in the ideal feature space. Based on this property, we can introduce optimal weightings and correlations in a supervised fashion. We will describe this method below.

3.3.1 The Basic Idea

As stated above, vectors in the same cluster must have a small distance between each other in the ideal geometry. When we measure an L_2-distance between $\mathbf{x}$ and $\mathbf{y}$ by a Mahalanobis distance parameterized by M:

$$\begin{aligned} d_M(\mathbf{x},\mathbf{y})^2 &= (\mathbf{x}-\mathbf{y})^T M(\mathbf{x}-\mathbf{y}) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}(x_i-y_i)(x_j-y_j), \end{aligned} \tag{3.3}$$

where symmetric metric matrix $M = [m_{ij}]$ gives both corresponding feature weights and feature correlations. When we take $M = I$ (unit matrix), we recover the original Euclidean distance (3.1).



(a) Original space (b) Mapped space

Figure 3.1. Geometry of feature space.

Equation (3.3) can be rewritten as (3.4) because M is symmetric:

$$d_M(\mathbf{x}, \mathbf{y})^2 = (M^{1/2}(\mathbf{x} - \mathbf{y}))^T (M^{1/2}(\mathbf{x} - \mathbf{y})). \tag{3.4}$$

Therefore, this distance amounts to a Euclidean distance in the $M^{1/2}$-mapped space (Xing et al., 2002).

Note that this distance is global, and *different* from the ordinary Mahalanobis distance in pattern recognition (for example, (Duda et al., 2000)) that is defined for each cluster one by one, using a cluster-specific covariance matrix. That type of distance cannot be generalized to new kinds of data; therefore, it has been used for local classifications. What we want is a global distance metric that is generally useful, not a measure for classification to predefined clusters. In this respect, (Xing et al., 2002) shares the same objective as ours.

Therefore, we require an optimization over all the clusters in the training data. Generally, data in the clusters are distributed as in Figure 3.1(a), comprising hyperellipsoidal forms that have high (co-)variances for some dimensions and low (co-)variances for other dimensions. Further, the cluster is not usually aligned to the axes of coordinates. When we find a global metric matrix M that minimizes the cluster distortions, namely, one that reduces high variances and expands low variances for the data to make a spherical form as good as possible in the $M^{1/2}$-mapped space (Figure 3.1(b)), we can expect it to capture necessary and unnecessary variations and correlations on features, combining information from many clusters to produce a more reliable metric that is not only locally optimal. We will find this optimal M below.

3.3.2 Global optimization over clusters

Suppose that each data (for example, sentences or documents) is expressed as a vector $\mathbf{x} \in \mathbb{R}^n$, and the whole corpus can be divided into N clusters, $\mathbf{X}_1 \dots \mathbf{X}_N$. That is, each vector has a dimension n, and the number of clusters is N. For each cluster $\mathbf{X}_i$, cluster centroid c_i is computed as $\mathbf{c}_i = 1/|\mathbf{X}_i| \sum_{\mathbf{x} \in \mathbf{X}_i} \mathbf{x}$, where $|\mathbf{X}|$ denotes the number of data in $\mathbf{X}$. When necessary, each element in $\mathbf{x}_j$ or $\mathbf{c}_i$ is referenced as x_{jk} or c_{ik} $(k = 1 \dots n)$.

The basic idea above is formulated as follows. We seek the metric matrix M that minimizes the metric distance between each data $\mathbf{x}_j$ and the cluster centroid $\mathbf{c}_i$, $d_M(\mathbf{x}_j, \mathbf{c}_i)$ for all clusters $\mathbf{X}_1 \dots \mathbf{X}_N$. Mathematically, this is formulated as a quadratic minimization problem

$$\begin{aligned} M &= \arg\min_M \sum_{i=1}^{N} \sum_{\mathbf{x}_j \in \mathbf{X}_i} d_M(\mathbf{x}_j, \mathbf{c}_i)^2 \\ &= \arg\min_M \sum_{i=1}^{N} \sum_{\mathbf{x}_j \in \mathbf{X}_i} (\mathbf{x}_j - \mathbf{c}_i)^T M (\mathbf{x}_j - \mathbf{c}_i) \end{aligned} \tag{3.5}$$

under a scale constraint ($|\cdot|$ means determinant)

$$|M| = 1. \tag{3.6}$$

This scale constraint is necessary for excluding a degenerate solution $M = O$. 1 is an arbitrary constant: when we replace 1 by c, $c^2 M$ becomes a new solution. This minimization problem is an extension to the method of *MindReader* (Ishikawa et al., 1998) to multiple clusters and has a unique solution below.

Theorem The matrix that solves the minimization problem (3.5,3.6) is

$$M = |A|^{1/n} A^{-1}, \tag{3.7}$$

where $A = [a_{kl}]$ is defined by

$$a_{kl} = \sum_{i=1}^{N} \sum_{\mathbf{x}_j \in \mathbf{X}_i} (x_{jl} - c_{il})(x_{jk} - c_{ik}) \ . \tag{3.8}$$

Proof: We want to derive M that satisfies the condition

$$\min_M \sum_{i=1}^{n} \sum_{\mathbf{x}_j \in \mathbf{X}_i} (\mathbf{x}_j - \mathbf{c}_i)^T M (\mathbf{x}_j - \mathbf{c}_i) \ , \tag{3.9}$$

under the constraint

$$|M| = 1. \tag{3.10}$$

Expanding (3.9), we get

$$\sum_i \sum_{\mathbf{x}_j} \left[\sum_{k=1}^{n} \sum_{l=1}^{n} (x_{jk} - c_{ik}) m_{kl} (x_{jl} - c_{il}) \right] \ , \tag{3.11}$$

and from (3.10), for all k

$$\sum_{l=1}^{n} (-1)^{k+l} m_{kl} |M_{kl}| = 1 \ .$$

Therefore

$$\sum_{k=1}^{n} \sum_{l=1}^{n} (-1)^{k+l} m_{kl} |M_{kl}| = n, \tag{3.12}$$

where M_{kl} denotes an adjugate matrix of m_{kl}.

Therefore, we come to minimize (3.11) under the constraint (3.12).

By introducing the Lagrange multiplier λ, we define

$$L = \sum_{i=1}^{N} \sum_{\mathbf{x}_j} \left[\sum_k \sum_l (x_{jk} - c_{ik}) m_{kl} (x_{jl} - c_{il}) \right] - \lambda \left[\sum_k \sum_l (-1)^{k+l} m_{kl} |M_{kl}| - n \right].$$

Differentiating by m_{kl} and setting to zero, we obtain

$$\begin{aligned} \frac{\partial L}{\partial m_{kl}} =& \sum_i \sum_{\mathbf{x}_j} (x_{jk} - c_{ik})(x_{jl} - c_{il}) \\ & - \lambda (-1)^{k+l} |M_{kl}| = 0 \\ \Leftrightarrow \ |M_{kl}| =& \frac{\sum_i \sum_{\mathbf{x}_j} (x_{jk} - c_{ik})(x_{jl} - c_{il})}{\lambda (-1)^{k+l}}. \end{aligned} \tag{3.13}$$

Let us define $M^{-1} = [m_{kl}^{-1}]$. Then,

$$\begin{aligned} m_{kl}^{-1} &= \frac{(-1)^{k+l}|M_{kl}|}{|M|} \\ &= (-1)^{k+l}|M_{kl}| \quad (\because \ (3.10)) \\ &= \frac{\sum_i \sum_{\mathbf{x}_j} (x_{jk} - c_{ik})(x_{jl} - c_{il})}{\lambda} \\ &\quad (\because \ (3.13)) \end{aligned} \tag{3.14}$$

Therefore, when we define

$$A = [a_{kl}] \tag{3.15}$$

as

$$a_{kl} = \sum_{i=1}^{N} \sum_{\mathbf{x}_j \in \mathbf{X}_i} (x_{jl} - c_{il})(x_{jk} - c_{ik}) \ , \tag{3.16}$$

from (3.14),

$$\begin{aligned} & A = \lambda M^{-1} \\ \Leftrightarrow \ & |A| = \lambda^n |M^{-1}| = \lambda^n \\ \Leftrightarrow \ & \lambda = |A|^{1/n} \ , \end{aligned}$$

where A is defined by (3.15), (3.16). $\square$

When A is singular, we can use as A^{-1} a Moore-Penrose matrix pseudoinverse A^+. Generally, A consists of linguistic features and therefore is very sparse and often singular. Hence A^+ is nearly always necessary for the above computation. For details, see Appendix C.

3.3.3 Generalization

While we assumed through the above construction that each cluster is equally important, this is not the case in general. For example, clusters with a small number of data may be considered weak, and in the hierarchical clustering situation, a "grandmother" cluster may be weaker. If we have confidences $\xi_1 \ldots \xi_N$ for the strength of clustering for each cluster $\mathbf{X}_1 \ldots \mathbf{X}_N$, this information can be incorporated into (3.5) by a set of normalized cluster weights ξ_i^*:

$$M = \arg\min_M \sum_{i=1}^{N} \xi_i^* \sum_{\mathbf{x}_j \in \mathbf{X}_i} (\mathbf{x}_j - \mathbf{c}_i)^T M (\mathbf{x}_j - \mathbf{c}_i),$$

where $\xi_i^* = \xi_i / \sum_{j=1}^{N} \xi_j$, and we obtain a respectively weighted solution in (3.8). Further, we note that when $N = 1$, this metric recovers the ordinary Mahalanobis distance in pattern recognition. However, because the number of data in each cluster was approximately equal we used equal weights for the experiments below.

3.4. Experiments

We evaluated the proposed metric distance on three tasks: synonymous sentence retrieval, document retrieval, and the K-means clustering of general vectorial data. Since the proposed method utilizes only the general property of vector space, it should be effective on general vectorial datasets last mentioned.

The procedure of the experiments is as follows. First, the metric matrix is computed from the cluster structure of training data. Second, in the retrieval task, for each datum in the test set the distances to the other data are computed and sorted in an ascending order. Within this sorted list of distances, the correct answers are the data from the same original cluster as the datum in consideration belongs to. These correct answers in the sorted list yield a precision-recall curve and the R-precision (Baeza-Yates and Ribeiro-Neto, 1999). When we set R to the number of data in the original cluster minus 1, R-precision equals 1 when the first R data are totally from the original cluster and equals 0 when they do not include data from the original cluster at all; thus R-precision means a precision of original cluster recovery. The distribution below the first R data is expressed as the precision-recall curve and its point summary, 11-point average precision.

We conducted this procedure for each datum in the test set: hence its computational complexity is quadratic to the size of the test set. Precisions in the clustering task is described in section 3.4.3.

For all the distance computation above, we compared the case using the metric distance with the baseline case of Euclidean distance.

3.4.1 Synonymous sentence retrieval

Searching synonymous sentences from the corpus or the set of example sentences constitutes a fundamental technology underlying some natural language process-

ing tasks such as Example-based machine translation or candidate retrieval in Question Answering.

3.4.1.1 Sentence cluster corpus

We used a paraphrasing corpus of travel conversations (Sugaya et al., 2002) for sentence retrieval. This corpus consists of 33,723,164 Japanese translations, each of which corresponds to one of the original English sentences. By way of this correspondence, Japanese sentences are divided into 10,610 clusters. Therefore, each cluster consists of Japanese sentences that are possible translations from the same English seed sentence that the cluster corresponds to. From this corpus, we constructed 10 sets of data. Each set contains random selection of 200 training clusters and 50 test clusters, where each cluster consists of maximum 100 sentences[2]. Experiments were conducted on these 10 datasets to yield an average statistics, using each level of dimensionality reductions described below.

3.4.1.2 Features and dimensionality reduction

As the features of a sentence, we used unigrams of all words and bigrams of functional words[3] that consist of the sentence. This is because the concatenations of functional words are considered important for comparison in the conversational domain.

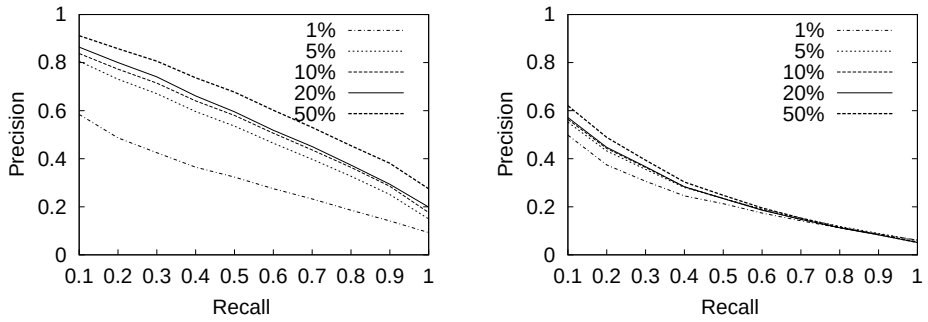
While the lexicon is limited for travel conversations, the number of the features exceeds several thousands or more; that may prohibit the computation of the metric matrix in its full form and produces unreliable metric matrix due to its excessive sparseness. Therefore, we first compressed the features using singular valude decomposition (SVD), the same method of Latent Semantic Indexing (Deerwester et al., 1990).

3.4.1.3 Sentence retrieval results

Qualitative results Figure 3.7 shows a sample result of synonymous sentence retrieval. Sentences with (*) mark at their end are the correct answers, that is, the

[2]When the number of data in the cluster exceeds this limit, 100 sentences are randomly sampled. All sampling are made without replacement.

[3]Using the provided part-of-speech information, here functional words are defined as the words other than content words, that is, nouns, proper nouns, numerals, and main verbs.



(a) Metric distance + idf (b) Euclidean + idf

Figure 3.2. Precision-recall of sentence retrieval.

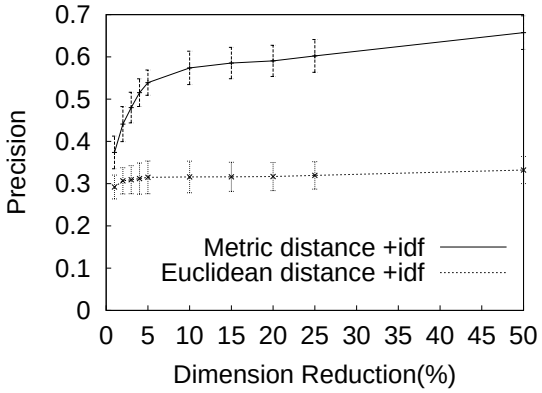


Figure 3.3. 11-point average precision.

sentences that belong to the same cluster as the query. We see that the results using the metric distance contain less noises than using a standard Euclidean baseline with tf.idf weighting, achieving a high-precision retrieval in vector space. Although in case of high rate of dimensionality reduction in Figure 3.8 shows a performance degradation owing to the lower dimensional projection, the effects of the metric distance are still apparent despite these bad circumstances.

Quantitative results Figure 3.2 shows the averaged precision-recall curves of the retrieval and Figure 3.3 shows the 11-point average precisions for each rate of dimensionality reduction. Clearly, the proposed method achieves higher precision than the standard method and does not degrade much with the feature compression (dimensionality reduction) unless we reduce dimensions too much, i.e. less than 5%.

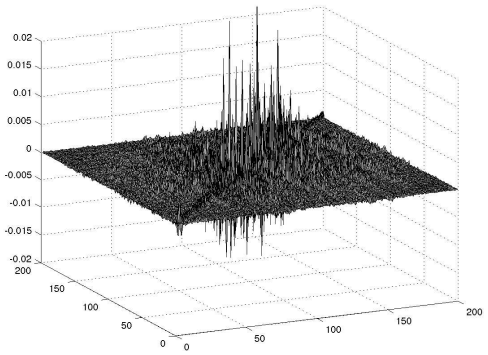


Figure 3.4. A metric matrix obtained from synonymous clusters.

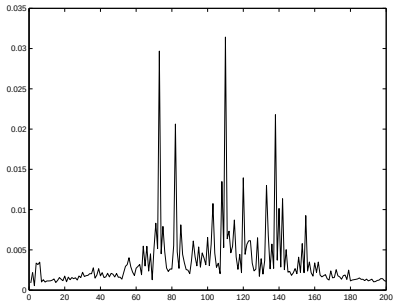


Figure 3.5. Diagonal elements of the metric matrix.

3.4.1.4 Metric Matrix

Figure 3.4 shows a sample metric matrix obtained from the synonymous sentence clusters. Features were first preprocessed by tf.idf and compressed to 200 dimensions by SVD. Here, standard Euclidean (cosine) distance corresponds to a unit matrix where only the diagonal elements from upper right to lower left equal to 1. Apparently, non-diagonal elements take many different values, representing positive or negative correlations between the compressed features. Sum of the absolute values of the diagonal elements occupied only 3.5% of the total sum over the whole matrix elements.

Moreover, even the weight of diagonal elements are not uniform as shown in Figure 3.5. This plot show that the tf.idf weights are further modified from the synonymous cluster information in the training data.

3.4.2 Document retrieval

As a method of tackling clusters of texts, the text classification task has recently made great advances with a Naïve Bayes or SVM classifiers (for example, (Joachims, 1998)). However, they all aim at classifying texts into a few predefined clusters and cannot deal with a document that fits neither of these clusters. For example, when we regard a website as a cluster of documents, possible clusters are numerous and constantly increasing, which precludes classificatory approaches. For these circumstances, document clustering or retrieval will benefit from a global distance metric that exploits the multitude of cluster structures themselves.

3.4.2.1 Newsgroup text dataset

For this purpose, we used the 20-Newsgroup dataset (Lang, 1995). This is a standard text classification dataset that has a relatively large number of classes, 20. Among the 20 newsgroups, we selected 16 clusters of training data and 4 clusters of test data to perform a five-fold cross validation. The maximum number of documents per cluster is 100; when it exceeds this limit, we made a random sampling of 100 documents from the cluster in consideration.

Since the proposed metric is computed from the distribution of vectors in high-dimensional feature space, it will not be meaningful when the norm of the vector (largely proportional to document length) differs much from document to document.[4] Therefore, we conducted subsampling/oversampling for the training document to be the median length (130 words). We used unigrams as features and preprocessed them with tf.idf as a baseline method.

3.4.2.2 Results

Table 3.1 shows R-precision and 11-point average precision for the document retrieval. Since the test data contains 4 clusters, the baseline of precisions is 0.25. We can see from both results that the metric distance attains a better

[4]Normalizing documents to unit length effectively maps them to a high-dimensional hypersphere; this proved to produce an unsatisfactory result. Defining metrics that work on a hypersphere like spherical K-means (Dhillon and Modha, 2001) requires further research.

Dim.	R-precision		11-pt Avr. Prec.	
Red.	Metric	Euclid	Metric	Euclid
0.5%	**0.421**	0.399	**0.476**	0.455
1%	**0.388**	0.368	**0.450**	0.430
2%	**0.359**	0.343	**0.425**	0.409
3%	**0.344**	0.330	**0.411**	0.399
4%	**0.335**	0.323	**0.402**	0.392
5%	**0.329**	0.318	**0.397**	0.388
10%	**0.316**	0.307	**0.379**	0.376
20%	**0.343**	0.297	**0.397**	0.365

Table 3.1. Newsgroup text retrieval results.

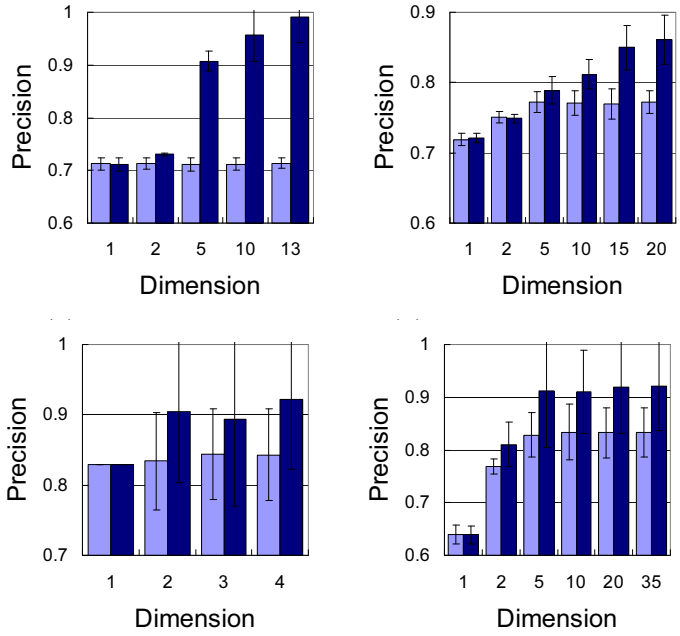
retrieval over tf.idf and dot product. However, precision refinements are certain (average $p = 0.0243$) but subtle.

This can be considered the effect of the dimensionality reduction performed. We first decompose data matrix X by SVD: $X = USV^{-1}$ and build a k-dimensional compressed representation $X_k = V_k X$; where V_k denotes a k-largest submatrix of V. From the equation (3.4), this means using an Euclidean distance between the rows of $M^{1/2}X_k = M^{1/2}V_k X$. Therefore, V_k may subsume the effect of M in a preprocessing stage. Close inspection of table 3.1 shows this effect as a tradeoff between the metric effect and dimensionality reduction. To make the most of metric distance, we should consider metric induction and dimensionality reduction simultaneously or reconsider the problem in kernel Hilbert space.

3.4.3 K-means clustering and general vectorial data

Metric distance can also be used for clustering or general vectorial data. Figure 3.6 shows the K-means clustering result of applying the metric distance to some of the UCI Machine Learning datasets (Blake and Merz, 1998). K-means clustering was conducted 100 times with random starts, where K equals the known number of classes in the data.[5] Clustering precision was measured as an average probability

[5]Because of the small size of the dataset, we did not use cross-validation as in the other experiments.



(c) "iris" dataset (d) "soybean" dataset

Figure 3.6. K-means clustering of UCI Machine Learning dataset results. The horizontal axis shows compressed dimensions (rightmost is original). The right bar shows clustering precision using Metric distance and the left bar shows that using Euclidean distance.

that a randomly picked pair of data will conform to the case of true clustering (Xing et al., 2002).

We also conducted the same clustering for documents of the 20-Newsgroup dataset to obtain a small increase in precision like the document retrieval experiment in section 3.4.2.

3.5. Discussion

In this chapter, we proposed an optimal distance metric based on the idea of minimum cluster distortion in training data. Although vector distances have frequently been used in natural language processing, this is a rather neglected but recently highlighted problem. Unlike recently proposed methods with spectral methods or SVMs, our method assumes no such additional scenarios and can be

considered as a straight successor to (Xing et al., 2002).

Their work has the same perspective as ours and they calculate a metric matrix A that is similar to ours based on a set $\mathcal{S}$ of vector pairs $(\mathbf{x}_i, \mathbf{x}_j)$ that can be regarded as similar. They report that the effectiveness of A increases as the number of the training pairs $\mathcal{S}$ increases; this requires $O(n^2)$ sample points from n training data and must be optimized by a computationally expensive Newton-Raphson iteration. On the other hand, our method uses only linear algebra and can induce an ideal metric using all the training data at once. We believe this metric can be useful for many vector-based language processing methods that have been using cosine similarity.

There remains some future directions for research. First, as we stated in section 3.3.3, the effect of a cluster weighted generalized metric must be investigated and optimal weighting must be induced. Second, as noted in section 3.4.2.1, the dimensionality reduction required for linguistic data may constrain the performance of the metric distance. To alleviate this problem, simultaneous dimensionality reduction and metric induction may be necessary, or the same idea in a kernel-based approach is worth considering. The latter obviates the problem of dimensionality, while it restricts the usage to situations where the kernel-based approaches are available.

3.6. Summary

This chapter proposed a global metric distance that is useful for many natural language processing tasks such as clustering or retrieval in place of the Euclidean distance that has been used. This distance is optimal in the sense of quadratic minimization over all the clusters in the training data. Experiments on sentence retrieval, document retrieval and K-means clustering all showed improvements over Euclidean distance, with a significant refinement with tight training clusters in sentence retrieval.

Query: “合計でいくらですか”
('How much is the total?')

Metric distance:

distance	synonymous sentence
0.2712	合計でいくらでしょうか *
0.3444	内金はいくらですか
0.3444	入場料はいくらですか
0.369	手付金はいくらですか
0.4377	合計でいくらいたしますか *
0.4479	合計でいくらいたしますでしょうか *
0.4505	全部でいくらですか *
0.4558	合計でいくらになりますか *
0.4602	合計でいくらになりますでしょうか *
0.4682	合計でいくらになるでしょうか *
0.4729	合計でいくらしますか *
0.4851	合計でいくらしますでしょうか *

Euclidean distance:

distance	synonymous sentence
0.1732	全部でいくらですか *
1.781	合計でおいくらですか *
1.902	紫外線防止ですか
1.966	内金はいくらですか
1.966	入場料はいくらですか
1.974	手付金はいくらですか
1.983	全部でおいくらですか *
2.283	どんな兆候ですか
2.505	どんな症状ですか
2.65	お一人ですか
2.729	放送で呼び出してください
2.749	紫外線防止ですね

(* denotes the right answers.)

Figure 3.7. Example of synonymous sentence retrieval.

Query: “デザートに果物をくれないでしょうか”
('I'd like some fruit for dessert.')

Metric distance:

dist.	synonymous sentence
0.3531	請求書をすぐにくれないでしょうか
0.3709	デザートとして果物をくれますか *
0.596	請求書をすぐにくれませんか
0.6104	伝票をすぐにくれますか
0.621	伝票をすぐにくれますでしょうか
0.6255	お勘定書をすぐにくれますか
0.6295	伝票をすぐにくれませんでしょうか
0.6343	お勘定書をすぐにくれませんか
0.6685	伝票をすぐにくれないですか
0.7966	デザートには果物をくれないですか *

Euclidean distance:

dist.	synonymous sentence
1.036	請求書をすぐにくれないでしょうか
1.421	朝ごはんを部屋に運んでもらえないでしょうか
1.491	ウィスキーを二人分くれないでしょうか
1.499	ウィスキーを二つくれないでしょうか
1.535	薬をくれないでしょうか
1.622	朝食を部屋に運んでもらえないでしょうか
1.622	朝食を部屋に運んでもらえないでしょうか
:	:
2.787	デザートとして何か果物をくれないでしょうか *
2.854	この円をポンドに換算くださらないでしょうか

Figure 3.8. High level of dimensionarity reduction of features.

Chapter 4

Statistical Language Modeling of Contexts

4.1. Introduction

Contextual effect constitutes an essential part of linguistic phenomena, and its estimation is also crucial for natural language processing. We infer context in which we are involved to create an adaptive linguistic behavior based on the appropriate model selection on that information.

Considering actual applications such as speech recognition, machine translation or robotics, they may benefit from the model to detect context shifts and produce the most appropriate adaptive outputs for the current context.

In natural language processing terms, this issue can be considered a problem of the adaptation of a long-distance language model beyond n-grams. While "syntactic" probabilities such as trigrams are comparatively stable and unaffected by context, "semantic" probabilities like unigrams are heavily affected by context, thus dynamic adaptation to context is an important problem to be solved.

Long-distance language modeling began with a classic approach such as triggers/caches (Jelinek, 1998), followed by a method using Latent Semantic Indexing (LSI) that can capture word cooccurrences that are not necessarily present in a training corpus (Bellegarda, 1998). This method was extended by a strict probabilistic approach using Probabilistic Latent Semantic Indexing (PLSI), which assumes hidden topic variables present in the current context to predict the next

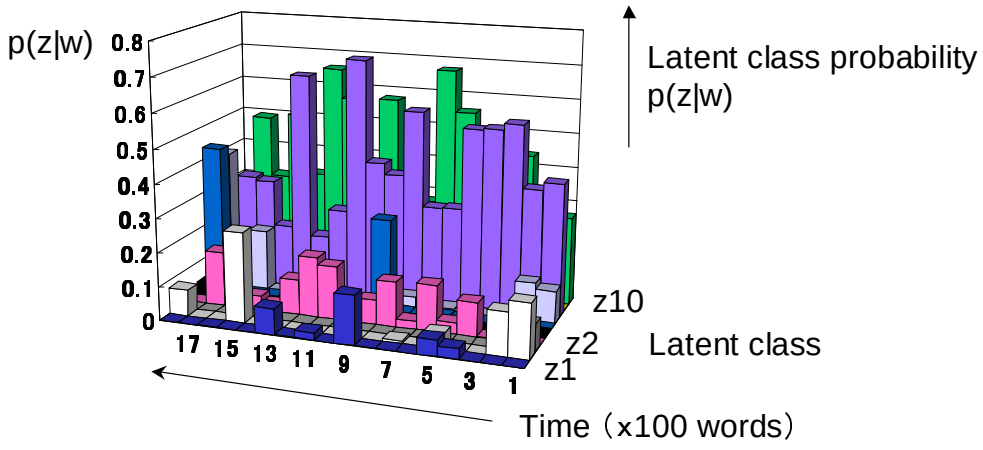


Figure 4.1. Example PLSI decomposition of 1,700 words text.

word (Gildea and Hofmann, 1999). Recently, this approach was further sophisticated using Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to produce a lower perplexity thanks to Bayesian methods (Mishina and Yamamoto, 2004).

However, these models implicitly assumed stable context and do not consider *dynamic change of context* or even topic shifts. Since these models are application of bag-of-words text modelings, the history is simply a bag of words from the beginning of the document or a pre-determined threshold like 1,000 words (Kurohashi and Ori, 2000), totally ignoring the chronological order of the history. In other words, so far long-distance language models are approaches that regard a text as a stationary information source, no matter how long or heterogeneous it is, to gradually improve the estimation of the static parameter as the data become available as "context."

Apparently, this is not a desirable approach but only an approximation. Figure 4.1 is a sample PLSI decomposition of a Mainichi newspaper article of about 1,700 words, segmented by each 100 words from the beginning of the text. Although we recognize that the main topic of this article largely corresponds to 8th topic, the proportion varies along time, giving some probability mass to the other latent subtopic and representing the semantic heterogeneity of the text. Since the actual number of latent topics is far larger than 10 here for explanation, this kind of semantic random walk may be more apparent than this simple example in the actual texts.

In fact, TextTiling (Hearst, 1994) is an algorithm that divides a text into subtopic segments by using such a text heterogeneity. Even in the language modeling research, Bellegarda (1998) we mentioned above noticed a best "forgetting coefficient" that produced the lowest perplexity for the corpus he used.

Contrary to previous models, in this chapter we view context shift as a latent stochastic process and give an explicit probabilistic model that describes hidden topic shifts in a text stream to estimate its state and parameters sequentially.

In words of signal processing or control theory, this approach amounts to a *filtering* that estimates current state of a system along time, rather than a simple averaging of past observations like the previous approaches. Based on the filtering approach, we propose a novel long-distance language model that can estimate subtopic changes and their rate online and automatically uses an appropriate length of context.

Essentially, this model is a nonlinear HMM that cannot be decoded by a traditional Baum-Welch algorithm or Kalman Filters. For this purpose, we used a Particle Filter, a sequential Monte Carlo method described in Chapter 2 that has been mainly used in signal processing and robotics research, to estimate the nonlinear dynamics of context sequentially.

Our contribution in this chapter is threefold. First, we view a long-distance language model a filtering problem to solve it by introducing an explicit generative model that describes hidden topic shifts. Second, for this purpose we extended the multinomial Particle Filter of DNA sequence recently proposed in statistics to natural language, by combining it with LDA and DM described in Chapter 2. Third, in this model we propose an online update of hyperparameter that was assumed to be known and fixed in the original modeling.

4.2. Previous Work

Although "context" is a polysemous word that may mean some local syntactic environment or global semantic environment as usually conceived, here we concentrate on the latter as an optimal prediction problem utilizing the global semantic environment. Hereafter, we use context modeling as a synonym for that kind of language modeling.

4.2.1 Ad Hoc Approaches to Context Modeling

Since n-gram language model (usually, trigrams or bigrams) only answers for local regularities of natural language, long-distance modeling beyond n-grams has attracted much research interests.

Long-distance language modeling began with a classic approach such as triggers or caches (Jelinek, 1998) that boosts probabilities of the words themselves or related words seen in the history considered. In the trigger approach, for each word a set of "related" words are assigned with the relatedness and the length of effect computed from corpora by typically a maximum entropy approach (Beeferman et al., 1997b).

However, there are two problems about this frequently used approach:

1. There are no guarantee that these "related" words are exhaustive. In fact, complete enumeration of association pairs requires $O(L^2)$ spaces with the lexicon of L words that needs some cut-off. However, such a cut-off may cause performance degradation especially for low-frequency words that are important for prediction but are often discarded by the cut-off.

2. Since the association pairs are used independently in the trigger models, agglomerative effects are not considered. Many words have polysemy that can be usually resolved through an interaction between the other words in the history. For example, if 'bank' and 'interest' are both contained in the history, only the probabilities of financial words should be increased. However, in the trigger approach in this case, river-related probabilities are also increased misguidedly.[1]

Therefore, we need a model that utilizes not only the surface appearances of words but also hidden semantic community inferenced from the consisted words. In statistical terms, we need latent structure introduced into the probabilistic modeling, explicating in the next subsection.

As an approximation to the latent structure mentioned above, Bellegarda (1998) first introduced a LSI language model using Latent Semantic Indexing (Deer-

[1]Of course, this deficiency can be avoided by introducing pairs of features like '(bank,interest)' into the model. However, those augmented model inevitably suffers from data sparseness and exponentially increase of the number of features, becoming virtually useless.

wester et al., 1990). In LSI language model, the "centroid" of the previous words represented as a K-dimensional vector in the reduced vector space, called Latent Semantic Space, is computed to produce a unigram probability as the value of a cosine distance between a word and the centroid, normalized to sum to 1.

Although this approach is important that introduces a hidden space other than the surface appearances of words, it is not a strictly statistical method and therefore requires many ad-hoc parameters and preprocessings.

Florian (1999) proposes another approach using hierarchical document clustering conducted in advance. This approach first make a hiearchical clustering of training documents and allocates a history regarded as a pseudo-document on the one of the derived clusters to predict the next word.

This model is interesting in using hierarchical clustering of documents as opposed to the flat clustering usually adopted. However, it cannot escape from the limitation we will relax in this study in the following sections.

4.2.2 Latent Variable Context Modeling

Contrary to the ad hoc methods mentioned above, strict probabilistic approach based on mixture models have been proposed lately. Because the proposed method is an orthogonal extension to these models, here we explain them somewhat closely.

4.2.2.1 PLSI Language Model

Gildea and Hofmann (1999) proposed a probabilistically sound context modeling using the EM algorithm as opposed to previous modelings.

This method is a simple application of PLSI (Hofmann, 1999) described in Chapter 2 using the EM algorithm, namely, maximum likelihood estimate.

When a history

$$\mathbf{h} = w_1 w_2 \dots w_h \tag{4.1}$$

is observed, PLSI language model first conducts the following EM algorithm to compute a maximum likelihood estimate of the low-dimensional posterior topic distribution $p(t|\mathbf{h})$, while PLSI parameters $p(w|z)$ and $p(z)$ fixed:

E step:

$$p(z|w, \mathbf{h}) \propto p(z)p(w|z)p(\mathbf{h}|z) \tag{4.2}$$

M step:

$$p(\mathbf{h}|z) \propto \sum_{w} \sum_{z} n(w)p(z|w, \mathbf{h}) \tag{4.3}$$

$$p(z) \propto \sum_{d} \sum_{w} n(w)p(z|w, \mathbf{h}) \tag{4.4}$$

Figure 4.2. EM algorithm for PLSI Language Model.

Moreover, they also proposed an online EM algorithm based on (Neal and Hinton, 1998). In this modeling, history $\mathbf{h}$ is regarded as a pseudo document of bag-of-words. Threfore, sequential information in $\mathbf{h}$ has been discarded here.

Gildea and Hofmann (1999) reports experimental results in TDT-1 and WSJ corpus with significant perplexity reduction using unigrams, and trigrams combined with rescaled unigrams.

4.2.2.2 LDA Language Model

Since PLSI language model builds on a maximum likelihood estimate of $\boldsymbol{\lambda} = \{p(t|\mathbf{h})\}_{t=1}^{M}$, it sometimes suffers from severe overfitting to provided history to produce a poor prediction based on the overfitted topic estimate.

To alleviate this problem, Mishina and Yamamoto (2002) proposes to use a Bayesian estimate instead of maximum likelihood estimate; namely, using Latent Dirichlet Allocation (LDA) (Blei et al., 2001) described in Chapter 2 in place of PLSI.

Specifically, when we assume that $\boldsymbol{\lambda} = \{p(t|\mathbf{h})\}_{t=1}^{M}$ have a prior Dirichlet distribution

$$\boldsymbol{\lambda} \sim \mathrm{Dir}(\boldsymbol{\alpha}) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} \lambda_k^{\alpha_k - 1}, \tag{4.5}$$

the posterior distribution of $\boldsymbol{\lambda}$ given $\mathbf{h}$, $p(\boldsymbol{\lambda}|\mathbf{h})$, is given by the following Variational Bayes EM (VB-EM) algorithm.

Here, $\boldsymbol{\alpha}$ and $p(w|t)$ are the LDA parameters computed in advance from a collection of documents as a prior knowledge.

VB-E step:

$$q(z_i^t = 1|\mathbf{h}) \propto \ p(w_i|t) \exp(\Psi(\alpha_t + n_t)) \quad (4.6)$$

VB-M step:

$$q(\boldsymbol{\lambda}|\mathbf{h}) \propto \prod_{t=1}^{K} \lambda_t^{\alpha_t + n_t - 1} \quad (4.7)$$

$$\text{where } n_t = \sum_{i=1}^{h} q(z_i^t = 1|\mathbf{h}).$$

Figure 4.3. VB-EM algorithm for LDA Language Model.

$q(\boldsymbol{\lambda}|\mathbf{h})$ is a variational approximation of $p(\boldsymbol{\lambda}|\mathbf{h})$: using $q(\boldsymbol{\lambda}|\mathbf{h})$ instead of $p(\boldsymbol{\lambda}|\mathbf{h})$, the probability of the next word y given $\mathbf{h}$ is

$$p(y|\mathbf{h}) = \int \sum_z p(y, z, \boldsymbol{\theta}|\mathbf{h}) d\boldsymbol{\theta} \quad (4.8)$$

$$= \int \sum_z p(y|z) p(z, \boldsymbol{\theta}|\mathbf{h}) d\boldsymbol{\theta} \quad \text{(Model assumption)} \quad (4.9)$$

$$\simeq \sum_z p(y|z) \int \theta_z q(\boldsymbol{\theta}|\mathbf{h}) d\boldsymbol{\theta} \quad \text{(Variational approximation)} \quad (4.10)$$

$$= \sum_z p(y|z) \langle \theta_z \rangle_{q(\boldsymbol{\theta}|\mathbf{h})} \,. \quad (4.11)$$

That is, prediction in LDA language model is a mixture of unigrams, where the mixing weights are the expectation of the variational posterior Dirichlet distribution of the mixture given the history.

Mishina and Yamamoto (2002) reports that this language model produces lower perplexity as a context modeling by avoiding the overfitting problem seen in the PLSI language model. Since in context modeling the available data are usually small, this Bayesian estimate well suits for context modeling. In fact, this chapter will extend the LDA and DM language model described below.

4.2.2.3 DM Language Model

While LDA language model provides robust estimation using a Bayesian inference of topic decomposition of the history, it has a problem that it cannot model a whole word simplex where the final predictive distribution reside.

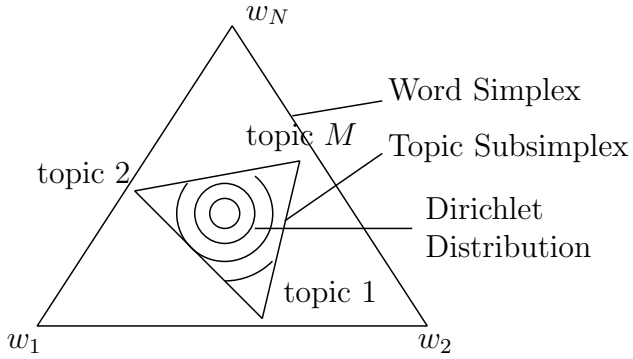


Figure 4.4. Word simplex and Topic subsimplex.

As shown in Figure 4.4, since the topic simplex is a lower-dimensional subsimplex of the word simplex, many regions outside the topic simplex cannot be used in LDA language modeling.

Contrary to LDA, Dirichlet Mixture (DM) (Yamamoto et al., 2003; Sadamitsu et al., 2004) is another Bayesian text model that works directly on the word simplex. DM has two parameters, $\boldsymbol{\lambda}$ and $\boldsymbol{\alpha}_1^M = \boldsymbol{\alpha}_1 \cdots \boldsymbol{\alpha}_M$; model details and the parameter estimation are described in Chapter 2.

Under DM, the predictive probability $p(y|\boldsymbol{h})$ is

$$p(y|\boldsymbol{h}) = \sum_{m=1}^{M} p(y|m)p(m|\boldsymbol{h}) \tag{4.12}$$

$$= \sum_{m=1}^{M} \left(\int p(y|\boldsymbol{p})p(\boldsymbol{p}|\boldsymbol{\alpha}_m)d\boldsymbol{p} \right) \cdot p(m|\boldsymbol{h})$$

$$= \sum_{m=1}^{M} C_m \, \frac{\alpha_{my}+n_y}{\sum_y(\alpha_{my}+n_y)}\,, \tag{4.13}$$

where

$$C_m = p(m|\mathbf{h}, \boldsymbol{\alpha}, \boldsymbol{\lambda}) \tag{4.14}$$

$$= \left(\lambda_m \, \frac{\Gamma(\sum_y \alpha_{my})}{\prod_y \Gamma(\alpha_{my})} \prod_y \frac{\Gamma(\alpha_{my}+n_y)}{\Gamma(\alpha_{my})} \right) \Big/ \sum_m \left(\lambda_m \, \frac{\Gamma(\sum_y \alpha_{my})}{\prod_y \Gamma(\alpha_{my})} \prod_y \frac{\Gamma(\alpha_{my}+n_y)}{\Gamma(\alpha_{my})} \right). \tag{4.15}$$

and n_y is the number of occurrences of y in $\mathbf{h}$.

This prediction can also be considered an extension to Dirichlet smoothing

(2.6) reproduced here for convenience:

$$p(y|\mathbf{h}) = \frac{\alpha_y + n_y}{\sum_y (\alpha_y + n_y)} \tag{4.16}$$

with multiple hyperparameters $\boldsymbol{\alpha}_m$ to weigh them accordingly by C_m. Therefore, (2.6) is a special case of (4.15) where the number of mixtures $M = 1$.

Yamamoto et al. (2003) reports a further perplexity reduction with DM than LDA on the experiments using Japanese newspaper corpus.

4.2.3 Problem of Context Models So Far

Although long-distance language models have been sophisticated lately as described above, they all share a critical deficiency as a context modeling. Because all of them are applications of probabilistic text modelings such as PLSI, LDA, and DM that use "bag-of-words" assumption of documents and words, all the latent variable context modelings above regard a history as a simple bag-of-words, totally dropping a chronological information that is crucial for context modeling. In fact, Beeferman et al. (1997a) found that the semantic effect of context decreases exponentially with time, by investigating the intervals of self triggers (the reappearance of the same word) found in a corpus.

Moreover, previous models also have problems when the context becomes long. Because they use all the words from the beginning of a document as a context, the estimation becomes slower and slower as it proceeds, as well as it gradually becomes useless by averaging over all the sequence observed so far, approximating a simple unigram.

Previously, Li and Yamanishi (2000) notices the same structure of text and proposes a probabilistic clustering approach like PLSI. However, they provide no generative process that accounts for context shifts, therefore rely on a heuristic parameters such as a fixed length window and an *ad hoc* segmentation threshold. Moreover, they are based on a maximum likelihood estimate as PLSI: that may cause a severe overfitting problem especially in context modeling where the amount of data available is generally small.

On the other hand, we propose a principled approach to this problem, using a Bayesian method that avoids overfitting and without no such *ad hoc* thresholds.

We use an explicit probabilistic generative model that describes the process of latent context shifts. Below, we introduce this process called a Mean Shift Model.

4.3. Mean Shift Model

4.3.1 HMM for Multinomial Distributions

The long-distance language models mentioned in section 2 all assume a hidden multinomial distribution, such as a mixture distribution over latent topics, or even a unigram distribution, to predict the next word by updating its estimate according to the context provided. Therefore, to track context shifts, we need a model that can describe the changes of multinomial distribution.

One model for this purpose is a multinomial extension of the Mean Shift model (MSM) (Yao, 1984; Chen and Lai, 2003) proposed in statistics.

This is a kind of HMM, but note that it is different from ordinary discrete HMMs. In discrete HMMs, the true state is one of M components and we estimate it stochastically as a multinomial distribution over the M components. On the other hand, since the true state here is itself a multinomial over the M components, we estimate it stochastically as a (possibly a mixture of) Dirichlet distribution, a distribution of multinomial distribution on the (M−1)-simplex.

This HMM has some similarity with Factorial HMM (Ghahramani and Jordan, 1995) in that its true state is a multinomial over discrete variables. However, we aim to track a multinomial random walk whose pattern is different from text to text, not directly estimating a single fixed dynamics between components as in FHMM.

Blei and Moreno (2001) approximate the posterior multinomial distribution of PLSI by a single component with maximum probability and propose an Aspect Hidden Markov Model by running a standard Baum-Welch algorithm to distinguish different texts joined together like a newswire.

However, to detect changes of subtopics within a document, we need to model the changes of whole multinomial distribution directly, because subtopic distribution will be present over the components that do not usually have maximum probability. From this point of view, this paper can be seen as a step toward an exact generalization of Blei and Moreno (2001).[2] Below, we introduce a mean

[2]Although in this chapter we propose a forward predictive language model, it is possible to extend it by a Monte Carlo forward-backward algorithm (Chen and Lai, 2003). However, this extension is beyond the scope of this chapter; it is a future work for us (section 6).

shift model for multinomial distribution.

4.3.2 Multinomial Mean Shift Model

Mean shift model (MSM) is a generative model which describes the intermittent changes of latent states and outputs according to them. Although there is a corresponding counterpart using Normal distribution first introduced (Chernoff and Zacks, 1964; Yao, 1984), here we concentrate on a multinomial extension of MSM, following Chen and Lai (2003) for DNA sequence modeling.

In a multinomial MSM, we assume time-dependent true multinomials θ_t that may change occasionally and the following generative model for discrete outputs $\mathbf{y}_T = y_1 y_2 \dots y_T$ ($y_t \in \mathcal{A}; \mathcal{A}$ is a set of alphabets).

$$\begin{cases} \theta_t \sim \mathrm{Dir}(\boldsymbol{\alpha}) & \text{with probability } \rho \\ \quad = \theta_{t-1} & \text{with probability } (1-\rho) \\ y_t \sim \mathrm{Mult}(\theta_t) & \end{cases} \tag{4.17}$$

where $\mathrm{Dir}(\boldsymbol{\alpha})$ and $\mathrm{Mult}(\theta)$ are a Dirichlet and multinomial distribution with parameters $\boldsymbol{\alpha}$ and θ, respectively. Here, we assume that the hyperparameter $\boldsymbol{\alpha}$ is known and fixed, an assumption we will relax in section 4.

This model first draws a multinomial θ from $\mathrm{Dir}(\boldsymbol{\alpha})$ and samples output y according to θ for some interval. When a change point occurs with probability ρ, new θ is sampled again from $\mathrm{Dir}(\boldsymbol{\alpha})$, and subsequent y is sampled from the new θ. This process continues recursively throughout which neither θ_t nor the change points are known to us; all we know is output sequence $\mathbf{y}_T$.

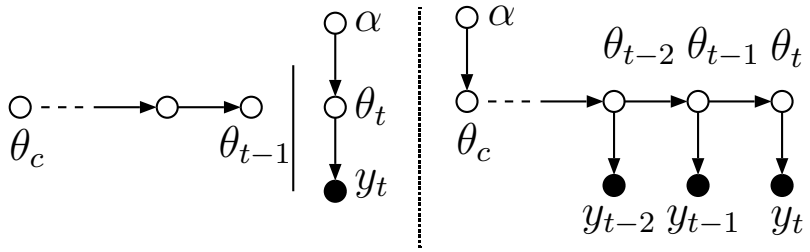
For example, consider a $T=100$ sequence in Figure 1 where a set of alphabets is $\mathcal{A}=\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. What is the next alphabet according to this sequence?

Apparently, this estimate depends on the last change point that is unknown to us. Let a binary variable be I_t that represents whether a change has occurred at time t: that is, $I_t=1$ means there was a change at time t ($\theta_t \neq \theta_{t-1}$), and $I_t=0$ means there was no change ($\theta_t = \theta_{t-1}$).

Case $I_t = 1$: According to the model (1), this case means $\theta = \theta_t \sim \mathrm{Dir}(\boldsymbol{\alpha})$ is newly sampled and y_t is output from θ as in Figure 4.6(a). Therefore, the

```
acaacacacaacaaccccccaaaccccccaacabcaabcaabbabbbbbbb\
bbbbcbbbbbbbbbbcbbabbbbbbbbbaabbaaacaaabbaaaababaa
```

Figure 4.5. Observed alphabet sequence.



(a) $I_t=1$ (b) $I_t=0$

Figure 4.6. Graphical Model of Mean Shifts.

probability of y_t given $\mathbf{y}_{t-1}$ and $I_t = 1$ is computed as

$$p(y_t|\mathbf{y}_{t-1}, I_t=1) = \int p(y_t|\theta)p(\theta|\boldsymbol{\alpha})d\theta \tag{4.18}$$

$$= \alpha_{y_t} / \textstyle\sum_y \alpha_y \,. \tag{4.19}$$

Case $I_t=0$: For this case, let the time of the last change be $t = c$ ($I_c = 1, I_{c+1} = \cdots = I_{t-1} = 0$). As seen in Figure 4.6(b), this means $\theta = \theta_c \sim \mathrm{Dir}(\boldsymbol{\alpha})$ was sampled at time c, and y_t is sampled after $y_c \cdots y_{t-1}$ has been sampled from θ. Therefore, an estimate of y_t based on those information is

$$\begin{aligned} p(y_t|\mathbf{y}_{t-1}, I_t=0) &= \int p(y_t|\theta)p(\theta|y_c \cdots y_{t-1})d\theta \\ &= \int \theta_{y_t} \cdot \mathrm{Dir}(\theta|\boldsymbol{\alpha} + \sum_{t'=c}^{t-1} \delta(y_{t'}))d\theta \\ &= \frac{\alpha_{y_t} + n_{y_t}}{\sum_y (\alpha_y + n_y)} \,, \end{aligned} \tag{4.20}$$

where $\delta(y)$ is a Dirac delta function at point y and n_y is the number of occurrences of y in $y_c \cdots y_{t-1}$.

The above derivation shows that once we know where the change occurred, predictive distribution for the next alphabet can be obtained in a closed form. Therefore, the essence of this problem lies in how to detect a change point given the data up to time t. In fact, this is a kind of *change point problem* common in statistics (Lee, 1997).

As we will see in the following sections, the change point estimate $p(I_t = 1)$ at time t depends on the previous change point, and this dependency goes recursively. Therefore, to solve this problem, we need at least a nonlinear dynamic programming.

However, to calculate that estimate requires that the previous change point be fixed deterministically. If we conduct a single sequence of Bernoulli trials on change points, we will fall into severe overfitting, resulting in quite unstable predictions. For this purpose, Sequential Monte Carlo method (Particle Filter) described in Chapter 2 offers stable prediction and further advantages that we will see.

For the Bayesian inference using parallel Bernoulli sampling through the Particle Filter, we must solve two subproblems. First, change point probability must be computed. Second, the particle weights must be updated recursively. Below, we describe the solutions to these problems in that order.

4.3.2.1 Change Point Detection by Particles

The first problem is to compute a change point probability at time t given the current observation y_t and the previous observations $\mathbf{y}_{t-1}$ and change points $\mathbf{I}_{t-1}$ up to time $(t-1)$: $p(I_t = 1|\mathbf{I}_{t-1}, \mathbf{y}_t)$. Here, $\mathbf{I}_{t-1}$ be a binary change point history already sampled up to time $t-1$: $\mathbf{I}_{t-1} = \{I_1 \dots I_{t-1}\}$.

Using Bayes' theorem,

$$
\begin{aligned}
p(I_t|\mathbf{I}_{t-1}, \mathbf{y}_t) &\propto p(I_t, y_t|\mathbf{I}_{t-1}, \mathbf{y}_{t-1}) && (4.21)\\
&= p(y_t|\mathbf{y}_{t-1}, I_t, \mathbf{I}_{t-1}) p(I_t|\mathbf{I}_{t-1}, \mathbf{y}_{t-1}) && (4.22)\\
&\simeq p(y_t|\mathbf{y}_{t-1}, I_t, \mathbf{I}_{t-1}) p(I_t|\mathbf{I}_{t-1}) && (4.23)\\
&= \begin{cases} p(y_t|\mathbf{y}_{t-1}, \mathbf{I}_{t-1}, I_t = 1) p(I_t = 1|\mathbf{I}_{t-1}) =: f(t) \\ p(y_t|\mathbf{y}_{t-1}, \mathbf{I}_{t-1}, I_t = 0) p(I_t = 0|\mathbf{I}_{t-1}) =: g(t)\,, \end{cases} && (4.24)
\end{aligned}
$$

where we assumed in (4.23) that a *prior* probability of change only depends on previous changes.

Let two expressions in (4.24) be $f(t)$ and $g(t)$. Then

$$\begin{cases} p(I_t{=}1|\mathbf{I}_{t-1}, \mathbf{y}_t) = f(t) \,/\, (f(t) + g(t)) \\ p(I_t{=}0|\mathbf{I}_{t-1}, \mathbf{y}_t) = g(t) \,/\, (f(t) + g(t)) \,. \end{cases} \tag{4.25}$$

In the expression (4.24), the first term is a likelihood of observed output y_t when we know the change point, which can be calculated by (4.19) and (4.20). The second term is a prior probability of the context change, which can be set tentatively by constant ρ: thus $f(t)$ and $g(t)$ are obtained to give (4.25).

Actually, in a Particle Filter approach, each particle has a binary sequence of change point history $\mathbf{I}_{t-1}$, thus we can estimate ρ online on $\mathbf{I}_{t-1}$.

Specifically, by considering ρ a random variable whose prior distribution is a Beta distribution $\rho \sim \mathrm{Be}(\alpha, \beta)$, we get an estimate of ρ_t using the number of 1's and 0's in $\mathbf{I}_{t-1}$, $n_{t-1}(1)$ and $n_{t-1}(0)$, as an expectation of posterior Beta distribution in a standard Bayesian method (Liu, 2001) :

$$E[\rho_t|\mathbf{I}_{t-1}] = \int \rho_t p(\rho_t|\mathbf{I}_{t-1}) d\rho_t \tag{4.26}$$

$$= \int \rho_t \cdot \mathrm{Be}\left(\rho_t|\alpha + n_{t-1}(1), \beta + n_{t-1}(0)\right) d\rho_t \tag{4.27}$$

$$= \frac{\alpha + n_{t-1}(1)}{\alpha + \beta + t - 1}. \tag{4.28}$$

This means that we can estimate a "rate of topic shift" as time proceeds in a Bayesian fashion. Throughout the following experiments, we used this online estimate of ρ.

4.3.2.2 Particle Weight Updates

The second problem is to update particle weights w_t from w_{t-1} by the general formula (2.99), reproduced here convenience:

$$w_t \propto w_{t-1} \cdot p(y_t|\mathbf{x}_{t-1}) \,. \tag{4.29}$$

Here, since the general state $\mathbf{x}_{t-1}$ consists of a pair of variables $(\mathbf{y}_{t-1}, \mathbf{I}_{t-1})$ in the MSM, we get

$$\begin{aligned} p(y_t|\mathbf{x}_{t-1}) &= p(y_t|\mathbf{y}_{t-1}, \mathbf{I}_{t-1}) && (4.30)\\ &= \textstyle\sum_{I_t\in\{0,1\}} p(y_t, I_t|\mathbf{y}_{t-1}, \mathbf{I}_{t-1}) && (4.31)\\ &= \textstyle\sum_{I_t\in\{0,1\}} p(y_t|I_t, \mathbf{I}_{t-1}, \mathbf{y}_{t-1})p(I_t|\mathbf{I}_{t-1}) && (4.32)\\ &= f(t)+g(t)\,. && (4.33) \end{aligned}$$

Hence we can update particle weight easily using $f(t)$ and $g(t)$ calculated in (4.24).

4.3.3 Multinomial Particle Filter

Now we have a Particle Filter algorithm for the multinomial MSM from the derivations above, graphically displayed in Figure 4.12 (excluding prior updates):

1. For particles $i = 1 \ldots N$,
   (a) Calculate $f(t)$ and $g(t)$ according to (4.24).
   (b) Sample $I_t^{(i)} \sim$ Bernoulli $(f(t)/(f(t)+g(t)))$, and update $\mathbf{I}_{t-1}^{(i)}$ to $\mathbf{I}_t^{(i)}$.
   (c) Update weight $w_t^{(i)} = w_{t-1}^{(i)} \cdot (f(t)+g(t))$.
2. Find a predictive distribution by (2.96), using $w_t^{(1)} \ldots w_t^{(N)}$ and $\mathbf{I}_t^{(1)} \ldots \mathbf{I}_t^{(N)}$ as well as the data observed so far.

The above algorithm runs for each observation y_t $(t = 1 \ldots T)$. If we observe a "strange" word that is more predictable from the prior than the contextual distribution, (4.24) may make $f(t)$ larger than $g(t)$, which leads to a higher probability that $I_t = 1$ will be sampled in the Bernoulli trial of the algorithm 1(b).

Although this sampling of context change is probabilistic and also relaxed by the N particles, it will occur somewhere during the subsequent observations if they actually indicate latent changes.

Additionally, step 1(c) in fact includes a step called *resampling* when the weights become too biased. It "kills" the infinitesimal weight particles and makes "children" of the heavier particles to adapt to the observed data. This means that in our case the process gradually selects particles that have appropriate Beta

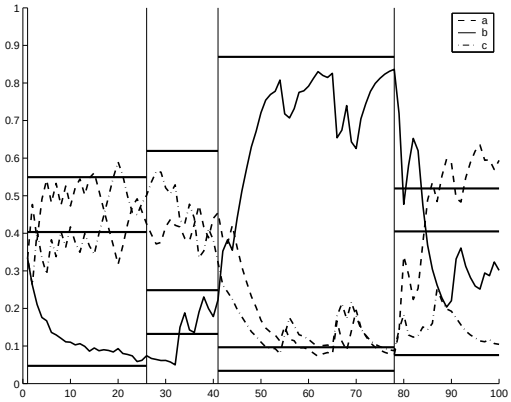


Figure 4.7. Particle Filter estimate of latent multinomial in Figure 4.5. Horizontal lines show the true distribution.

distribution of context change from the data observed so far.[3] For the criterion of biasedness of particles weights, the coefficient of variation (CV) is known to be useful (Doucet et al., 2001). In the following experiments, we used the CV threshold 1 or 2 by preliminary investigations.

Figure 4.7 shows a Particle Filter estimate of latent multinomial θ_t that lies in the example sequence of Figure 4.5. Note that this is a "forward" estimate that utilizes only backward data $y_1 .. y_t$ to estimate θ_t.

4.4. Mean shift model of Natural Language

Chen and Lai (2003) used this algorithm to analyze DNA sequences. However, when extending this approach to natural language, i.e. word sequences, we meet two serious problems.

The first problem is that in natural languages the number of words is extremely large. As opposed to DNAs that have only four alphabets of A/T/G/C, a natural language usually has a minimum of some tens of thousands of words as its alphabets, and they have strong positive and negative correlations between them.

For example, if 'nurse' follows 'hospital' we believe that there has been no

[3]From this point of view, the Particle Filter algorithm has some similarity to a beam search or a genetic algorithm.

context shift: but if 'university' follows 'hospital', the context probably has been shifted to a "medical school" subtopic, even though the two words are equally distinct from 'hospital.' Of course, this is due to the semantic relationship we can assume between these words. However, the original multinomial MSM cannot capture this relationship because it treats the alphabets independently. Therefore, in the original modeling, 'hospital' after 'nurse' may be erroneously interpreted as a sign of context change. To incorporate these relationship, we require a probabilistic model that works as an extensive prior knowledge of words.

The second problem is that in model equation (4.17), the parameter $\boldsymbol{\alpha}$ of prior Dirichlet distribution of the latent multinomial is assumed to be known. In the case of natural language, this means we know beforehand what words or topics will be spoken for all of the texts. Apparently, this is not a natural assumption: we need an online estimation of $\boldsymbol{\alpha}$ as well when we want to extend MSM to natural language.

To solve these problems, we extended a multinomial MSM using probabilistic text models DM and LDA. Below we introduce MSM-DM and MSM-LDA, in this order.

4.4.1 MSM-DM

When we combine the original MSM with the Dirichlet Mixture (DM) language model described in section 4.2.2.3, we get a natural extension of MSM using a mixture of Dirichlet distributions rather than a single Dirichlet distribution as a prior distribution for the multinomials.

The problem of the original MSM lies in the simple prediction (4.20) and (4.19) of each particle, reproduced here for convenience:

$$p(y|\mathbf{y}_{t-1}, I_t=1) = \alpha_y / \textstyle\sum_y \alpha_y \tag{4.34}$$

$$p(y|\mathbf{y}_{t-1}, I_t=0) = \frac{\alpha_y + n_y}{\sum_y (\alpha_y + n_y)} \,. \tag{4.35}$$

Especially for (4.35), this means that only the probabilities of words seen in the history are increased. For the other words that have high relevancies with the words in the history, their probabilities remain constant however related they

1. Draw $m \sim \text{Mult}(\boldsymbol{\lambda})$.
2. For $t = 1 \cdots T$,
   (a) Draw $c \sim \text{Binomial}(\rho)$.
   (b) If $c = 0$ then Set $\boldsymbol{p}_t = \boldsymbol{p}_{t-1}$.
   If $c = 1$ then Draw $\boldsymbol{p}_t \sim \text{Dir}(\boldsymbol{\alpha}_m)$.
   (c) Draw $y_t \sim \text{Mult}(\boldsymbol{p}_t)$.

Figure 4.8. Generative Model of MSM-DM.

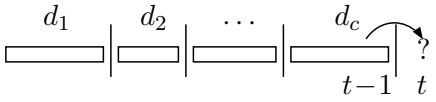


Figure 4.9. History segmented into "pseudo documents" by the change points.

are. When we replace (4.35) by a Dirichlet Mixture prediction (4.13):

$$p(y|\mathbf{y}_{t-1}, I_t = 0, \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M) = \sum_{m=1}^{M} C_m \frac{\alpha_{my} + n_y}{\sum_y (\alpha_{my} + n_y)}, \tag{4.36}$$

we get a flexible model MSM-DM that considers both the semantic correlations and the dynamic mean shift property of natural language.

Combining MSM and DM, MSM-DM has the following generative model for the discrete observations $\mathbf{y}_T = y_1 \dots y_T$.

First, this process selects a model m from the M hyperparameters.

Second, the mean shift process generates the observations $\mathbf{y}_T = y_1 \dots y_T$ with occasional resampling of latent unigram $\boldsymbol{p}_t$ from the m'th Dirichlet distribution with the probability ρ.

Since the model index m is drawn only once at first, we also need to infer the posterior estimate of m, $p(m)$ online rather than using a prior value λ_m. Fortunately, a maximum likelihood estimate of $p(m)$ can be inferenced in the Particle Filter approach.

In the Particle Filter estimation, for each particle the history is segmented into pseudo "documents" $d_1 \dots d_c$ by the change points sampled so far (Figure 4.9).

1. $t = 0$: *Initalize particles* 1 .. N.
2. For $t = 1$ .. $T - 1$,
   (a) For $j = 1 \ldots N$,
      1. Compute $f(t)$ and $g(t)$ as the observation probability $p(y_t)$ from DM posterior and DM prior.
      2. Update particle weight $w_t^{(j)} = w_{t-1}^{(j)} \cdot (f(t) + g(t))$.
      3. Draw $c \sim \text{Bernoulli}(f(t)/(f(t) + g(t)))$.
      4. If $c = 1$ then
         – Compute p_{im} from the last change point;
         – Update $\boldsymbol{\lambda} \propto \sum_m p_{im}$.
   (b) Normalize particle weights.
   (c) Predict $p(y_{t+1})$ by the particles 1 .. N and weights $w_t^{(1)}$ .. $w_t^{(N)}$.
   (d) If weights $w_t^{(1)}$ .. $w_t^{(N)}$ are too biased, conduct *resampling* to regenerate new particles.

Figure 4.10. Particle Filter algorithm of MSM-DM.

Since each "pseudo document" d_i $(i = 1 \ldots c)$ has a model posterior

$$p(m|d_i) \propto \lambda_m \frac{\Gamma(\sum_v \alpha_{mv})}{\Gamma(\sum_v \alpha_{mv} + |d_i|)} \prod_{v=1}^{V} \frac{\Gamma(\alpha_{mv} + n_{iv})}{\Gamma(\alpha_{mv})}, \tag{4.37}$$

the maximum likelihood estimate of the common prior $p(m|\mathbf{h})$ is given by a simple summation

$$p(m|\mathbf{h}) \propto \sum_{i=1}^{c} p(m|d_i), \tag{4.38}$$

the same formula used in the M step of the DM parameter estimation in Chapter 2. For this purpose, only the sufficient statistics $p(m|d_i)$ $(i = 1 \ldots c)$ need to be stored for each particle to render itself an online algorithm.

Using this online inference of the posterior estimate of the mixing parameter $\boldsymbol{\lambda}$, $p(m|\mathbf{h})$, the final Particle Filter estimation algorithm of MSM-DM is depicted in Figure 4.10.

1. For $t = 1 \dots T$,
   (a) Draw $c \sim \text{Binomial}(\rho)$.
   (b) If $c = 0$ then Set $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1}$.
   If $c = 1$ then Draw $\boldsymbol{\lambda}_t \sim \text{Dir}(\alpha)$.
   (c) Draw $z \sim \text{Mult}(\boldsymbol{\lambda})$.
   (d) Draw $w_t \sim \text{Mult}(\boldsymbol{\beta}_z)$.

Figure 4.11. Generative Model of MSM-LDA.

4.4.2 MSM-LDA

When we combine MSM with the Latent Dirichlet Allocation (LDA) (Blei et al., 2001), we get another interesting model, MSM-LDA, that tracks *latent topic distribution* other than the unigram distribution as MSM-DM.

As we have seen in section 4.2.2.2, given a history $\mathbf{h} = w_1 w_2 \cdots w_h$ the posterior distribution of topic mixture $\boldsymbol{\lambda} = \{p(t|\mathbf{h})\}_{t=1}^K$, namely a Dirichlet distribution $p(\boldsymbol{\lambda}|\mathbf{h}) = \text{Dir}(\boldsymbol{\lambda}|\boldsymbol{\alpha})$, can be computed by the variational Bayes EM algorithm of Figure 4.3. Therefore, regarding this $\boldsymbol{\lambda}$ as the latent multinomial θ in the general mean shift model of (4.17), we get a model that tracks online the multinomial mixing distribution $\boldsymbol{\lambda}$ of the latent topics. When the current estimate of $\boldsymbol{\lambda}_t$, $q(\boldsymbol{\lambda}_t|\mathbf{h})$ is obtained, predictive distribution of words can be computed by the mixture of the class unigrams of LDA with the expectation of $p(\boldsymbol{\lambda}_t|\mathbf{h})$:

$$p(y|\mathbf{h}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \int p(y, \boldsymbol{\lambda}|\mathbf{h}, \boldsymbol{\alpha}, \boldsymbol{\beta}) d\boldsymbol{\lambda} \tag{4.39}$$

$$= \sum_{t=1}^{K} p(y|t) \langle \lambda_z \rangle_{q(\boldsymbol{\lambda}|\mathbf{h})} , \tag{4.40}$$

as we also showed in (4.11).

MSM-LDA has the generative model of Figure 4.11 for the observations $\mathbf{y}_T = y_1 \cdots y_T$ given the LDA parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta} = \{\boldsymbol{\beta}_t\}_{t=1}^K = \{p(w|t)\}_{t=1}^K$. In the step 1 of Figure 4.11, we assume $c = 1$ is always sampled at $t = 1$.

The remaining problem here is that the hyperparameter $\boldsymbol{\alpha}$ in the Step 2 is assumed to be known and fixed. Because $\boldsymbol{\alpha}$ governs the Dirichlet distribution

representing topic mixture $\boldsymbol{\lambda}$, assuming $\boldsymbol{\alpha}$ to be known means that the same topic mixture will be assumed *a priori* for all the texts to be processed. Apparently, this is not a natural assumption: we also need to estimate the hyperparameter $\boldsymbol{\alpha}$ online.

Fortunately, also in this case a maximum likelihood online estimate of $\boldsymbol{\alpha}$ can be obtained in the Particle Filter approach. Since the history has been segmented into "pseudo documents" $d_1 \dots d_c$ by the change points as in MSM-DM, for each d_i a posterior Dirichlet distribution $q(\boldsymbol{\lambda}|d_i)$ $(i = 1 \dots c)$ can be computed. As described in Chapter 2, we can infer the common Dirichlet prior $\mathrm{Dir}(\boldsymbol{\lambda}|\boldsymbol{\alpha})$ by a linear-time Newton-Raphson method efficiently.

Specifically, given the parameters $\boldsymbol{\gamma}_1 \dots \boldsymbol{\gamma}_c$ of the Dirichlet posteriors $p(\boldsymbol{\lambda}|d_1) \dots p(\boldsymbol{\lambda}|d_c)$, $\boldsymbol{\alpha}$ can be computed by a Newton-Raphson iteration

$$\boldsymbol{\alpha}' = \boldsymbol{\alpha} - H^{-1}\mathbf{g}\,, \tag{4.41}$$

where $\mathbf{g}$ and the Hessian H are

$$g_i = M\{\Psi(\sum_i \alpha_i) - \Psi(\alpha_i)\} + \sum_{d=1}^{M}\{\Psi(\gamma_{di} - \Psi(\sum_i \gamma_{di})\}\,, \tag{4.42}$$

$$h_i = -\Psi(\alpha_i)\,, \tag{4.43}$$

$$z = \Psi'(\sum_i \alpha_i)\,, \tag{4.44}$$

$$H = \mathrm{diag}(\mathbf{h}) + \mathbf{1}z\mathbf{1}^T \tag{4.45}$$

where H can be inverted in linear time. For the details of the derivation, see Chapter 2.

Also in MSM-LDA, only the sufficient statistics $p(\boldsymbol{\lambda}|d_i)$ $(i = 1 \dots c)$ must be stored for each particle to make itself an efficient filtering algorithm.

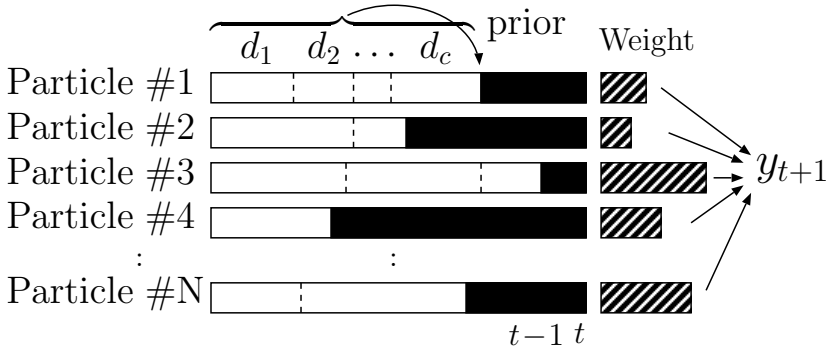


Figure 4.12. Proposed Particle Filter for Contexts, and Sequential Updates of Priors.

4.5. Experiments

We conducted experiments using a British National Corpus (BNC) (Burnage and Dunlop, 1992). BNC is a balanced corpus that has long texts on quite various topics, appropriate for our experimental objectives. We randomly selected 100 files of BNC written texts as an evaluation set, and the remaining 2,943 files as a training set for parameter estimation of LDA and DM.

4.5.1 Training and Evaluation Data

Training Data Since LDA and DM proved unable to be used for long texts like BNC, we divided them into segments of a minimum of 10 sentences[4] to create pseudo documents for LDA/DM parameter estimation. Due to the huge size of the BNC, we randomly selected a maximum of 20 pseudo documents from each of the 2,943 files to produce a final corpus of 56,939 pseudo documents that consisted of a total 11,032,233 words. This data amounts to a random 1/10 of the BNC corpus as a whole. We used a lexicon of 52,846 words with a frequency ≥ 5.

Evaluation Data The proposed method is an algorithm that captures topic shifts and their rate in a text to predict the next word. Therefore, we need evaluation texts that have different topic shift rates.

For this purpose, we prepared four kinds of texts by sampling from the long BNC texts. Specifically, we conducted sentence-based random sampling as fol-

[4]For the rest of this paper, we call a BNC segment divided by `<s>..</s>` a "sentence."

1. $t = 0$: *Initalize particles* 1 .. N.
2. For $t = 1$ .. $T - 1$,
   (a) For $j = 1 \ldots N$,
      1. Compute $f(t)$ and $g(t)$ as the observation probability $p(y_t)$ from LDA posterior and LDA prior via VB-EM.
      2. Update particle weight $w_t^{(j)} = w_{t-1}^{(j)} \cdot (f(t) + g(t))$.
      3. Draw $c \sim \text{Bernoulli}(f(t)/(f(t) + g(t)))$.
      4. If $c = 1$ then
         – Compute γ_{im} from the last change point via VB-EM.
         – Update $\boldsymbol{\alpha}$ using γ_{im} by a Newton-Raphson method (4.41).
   (b) Normalize particle weights.
   (c) Predict $p(y_{t+1})$ by the particles 1 .. N and weights $w_t^{(1)}$ .. $w_t^{(N)}$: Weight each particle's prediction obtained via the VB-EM.
   (d) If weights $w_t^{(1)}$ .. $w_t^{(N)}$ are too biased, conduct *resampling* to generate new particles.

Figure 4.13. Particle Filter algorithm of MSM-LDA.

lows.

(1) Select a first sentence randomly for each text.
(2) Sample contiguous X sentences from that sentence.
(3) Skip Y sentences.
(4) Continue steps (2) and (3) until a desired length of text is obtained.

In the procedure above, X and Y are random variables that have uniform distributions in Table 4.1. We sampled 100 sentences from each of the 100 files by this procedure to create four types of evaluation text sets listed in the table.

Name	Property
Raw	$X = 100, Y = 0$
Slow	$1 \leq X \leq 10, 1 \leq Y \leq 3$
Fast	$1 \leq X \leq 10, 1 \leq Y \leq 10$
VeryFast	$X = 1, 1 \leq Y \leq 10$

Table 4.1. Types of Evaluation Texts.

4.5.2 Parameter Settings

The number of latent classes in LDA and DM are set to 200 and 50, respectively. Although this number was chosen mainly due to computational limitations, larger parameters produced virtually the same performance in a preliminary experiment.[5] The number of particles was set to $N = 20$. This figure is relatively small because each particle can conduct an exact Bayesian prediction of multinomial distribution, once previous change points have been sampled.

Beta prior distribution of context change can be initialized as a uniform distribution, $(\alpha, \beta) = (1, 1)$. However, based on a preliminary experiment we set it to $(\alpha, \beta) = (1, 50)$: this means we initially assume a context change rate of once every 50 words in average, which will be updated adaptively.

4.5.3 Experimental Results

Table 4.2 shows the unigram perplexity of contextual prediction for each type of evaluation sets. While MSM-LDA slightly improves LDA due to the topic space compression explained in section 4.1, MSM-DM produces a consistently better prediction, and its performance is more significant for texts whose subtopics change faster.

Figure 4.14 shows a plot of the actual improvements relative to DM, $\mathrm{PPL}_{\mathrm{MSM}} - \mathrm{PPL}_{\mathrm{DM}}$. We can see that prediction improves for most documents by automatically selecting appropriate contexts. The maximum improvement was –365 in PPL for one of the evaluation texts.

[5] We deliberately chose a smaller number of mixtures in DM because it is reported to have a better performance in small mixtures since it is essentially a unitopic model as opposed to LDA.

Text	MSM-DM	DM	MSM-LDA	LDA
Raw	**870.06** (−6.02%)	925.83	**1028.04**	1037.42
Slow	**893.06** (−8.31%)	974.04	**1047.08**	1060.56
Fast	**898.34** (−9.10%)	988.26	**1044.56**	1061.01
VFast	**960.26** (−7.57%)	1038.89	1065.15	1050.83

Table 4.2. Contextual Unigram Perplexities for Evaluation Texts.

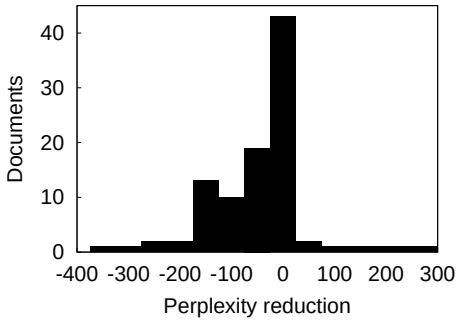


Figure 4.14. Perplexity reductions relative to DM.

However, since our method samples change points word by word, it is sometimes unstable to noisy word, which causes radically bad prediction infrequently. To avoid this, we require the change points be sampled by sentences or by paragraphs; however, because the underlying generative model assumes a word base generation, this is not trivial. Even no systematic method are found in SMC that can handle multiple observation at once (Kwok et al., 2002).

Finally, we show in Figure 4.15 a sequential plot of context change probabilities $p^{(i)}(I_t=1)$ $(i = 1..N, t = 1..T)$ calculated by particles for the first 1,000 words of one of the evaluation texts.

As seen in the figure, our method can also be considered a probabilistic variant of TextTiling as an ancillary process to estimate topic segments online.

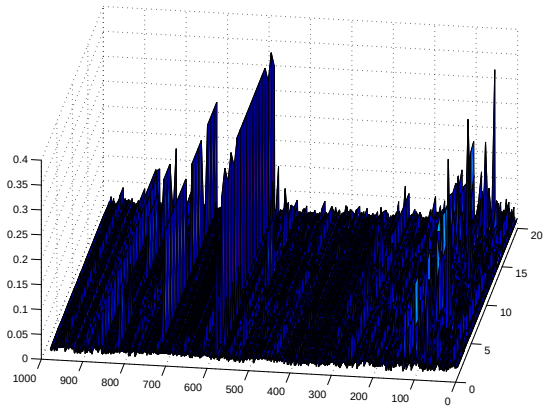


Figure 4.15. Context change probabilities by Particles.

4.6. Summary and Future Directions

In this chapter, we gave a novel Bayesian statistical language model of context that recognizes topic shifts and their rate on-line. It makes an optimal prediction using a mixture of different length of histories sampled by a multinomial Particle Filter.

The heart of our approach lies in twofold:

1) We view a context estimation as a multinomial filtering problem rather than a simple averaging by introducing a latent state space model that is recently proposed in the DNA sequence modeling to account for topic shifts, and solve it by a sequential Monte Carlo method (Particle Filter).

2) We extend the original multinomial filter to incorporate an extremely large number of symbols and strong semantic correlations between them by combining the filter with Bayesian probabilistic text models LDA and DM, which give prior semantic prior knowledge between the symbols.

As a result, we obtain a Bayesian long-distance language model that has the lowest perplexity in the current state of art. Experiments on the standard British National Corpus confirmed that the proposed model can track the contexts in real data.

Moreover, proposed method is not only a model in natural language but also useful for a filtering in similar high-dimensional discrete data domains that have

high correlations between them, and a generic mixture modeling such as Gaussian mixtures.

Although this thesis concentrates on a predictive language modeling, it can be extended in several ways to model a textual heterogeneity more appropriately. The first is a nonlinear forward-backward extension; this can be executed by a similar Monte Carlo method, or an analytical approximation such as Expectation Propagation (Minka, 2001), for example. The second is an extension to a collection of documents. Since probabilistic text models such as LDA and DM is incapable to deal with a corpora of long texts, this extension opens up a new way to handle long texts such as literary texts, ordinary books, and sequential speech transcripts.

Through such extensions, we hope to find a more appropriate "semantic unit" of texts statistically beyond a *document* that has been assumed naïvely for that purpose, into a statistical interpretation of Saussurean "Paradigme" of meaning (de Saussure, 1916).

Chapter 5

Conclusion

5.1. Summary of this dissertation

The goal of this dissertation is to propose a more sophisticated treatment of distributional models, specifically, along the two kinds of units of semantic processing:

1. *Static Units.* Since for a specified task the useful units for that objective are often already given, we wish to use a more elaborated method of semantic treatment for them by utilizing our prior knowledge of "similarity."

2. *Dynamic Units.* The most natural units of semantic processing are often hidden and embedded within the static units just described. We wish to reveal such units of semantic coherence by a statistical inference to be utilized more effectively in natural language processing.

To achieve these goals, this dissertation proposed the following solutions in Chapter 3 and Chapter 4, respectively.

In Chapter 3, we proposed a novel metric distance function between the static units in place of the Euclidean distance extensively used in natural language processing so far. This metric is derived from our prior knowledge of "similarity" as cluster structures in the training data by a semi-supervised learning approach.

Contrary to the previously proposed method in the field other than in natural language processing, this metric can be computed in closed form using the whole data at once without any iterative optimization.

We confirmed the effect of proposed metric distance function on the retrieval and clustering task of documents and sentences, as well as on a general vectorial dataset commonly used in Machine Learning research.

In Chapter 4, we relaxed the i.i.d. assumption of words in a text that has been implicitly assumed for text modelings by introducing a Mean shift model known in statistics. We extended a Multinomial Mean shift model to natural language by combining it with LDA and DM, Bayesian text models recently proposed, to propose two extensions of Multinomial Mean shift model to deal with the huge number of symbols and their semantic correlations in natural language.

Essentially, this model is a nonlinear HMM. For an online inference, we used a Bayesian nonlinear filtering algorithm called a Particle Filter to track hidden changes of context dynamically in the long-distance statistical language modeling framework. This model amounts to finding the units dynamically in text within which i.i.d. property is considered preserved, and we make an optimal prediction of next word based on these dynamic units inferenced on the fly. As a result, we obtain a Bayesian long-distance statistical language model that has the lowest perplexity in the current state of art. We confirmed the performance of proposed model on various types of texts from the standard British National Corpus.

Through the proposed methodological sophistications along the two kinds of units, we expect more natural, less fixed natural language processing that deal with semantic aspects of language.

5.2. Future Work

Although we proposed the solutions to two kinds of units independently, ideally they must be fused into a unified framework.

In the statistical language modeling problem in Chapter 4, we implicitly assumed that the simplical manifold, i.e. the word simplex and the topic subsimplex, is isotropic without no prior information of actual data distribution within the manifold. We expect to relax this assumption by a geometrical approach that is similar in spirit to the metric learning problem in Chapter 2 by a Bayesian learning method.

Besides an integration of two approaches, separate modelings also require

more sophistication.

In the metric learning problem in Chapter 3, we noticed the necessity of an extension to kernel Hilbert space by the same objective of minimum cluster distortions. By this extension, dimensionality reductions are not necessary and the method can be extended to more general data structures, such as trees or graphs, where appropriate kernels have been already defined.

In the statistical language modeling problem in Chapter 4, we focused on the forward estimation in a language modeling framework. However, as mentioned in the summary of Chapter 4, this process can be extended to forward-backward estimation, or even to the learning from a collection of documents as a more elaborated text modeling.

Through such extensions, we will obtain a clearer perspective of semantic heterogeneities existent and essential to natural language.

References

[Bach and Jordan2003] Francis R. Bach and Michael I. Jordan. 2003. Learning Spectral Clustering. In *NIPS 2003*.

[Baeza-Yates and Ribeiro-Neto1999] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. 1999. *Modern Information Retrieval*. ACM Press / Addison-Wesley.

[Beeferman et al.1997a] Doug Beeferman, Adam Berger, and John Lafferty. 1997a. A Model of Lexical Attraction and Repulsion. In *Proc. of ACL-EACL '97*, pages 373–380.

[Beeferman et al.1997b] Doug Beeferman, Adam Berger, and John Lafferty. 1997b. Text Segmentation Using Exponential Models. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 35–46.

[Bellegarda1998] Jerome R. Bellegarda. 1998. A Multispan Language Modeling Framework for Large Vocabulary Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, 6(5):468–475.

[Blake and Merz1998] C. L. Blake and C. J. Merz. 1998. UCI Repository of machine learning databases. http://www.ics.uci.edu/˜mlearn/MLRepository.html.

[Blei and Moreno2001] David Blei and Pedro Moreno. 2001. Topic Segmentation with an Aspect Hidden Markov Model. In *Proc. of SIGIR 2001*, pages 343–348. ACM Press.

[Blei et al.2001] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2001. Latent Dirichlet Allocation. In *Neural Information Processing Systems 14*.

[Blei et al.2003] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

[Burnage and Dunlop1992] Gavin Burnage and Dominic Dunlop. 1992. Encoding the British National Corpus. *English Language Corpora: Design, Analysis and Exploitation*, pages 79–95.

[Chen and Goodman1996] Stanley F. Chen and Joshua Goodman. 1996. An Empirical Study of Smoothing Techniques for Language Modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 310–318.

[Chen and Lai2003] Yuguo Chen and Tze Leung Lai. 2003. Sequential Monte Carlo Methods for Filtering and Smoothing in Hidden Markov Models. Discussion Paper 03-19, Institute of Statistics and Decision Sciences, Duke University.

[Chernoff and Zacks1964] H. Chernoff and S. Zacks. 1964. Estimating the Current Mean of a Normal Distribution Which is Subject to Changes in Time. *Annals of Mathematical Statistics*, 35:999–1018.

[Choi2000] Freddy Y. Y. Choi. 2000. Advances in domain independent linear text segmentation. In *Proceedings of NAACL-00*.

[Collins and Duffy2001] Michael Collins and Nigel Duffy. 2001. Convolution Kernels for Natural Language. In *NIPS 2001*.

[de Saussure1916] Ferdinand de Saussure. 1916. *Cours de linguistique generale*. Paris:Payot.

[Deerwester et al.1990] S. Deerwester, Susan T. Dumais, and George W. Furnas. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

[Dhillon and Modha2001] Inderjit S. Dhillon and Dharmendra S. Modha. 2001. Concept Decompositions for Large Sparse Text Data Using Clustering. *Machine Learning*, 42(1/2):143–175.

[Doucet et al.2001] Arnaud Doucet, Nando de Freitas, and Neil Gordon. 2001. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer-Verlag.

[Doucet1998] Arnaud Doucet. 1998. On Sequential Simulation-Based Methods for Bayesian Filtering. Technical Report CUED/F-INFENG/TR 310, Department of Engineering, Cambridge University.

[Duda et al.2000] Richard O. Duda, Peter E. Hart, and David G. Stork. 2000. *Pattern Classification *Second Edition.* John Wiley & Sons.

[Florian and Yarowsky1999] Radu Florian and David Yarowsky. 1999. Dynamic Nonlocal Language Modeling via Hierarchical Topic-Based Adaptation. In *Proceedings of ACL'99*, pages 167–174.

[Ghahramani and Jordan1995] Zoubin Ghahramani and Michael I. Jordan. 1995. Factorial Hidden Markov Models. In *Advances in Neural Information Processing Systems (NIPS)*, volume 8, pages 472–478. MIT Press.

[Gildea and Hofmann1999] Daniel Gildea and Thomas Hofmann. 1999. Topic-based Language Models Using EM. In *Proc. of EUROSPEECH '99*, pages 2167–2170.

[Gilks et al.1996] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. 1996. *Markov Chain Monte Carlo in Practice.* Chapman & Hall / CRC.

[Good1953] I.J. Good. 1953. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264.

[Griffiths and Steyvers2002] T.L. Griffiths and M Steyvers. 2002. A probabilistic approach to semantic representation. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, pages 381–386.

[Hearst1994] Marti Hearst. 1994. Multi-paragraph segmentation of expository text. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 9–16.

[Hofmann1999] Thomas Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *Proc. of SIGIR '99*, pages 50–57.

[Ishikawa et al.1998] Yoshiharu Ishikawa, Ravishankar Subramanya, and Christos Faloutsos. 1998. MindReader: Querying Databases Through Multiple Examples. In *Proc. 24th Int. Conf. Very Large Data Bases*, pages 218–227.

[Jaakkola and Haussler1999] Tommi S. Jaakkola and David Haussler. 1999. Exploiting generative models in discriminative classifiers. In *Proc. of the 1998 Conference on Advances in Neural Information Processing Systems*, pages 487–493.

[Jelinek1998] Frederick Jelinek. 1998. *Statistical Methods for Speech Recognition.* Language, Speech, and Communication Series. MIT Press.

[Jiang and Berry1998] Eric P. Jiang and Michael W. Berry. 1998. Information Filtering Using the Riemannian SVD (R-SVD). In *Proc. of IRREGULAR '98*, pages 386–395.

[Joachims1998] Thorsten Joachims. 1998. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98*, number 1398, pages 137–142.

[Jordan et al.1999] Michael I. Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence K. Saul. 1999. An Introduction to Variational Methods for Graphical Models. *Machine Learning*, 37(2):183–233.

[Kneser and Ney1995] Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-grma language modeling. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 181–184.

[Kurohashi and Ori2000] Sadao Kurohashi and Manabu Ori. 2000. Nonlocal Language Modeling based on Co-occurence Vectors. In *Proc. of EMNLP/VLC '00*, pages 80–86.

[Kwok et al.2002] Cody Kwok, Dieter Fox, and Marina Meilă. 2002. Real-time Particle Filters. In *Advances in Neural Information Processing Systems 15.*

[Landauer and Dumais1997] T.K. Landauer and S. T. Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240.

[Lang1995] Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339.

[Lee1997] Peter M. Lee. 1997. *Bayesian Statistics: An Introduction.* Arnold Publishers, Second edition.

[Li and Yamanishi2000] Hang Li and Kenji Yamanishi. 2000. Topic Analysis Using a Finite Mixture Model. In *Proc. of EMNLP-VLC'00*, pages 35–44.

[Liu2001] Jun S. Liu. 2001. *Monte Carlo Strategies in Scientific Computing.* Springer Series in Statistics. Springer-Verlag.

[MacKay and Peto1994] D. J. C. MacKay and L. Peto. 1994. A Hierarchical Dirichlet Language Model. *Natural Language Engineering*, 1(3):1–19.

[Manning and Schütze1999] Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing.* MIT Press.

[Minka2000a] Thomas P. Minka. 2000a. Beyond Newton's method. http://research.microsoft.com/˜minka/papers/newton.html.

[Minka2000b] Thomas P. Minka. 2000b. Estimating a Dirichlet distribution. http://www.stat.cmu.edu/˜minka/papers/dirichlet/.

[Minka2001] Thomas P. Minka. 2001. *A family of algorithms for approximate Bayesian inference.* Ph.D. thesis, Massachusetts Institute of Technology.

[Mishina and Yamamoto2002] Takuya Mishina and Mikio Yamamoto. 2002. Context adaptation using variational Bayesian learning for ngram models based on probabilistic LSA. *IPSJ 2002-NLC-73*, pages 13–18.

[Mishina and Yamamoto2004] Takuya Mishina and Mikio Yamamoto. 2004. Context adaptation using variational Bayesian learning for ngram models based on probabilistic LSA. *IEICE Trans. on Inf. and Sys. (Japanese edition)*, J87-D-II(7):1409–1417.

[Müller et al.2001] K. R. Müller, S. Mika, G. Ratsch, and K. Tsuda. 2001. An introduction to kernel-based learning algorithms. *IEEE Neural Networks*, 12(2):181–201.

[Neal and Hinton1998] Radford M. Neal and Geoffrey E. Hinton, 1998. *A View of the EM Algorithm that Justifies Incremental, Sparse, and other Variants*, pages 355–368. Dordrecht: Kluwer Academic Publishers.

[Nigam et al.2000] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. 2000. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2/3):103–134.

[Papadimitriou et al.1998] Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. 1998. Latent Semantic Indexing: A Probabilistic Analysis. pages 159–168.

[Rosenfeld1996] Ronald Rosenfeld. 1996. A Maximum Entropy Approach to Adaptive Statistical Language Modeling. *Computer, Speech and Language*, 10:187–228.

[Rosenfeld1997] Ronald Rosenfeld. 1997. A whole sentence maximum entropy language model. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding.*

[Sadamitsu et al.2004] Kugatsu Sadamitsu, Yuusuke Machitori, and Mikio Yamamoto. 2004. A smoothing method for parameters of Dirichlet mixtures using hierarchical Bayesian models. *IPSJ 2004-SLP-53*, pages 1–6.

[Sapir1921] Edward Sapir. 1921. *Language: An Introduction to the Study of Speech.* New York: Harcourt, Brace and company.

[Schultz and Joachims2003] Matthew Schultz and Thorsten Joachims. 2003. Learning a Distance Metric from Relative Comparisons. In *NIPS 2003.*

[Sugaya et al.2002] F. Sugaya, T. Takezawa, G. Kikui, and S. Yamamoto. 2002. Proposal for a very-large-corpus acquisition method by cell-formed registration. In *Proc. LREC-2002*, volume I, pages 326–328.

[Suzuki et al.2003] Jun Suzuki, Tsutomu Hirao, Yutaka Sasaki, and Eisaku Maeda. 2003. Hierarchical Directed Acyclic Graph Kernel: Methods for Structured Natural Language Data. In *41th Annual Meeting of Association for Computational Linguistics*, pages 32–39.

[Watanabe and Hori2003] Shinji Watanabe and Takaaki Hori. 2003. N-gram language modeling via Bayesian approach. In *Proceedings of the Acoustical Society of Japan*, number 2-6-10, pages 79–80.

[Weisstein2004] Eric W. Weisstein. 2004. Moore-Penrose Matrix Inverse. MathWorld – A Wolfram Web Resource.

[Xing et al.2002] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. 2002. Distance metric learning, with application to clustering with side-information. In *NIPS 2002*.

[Yamamoto et al.2003] Mikio Yamamoto, Kugatsu Sadamitsu, and Takuya Mishina. 2003. Context modeling using Dirichlet mixtures and its applications to language models. *IPSJ 2003-SLP-48*, pages 29–34.

[Yao1984] Yi-Chin Yao. 1984. Estimation of a noisy discrete-time step function: Bayes and empirical Bayes approaches. *Annals of Statistics*, 12:1434–1447.

[Yu et al.2005] Kai Yu, Shipeng Yu, and Volker Tresp. 2005. Dirichlet Enhanced Latent Semantic Analysis. In *AI & Statistics (AISTATS-2005)*.

List of Publications

Journal Papers

[1] 持橋大地, 菊井玄一郎, 北研二. 言語表現のベクトル空間モデルにおける最適な計量距離. **電気情報通信学会論文誌** D-II, Vol.J88, No.4, 2005. *to appear.*

International Conferences

[1] Daichi Mochihashi, Genichiro Kikui, and Kenji Kita. Learning Nonstructural Distance Metric by Minimum Cluster Distortions. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, pages 341–348, Barcelona, July 2004.

[2] Daichi Mochihashi, Genichiro Kikui, and Kenji Kita. A Globally Optimal Distance Metric to Find Synonymous Expressions. Proc. of International Workshop on MTMIR (IJCNLP-04 International Workshop on MT and Multilingual information Retrieval), Sanya City, Hainan Island, China. March, 2004.

List of Other Publications

[1] 持橋大地, 松本裕治. Particle Filter による文脈の動的ベイズ推定. **情報処理学会研究報告 自然言語処理研究会** *2005-NL-165*, pages 59–66, 2005.

[2] 持橋大地, 松本裕治. PLSA による確率的概念空間の評価. **情報処理学会研究報告 自然言語処理研究会** *2003-NL-153*, pages 41–47, 2003.

[3] 持橋大地, 松本裕治. 意味の確率的表現. **情報処理学会研究報告 自然言語処理研究会** *2002-NL-147*, pages 77–84, 2002.

[4] 持橋大地, 松本裕治. 連想としての意味. **情報処理学会研究報告 自然言語処理研究会** *1999-NL-134*, pages 155–162, 1999.

[5] 持橋大地, 加来田裕和. 個人の選好に応じた単語の重要度の学習と電子メールの重要度自動判別への応用. **情報処理学会研究報告 自然言語処理研究会** *1999-NL-129*, pages 35–39, 1999.

Appendix

A. LOO likelihood of Polya mixture

Here, we show

$$\log p(\mathbf{w}|\boldsymbol{\alpha}) = \log\left[\frac{\Gamma(\alpha)}{\Gamma(\alpha+n)}\prod_v \frac{\Gamma(\alpha_v+n_v)}{\Gamma(\alpha_v)}\right] \tag{5.1}$$

$$\simeq \sum_v n_v \log\left(\frac{n_v+\alpha_v-1}{n+\alpha-1}\right) \tag{5.2}$$

for a document $\mathbf{w}$ and Dirichlet hyperparameter $\boldsymbol{\alpha}$, where n_v is the count of occurrence of word v in $\mathbf{w}$ and $n = \sum_v n_v\,, \alpha = \sum_v \alpha_v\,.$

Making a Leave-One-Out (LOO) approximation

$$p(\mathbf{w}|\boldsymbol{\alpha}) \simeq \prod_v p(v|\mathbf{w}\backslash v, \boldsymbol{\alpha})^{n_v} \tag{5.3}$$

where

$$p(v|\mathbf{w}\backslash v, \boldsymbol{\alpha}) = \int p(v|\boldsymbol{p})p(\boldsymbol{p}|\mathbf{w}\backslash v, \boldsymbol{\alpha})d\boldsymbol{p} \tag{5.4}$$

$$= \int p_v \mathrm{Dir}(\alpha_1+n_1, \cdots, \alpha_v+n_v-1, \cdots, \alpha_V+n_V)d\boldsymbol{p} \tag{5.5}$$

$$= \frac{\alpha_v+n_v-1}{\alpha+n-1}\,, \tag{5.6}$$

expression (5.2) follows immediately. □

B. Derivation of two bounds

Here, we derive two simple lower bounds that has been used in section 3.3.1 of Dirichlet Mixture parameter estimation.

$$\log(x+n) = \log\left(q\cdot\frac{x}{q} + (1-q)\cdot\frac{n}{1-q}\right) \tag{5.7}$$

$$\geq q\log\frac{x}{q} + (1-q)\log\frac{n}{1-q} \tag{5.8}$$

$$= q\log x + (1-q)\log n - H(q), \tag{5.9}$$

where $H(q) = q\log q + (1-q)\log(1-q)$.

Equality holds at the point of contact between two curves:

$$\frac{\partial}{\partial x}\log(x+n) = \frac{\partial}{\partial x} q \log x \tag{5.10}$$

$$\therefore \quad q = \frac{x}{x+n}. \quad \square \tag{5.11}$$

By a Taylor expansion at $x = x_0$, we get

$$\log x = \log x_0 + \frac{x - x_0}{x_0} - O(x_0^2) \tag{5.12}$$

$$\leq \log x_0 + \frac{x}{x_0} - 1 = ax - 1 - \log a, \tag{5.13}$$

where $a = 1/x$ is the point of contact. $\square$

C. Moore-Penrose Matrix Pseudoinverse

The Moore-Penrose matrix pseudoinverse A^+ of A is a unique matrix that has a property of normal inverse in that $x = A^+ y$ is a shortest length least squares solution to $Ax = y$ even if A is singular (Weisstein, 2004).

A^+ can be calculated simply by a MATLAB function `pinv`. Or alternatively (Ishikawa et al., 1998), we can decompose A as

$$A = U \Sigma U^T \ , \tag{5.14}$$

where U is an orthonormal $n \times n$ matrix and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_R, 0, \ldots, 0)$ ($R = \mathrm{rank}(A)$). Then, A^+ is calculated as

$$A^+ = U \Sigma^+ U^T \ , \tag{5.15}$$

where $\Sigma^+ = \mathrm{diag}(1/\sigma_1, \ldots, 1/\sigma_R, 0, \ldots, 0)$. Therefore,

$$M = (\sigma_1 \sigma_2 \cdots \sigma_R)^{1/R} A^+. \quad \square \tag{5.16}$$