UNIVERSIDAD AUTÓNOMA DE MADRID
Escuela Politécnica Superior

Automatic aspect extraction in information retrieval diversity

DAVID ALFAYA SÁNCHEZ

DIRECTOR
Pablo Castells Azpilicueta (IRG UAM)

Trabajo de Fin de Máster presentado en la UAM para obtención del título de Máster en Investigación e Innovación en Tecnologías de la Información y las Comunicaciones en el curso académico 2014-2015

14 de Septiembre de 2015

Abstract

In this master thesis we describe a new automatic aspect extraction algorithm by incorporating relevance information to the dynamics of the Probabilistic Latent Semantic Analysis. An utility-biased likelihood statistical framework is described to formalize the incorporation of prior relevance information to the dynamics of the algorithm intrinsically. Moreover, a general abstract algorithm is presented to incorporate any arbitrary new feature variables to the analysis.

A tempering procedure is inferred for this general algorithm as an entropic regularization of the utility-biased likelihood functional and a geometric interpretation of the algorithm is described, showing the intrinsic changes in the information space of the problem produced when different sources of prior utility estimations are provided over the same data.

The general algorithm is applied to several information retrieval, recommendation and personalization tasks. Moreover, a set of post-processing aspect filters is presented. Some characteristics of the aspect distributions such as sparsity or low entropy are identified to enhance the overall diversity attained by the diversification algorithm. Proposed filters assure that the final aspect space has those properties, thus leading to better diversity levels.

An experimental setup over TREC web track 09-12 data shows that the algorithm surpasses classic pLSA as an aspect extraction tool for the search diversification.

Additional theoretical applications of the general procedure to information retrieval, recommendation and personalization tasks are given, leading to new relevance-aware models incorporating several variables to the latent semantic analysis.

Finally the problem of optimizing the aspect space size for diversification is addressed. Analytical formulas for the dependency of diversity metrics on the choice of an automatically extracted aspect space are given under a simplified generative model for the relation between system aspects and evaluation true aspects.

An experimental analysis of this dependence is performed over TREC web track data using pLSA as aspect extraction algorithm.

ACM Computing Classification System.

H.3.3 [**Search and Retrieval**]: retrieval models, relevance feedback

I.2.7 [**Natural Language Processing**]: Language models.

Key words and phrases. Intent space, aspect extraction, information retrieval, diversity, probabilistic latent semantic analysis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and problem definition

Nowadays, diversity enhancement has become a consolidated research area in both information retrieval and recommendation systems. A great variety of theoretical models and methodologies have arisen for describing, measuring and enhancing the diversity of search engines, recommendation systems and any other retrieval application.

Classic evaluation methodology for IR systems was focused on precision and relevance. Providing the user reiterative results was not penalized as far as the duplicate information was relevant. Diversity represents a complementary perspective for relevance, as it takes into account the variety of retrieved results – understood as a low redundancy between different retrieved elements – as a valuable objective for improving user access to information.

Some authors (Zhai et al., 2003) describe the transition from traditional information retrieval models to novel and diversity problems as arising from a change in perspective about relevance. The first ones consider an *independent relevance model*, where document relevance is an inherent characteristic of documents, depending only on the initial user needs. On the other hand, diversity framework proposes that information and relevance must be considered globally, depending on the whole document or item ranking presented to the user. For example, during a browsing session, the marginal of the information contained in a document for a certain user is conditioned on the amount of such information already provided to the user through previous documents of the ranking. We usually refer to this perspective as a *conditional relevance model*.

A fruitful approach for diversifying search results corresponds to representing the ambiguity of queries and documents through an abstract aspect or facet space. These spaces reflect all possible interpretations or subtasks underlying the particular choice of the query provided to the system. Some of the most popular diversity evaluating methodologies rely on the choice of a suitable aspect space or taxonomy (Zhai et al. (2003), Clarke et al. (2008), Agrawal et al. (2009), Santos et al. (2010), etc.). Moreover, once a set of suitable aspects has been fixed, there exist multiple models and algorithms which are able to use the intent information to increase the diversity of the results (Agrawal et al., 2009; Santos et al., 2010). For this reason,

aspect extraction becomes one of the main subproblems when aiming to improve the overall diversity of a system.

The present master thesis analyzes some of main strategies used in the literature to extract intent spaces which are suitable for diversification, focusing on the probabilistic ones: topic models. We analyze the theoretical background of two of the main model-driven methods for latent semantic analysis, Probabilistic Latent Semantic Analysis (Hofmann, 1999a,b, 2001) and Latent Dirichlet Allocation (Blei et al., 2003). An overview of the state of the art applications of these methods for intent space approximation is performed and a theoretical comparison between them is given.

The main research problem addressed in this work is to build automatic aspect space extraction algorithms that optimize the characteristics of the resulting spaces to make them suitable for diversification. In particular, we aim to incorporate relevance information to the system, in order to build query-specific spaces that capture the precise intent differences within each single considered query, therefore leading to extracted spaces being more informative about the diverse structures of the retrieved documents.

Finally, the general optimization problem of finding an optimal number of aspects for diversification will be approached. Some preliminary studies have been done (Vargas et al., 2012b), but an explicit analytic dependence of diversity metrics to the number of aspects of the extraction system is an open question.

1.2 Reserach goals

The main broad research objective of this master thesis is to develop new automatic aspect extraction algorithms which enhance the performance of the common diversification methods by creating more suitable and informative intent spaces.

The main particular research objectives leading to the main goal can be summarized as follows:

- State of the art study and analysis on the main-frame strategies used for treating the diversity problem in information retrieval, focused on the intent-aware methodology. An overview of classic explicit and implicit methods for aspect extraction will be described.

- Analysis of topic models theoretical framework, focusing on the structure and dynamics of pLSA and LDA. Applications of topic models to information retrieval tasks will be outlined and outstanding topic models usages as part of aspect extraction algorithms will be described.

- Development of a new generalization of pLSA involving relevance information from baseline ranking.

- Description of an abstract probabilistic framework that covers the previous model, allowing the incorporation of relevance notions to the pLSA dynamics, together with arbitrary new variable analysis. Proof of correctness, convergence, tempering variants and a geometrical interpretation will be given.

- Obtaining of instances of this generic algorithm to other information retrieval and recommendation tasks.
- Development of further aspect space optimization strategies in the form of aspect filters which guarantee some desirable properties of the final intent space, such as sparsity or low entropy.
- Computation of exact analytic formulas for the dependence of the diversification quality on the size of the aspect space.

1.3 Document structure

The rest of the document is structured in the following way. Chapters 2 and 3 correspond to a state of the art analysis about information retrieval diversity and the use of topic models to approximate user intents. Chapters 4 and 5 introduce our major contributions, namely a relevance aware pLSA generalization in the context of a general utility-biased statistical framework and some major aspect space optimization methods, in the form of a general tempered variant for the proposed algorithm, aspect filtering methods and an analytical study of aspect space size optimization. Particularly

- **Chapter 2** summarizes some of the most relevant state of the art techniques for measuring and enhancing information retrieval diversity. The notion of intent space is introduced and various aspect extraction algorithms are outlined.
- **Chapter 3** explores the concept of topic models. First of all, the basic computational statistics tools needed to describe the theoretical framework of latent models is described. Starting from these notions, two of the main probabilistic algorithms used for latent semantic analysis (pLSA and LDA) are described and analyzed. Some of the main recent applications of these methods to diversity tasks are described.
- In **Chapter 4** a new utility-biased expectation maximization algorithm is proposed as a framework for incorporating new variables and relevance to the pLSA dynamics. The Relevance aware Probabilistic Latent Semantic Analysis (RapLSA) is proposed, and various applications of the described framework are given, including search diversity, recommendation diversity, content based recommendation filtering and personalization. A tempering method is developed for the general abstract algorithm and an information-geometric interpretation of the framework is given. As an additional aspect space optimization, three families of aspect filters are described to improve the suitability of the extracted space for diversification. Finally, the proposed theoretical algorithms are tested in search and recommendation experiments.
- The optimization of the size of the aspect space for diversity enhancing is addressed in **chapter 5**. A simplified generative model for the relationship between extracted aspects and true evaluation subtopics is used to develop explicit exact formulas for the dependency of diversity quality to the size of the aspect space.

- **Chapter 6** exposes the conclusions of our work, remarks the obtained results and contributions and describes future work research lines.

Chapter 2

Information Retrieval Diversity

In this chapter we will present an outline of some of the most common methodologies for evaluating and enhancing diversity in information retrieval systems. Following our main research objectives, we will focus on intent-oriented approaches and we will overview some of the major classic aspect space extraction methods used in the literature.

2.1 Diversity Metrics

We will cover some of the most common methods used in the literature to measure the overall diversity of an information retrieval system. In order to simplify the notation and the terminology, we will describe them in the context of search diversity, but a recommendation counterpart can trivially be stated for each of them.

2.1.1 Subtopic Recall

One of the major changes in the point of view in information retrieval when focusing on the diversity problem is the notion of relevance. Usually, traditional retrieval systems work under the assumption of independent relevance, i.e., they assume that the probability of a document of being relevant is independent to the relevance of other documents in the ranking. Relevance of information is considered as an inherent property involving only a certain isolated document and the user.

Nevertheless, in the diversity framework it becomes clear that the user perceives the retrieved ranking as a whole common source of information, and that the absolute relevance of a document must be put in the context of the rest of the information contained in the rest of the retrieved documents. Zhai et al. (2003) distinguish the notion of independent relevance and dependent relevance, using query subtopics as quantized elementary information tokens that each document can cover.

Let us consider a topic Q with K subtopics $Z_1, \ldots, Z_K$. Let $D = (d_1, \ldots, d_N)$ be a ranked list of documents. For each document d_i, let $Z(d_i)$ be the set of subtopics covered by document d_i.

$$S - recall@\alpha(D) = \frac{|\bigcup_{i=1}^{\alpha} Z(d_i)|}{K}$$

This metric is problematic when trying to compare results from different topics, as the difference in the subtopic size directly influences the metric value, independently of he subtopic coverage distribution along the ranking. In order to account for the intrinsic difficulty, they propose to "invert" the recall problem and search for the minimum ranking for which a certain recall is attained. In particular, for $0 \leq r \leq 1$, let $\text{minRank}(D, r)$ be the minimal rank α at which $S-recall@\alpha(D) \geq r$, i.e.

$$\text{minRank}(D, r) = \min\{\alpha | S - recall@\alpha(D) \geq r\}$$

Subtopic precision (S-precision) at recall r is defined to be the collection-based normalization of the inverse of minRank, i.e.,

$$S - precision@r = \frac{\text{minRank}(D_{opt}, r)}{\text{minRank}(D, r)}$$

where D_{opt} is the optimal rank, i.e., the permutation of rank D that attains the minimum $\text{minRank}(\cdot, r)$.

2.1.2 Intent-Aware metrics

Classic information retrieval metrics, like precision, nDCG or ERR, measure the information provided by a document and its relevance as a whole, not taking into account the possibility of a document being really relevant for certain aspect of the query and not for the other. The latter is the usual scenario in ambiguous or diverse queries, where the sets of documents covering each of the sub-topics may be disjoint. Take, for example, the query "Java". Two main facets of the query are easily tracked, "Java" as a programming language and "Java" as an island. It is probable that retrieved documents belong to exactly one of those classes. Now let us suppose that we try to evaluate the relevance of an "island" document. In general, programing results are more likely than the island ones. Therefore global relevance of an "island" document would be low, while its relevance restricted to the "island" facet may be really high.

If we want to measure the diversity of a system showing results of both kinds of documents we can't use absolute notions of relevance alone, as that would lead to promoting non-diverse rankings where the predominant facet has been taken as the unique one. As a solution, Agrawal et al. (2009) propose the use of Intent-Aware variants of the classic common metrics.

Let us suppose that a certain taxonomy Z for the retrieved documents and the query itself is available. We will suppose that documents and queries can belong to more than one category and, in particular, we will assume that a distribution $p(z|q)$ measuring the probability of a given ambiguous query belonging to given categories. Moreover, let us assume that we have category-dependent relevance judgments for each ranked document, i.e., let us suppose that apart from the ad-hoc relevance information $r(d)$, we can determine the relevance of each document d restricted to class z, $r(d|z)$.

Intent-Aware metrics are obtained from classic metrics by taking the average of the metric when restricted to a certain aspect, pondered by the probability of the

query belonging to that aspect, i.e., given the metric $M(D,q)$, the corresponding IA-metric would be

$$M - IA(D,q) = \sum_{z \in Z} p(z|q) M(D,q|z)$$

where loosely, $M(D,q|z)$ corresponds to evaluating the metric M using conditioned relevance information $r(d|z)$ instead of the ad-hoc one $r(d)$. The exact meaning of this change is particular to each metric, but all the common ones share the same basic idea. As an example, we present intent aware versions for mean reciprocal rank (MRR) and average precision (MAP).

Classic MRR corresponds to the average among queries of the inverse of the first relevant element. Let r_i be the position of the first relevant document (for a certain sense of relevance binarization of r). Then

$$RR(D,q) = \frac{1}{r_1}$$

In the intent-aware scenario, taking $r_i(z)$ to be the position of the i-th relevant document for class z yields

$$RR - IA(D,q) = sum_{z \in Z} \frac{1}{r_1(z)}$$

On the other hand, the average precision of a ranked resultset is

$$AP = \frac{\sum_{j=1}^{N} r(j) \frac{\sum_{i=1}^{j} r(i)}{j}}{\sum_{j=1}^{N} r(j)}$$

The analogous intent-aware version results

$$AP - IA = \sum_{z \in Z} \frac{\sum_{j=1}^{N} r(j|z) \frac{\sum_{i=1}^{j} r(i|z)}{j}}{\sum_{j=1}^{N} r(j|z)}$$

As an example for a cascade browsing model metric, we will give the explicit equation for the intent aware expected reciprocal rank. Let us suppose now that $0 \leq r(i|z) \leq 1$ denotes the normalized relevance of document, interpreted as the probability of the document in position i being relevant for aspect z.

$$ERR - IA(D,q) = \sum_{z \in Z} p(z|q) \sum_{j=1}^{N} \frac{1}{j} r(i|z) \prod_{i=1}^{j-1} (1 - r(i|z)) \tag{2.1.1}$$

Finally, we will consider the intent-aware version of the Normalized Discount Cumulative Gain (nDCG). Discount Cumulative Gain is computed as the expected cumulative utility obtained by a user by browsing through the ranking. Explicitly,

$$DCG(D,q) = \sum_{j=1}^{N} \frac{r(d_j)}{\log_2(j+1)}$$

We can give the previous formula two interpretations up to a normalization constant. Either we estimate the probability of users reaching position j as proportional to $\frac{1}{\log_2(j+1)}$ and we model the utility of each of the visited documents as proportional to their relevance $r(d)$ or we assume that the probability that the user sees a certain document d_j is proportional to its independent relevance $r(d_j)$ and the marginal utility of visiting a document in the j-th position is proportional to $\frac{1}{\log_2(j+1)}$. It becomes clear that DCG is not a normalized metric. Instead, a model-normalized version is used, denoted as Normalized DCG (nDCG).

$$nDCG(D,q) = \frac{DCG(D,q)}{\max_{R\in\sigma(D)} DCG(R,q)}$$

where $\sigma(D)$ denotes the set re-rankings (permutations) of D. The intractability of an exact computation of the normalization constant results in a greedy algorithm being used to approximate the optimal ranking.

Following the general methodology, Agrawal et al. (2009) propose the following intent-aware version of nDCG. For each $z \in Z$, we consider the aspect-conditioned DCG for the ranking as the value of the usual DCG metric computed using the conditioned relevance information $r(d|z)$ instead of $r(d)$, i.e.

$$DCG(R,q|z) = \sum_{j=1}^{N} \frac{r(d_j|z)}{\log_2(j+1)}$$

Then, $nDCG - IA$ is described as

$$nDCG - IA = \sum_{z\in Z} p(z|q) \frac{DCG(D,q|z)}{\max_{R\in\sigma(D)} DCG(R,q|z)}$$

where each optimal re-ranking for each aspect is computed using a greedy algorithm. We notice that the previous metric would not be normalized even if the exact optimal rankings were used, as the existence of a common optimal rank for all subtopics would be impossible in general.

2.1.3 α-nDCG

α-nDCG corresponds to a novelty and diversity measuring analog of nDCG different from IA-nDCG proposed by Clarke et al. (2008) in the framework of a methodology for treating explicit subtopic relevance judgments for evaluating novelty and diversity.

Let Z denote a set of subtopics or "information nuggets" representing the diverse facets of a query. We will think both of the user and the documents as sets of those nuggets. Documents provide information about certain subtopics and users seek for a certain set of information tokens through the query. We will write $p(z \in d)$ and $p(z \in u)$ to denote the probability of aspect z belonging to a document or a user respectively. On the other hand, let us denote by $p(R|u,d)$ the probability of user u finding document d relevant. We estimate that the probability of a document being relevant to the user corresponds to the probability of a certain subtopic z

being present both in the document and the user. Assuming that all distributions $p(z \in d)$ and $p(z \in u)$ are independent to each other, we obtain

$$p(R|u,d) = 1 - \prod_{z \in Z} (1 - p(z \in d)p(z \in u))$$

We estimate the probability of a certain subtopic belonging to a document using explicit relevance judgments. Let us suppose that for every $d \in D$ and every $z \in Z$, a human assessor determines whether the topic is covered by the document or not. Let $J(d,z) = 1$ if the assessor thinks that the aspect is in the document and $J(d,z) = 0$ otherwise. We will estimate the probability $p(z \in d)$ as

$$p(z \in d) = \begin{cases} \alpha & \text{if } J(d,z) = 1 \\ 0 & \text{if } J(d,z) = 0 \end{cases}$$

where, $0 < \alpha \leq 1$ is a parameter modeling possible human errors. On the other hand, as we don't assume any additional knowledge of user preferences, we will take distributions $p(z \in u)$ to be independent and identically distributed, taking

$$p(z \in u) = \gamma$$

for some $\gamma \in (0, 1]$.

Then the following browsing model is assumed. The probability of a user u reaching document j in the ranking is the probability of user u finding document d_j still relevant after seeing documents d_1 to d_{j-1}, i.e., it corresponds to the probability of existence of an aspect $z \in u$ covered by d_j but not covered by any of $d_1, \ldots, d_{j-1}$. We get

$$p(z \in u | z \notin d_1, \ldots, z \notin d_{j-1}) = p(z \in u) \prod_{i=1}^{j-1} p(z \notin d_i)$$

Letting $r_j(z) = \sum_{i=1}^{j} J(d_i, z)$, we get

$$p(z \in u | z \notin d_1, \ldots, z \notin d_{j-1}) = \gamma(1 - \alpha)^{r_{j-1}(z)}$$

Therefore, we estimate

$$p(R_j|u, d_1, \ldots, d_{j-1}) = 1 - \prod_{z \in Z} \left(1 - \alpha J(d_j, z)\gamma(1 - \alpha)^{r_{j-1}(z)}\right)$$
$$\approx \gamma\alpha \sum_{z \in Z} J(d_j, z)(1 - \alpha)^{r_{j-1}(z)}$$

Using $\frac{1}{\log_2(j+1)}$ as the marginal utility for the information in position j, we obtain

$$\alpha - DCG(D,q) = \sum_{j=1}^{N} \frac{\sum_{z \in Z} J(d_j, z)(1 - \alpha)^{r_{j-1}(z)}}{\log_2(j+1)}$$

Alike nDCG, the previous quantity is not normalized, so a model-driven normalization is used

$$\alpha - nDCG(D,q) = \frac{\alpha - DCG(D,q)}{\max_{R \in \sigma(D)} \alpha - DCG(R,q)}$$

where, as usual, the optimal ranking is computed using a greedy algorithm.

2.2 Diversification Algorithms

Once the diversity problem has been stated and a ground evaluation methodology has been described, the next step is to describe algorithms improving the overall diversity of the system. In this section we will explore one family of such algorithms, the so called re-ranking methods.

These kind of algorithms take an initial undiversified ranking R and information about the subtopic structure of the retrieved documents and reposition the same retrieved elements to build a new diversified list S with maximum diversity. The explicit notion of the overall diversity of the resulting list S vary depending on the algorithm model and the used parameters, and it will be described precisely in each case.

2.2.1 IA-Select

In the same article introducing intent aware metrics, Agrawal et al. (2009) propose a diversification algorithm that uses prior subtopic-wise relevance of the retrieved documents to re-rank the baseline result maximizing the expected subtopic coverage.

As with IA-metrics, let us suppose that a given taxonomy Z of documents and queries is provided. Moreover, let us assume that we can compute an abstract relevance functional $V(d|q,z) \in [0,1]$ measuring the quality value of a document d for query q given the class z. A probabilistic interpretation of the functional is possible, taking it as the likelihood of the document d satisfying the user intent z for the query q. Nevertheless, only the following independence assumption will be made as for the choice of $V(d|q,z)$. If two documents d_1 and d_2 are retrieved for the same query q and belong to the same aspect z, the probability of none of them being relevant to the user for the intent z is

$$(1 - V(d_i|q,z))(1 - V(d_j|q,z)) \tag{2.2.1}$$

The diversification problem is then described as finding the re-rank S maximizing the probability of the average user finding at least one useful result within the top N results. In particular, given the original ranking R, the taxonomy Z and the quality functional V, IA-select outputs the list $S \subseteq R$ of M elements maximizing

$$p(S|q) = \sum_{z \in Z} p(z|q) \left(1 - \prod_{d \in S} (1 - V(d|q,z))\right) \tag{2.2.2}$$

The previous equation can be easily derived from the abstract problem using the probabilistic interpretation of functional V. Given an user intent $z \in Z$, the probability of the user finding at least one relevant document within the set S corresponds to one minus the probability of all documents not being relevant for query q and aspect z. Independence condition (2.2.1), implies that the latter is computed as the product of $1 - V(d|q,z)$ for all $d \in S$. Therefore, the probability of the user finding useful information about a certain topic z is given by

$$p(S|q,z) = 1 - \prod_{d \in S} (1 - V(d|q,z))$$

Taking the expected value over all possible query intentions Z leads to (2.2.2). The global optimization problem over S described by IA-Select is intractable (NP-complete). Agrawal et Al. propose the following greedy approximation. Let $U(z|q, \neg S)$ denote the probability of query q belonging to class $z \in Z$ given that none of the documents $d \in S$ satisfy the user for aspect z. This corresponds to

$$U(z|q, \neg S) = p(z|q) \prod_{d \in S} (1 - V(d|q, z))$$

The following greedy approximation is then computed

Algorithm 2.1 IA-Select algorithm

procedure IA-SELECT(q, R, Z, M, $V(d|q, z)$, $p(z|q)$)
  $S = \emptyset$
  for $z \in Z_q$ **do**
    $U(z|q, S) = p(c|q)$
  end for
  while $|S| < M$ **do**
    Select document d^* as

$$d^* = \underset{d \in R}{\operatorname{argmax}} \left(\sum_{z \in Z} U(z|q, S) V(d|q, z) \right)$$

    $S = S \cup \{d^*\}$
    for $z \in Z_{d^*}$ **do**
      $U(z|q, S) = (1 - V(d^*|q, z)) U(z|q, S \backslash \{d^*\}$
    end for
    $R = R \backslash \{d^*\}$
  end while
  return S
end procedure

where Z_q and Z_d correspond respectively to the subsets of classes which queries and documents belong to.

2.2.2 xQuAD

Santos et al. (2010) describe the xQuAD (eXplicit Query Aspect Diversification) algorithm following global optimization problem: "given an initial ranking R for query q, find the re-ranking S that has the maximum coverage and the minimum redundancy with respect to the different aspects underlying q".

The objective is engaged by a greedy approximation. The diversified ranking S is successively built as taking the document $d \notin S$ maximizing the mixture model

$$(1 - \lambda) p(d|q) + \lambda p\left(d, \neg S|q\right)$$

where $p(d|q)$ is a measure of ground relevance, corresponding to the likelihood of document d being observed by the user as relevant for query q, and $p\left(d, \neg S|q\right)$ is an

estimation of the likelihood of user finding document d relevant but not any other document already in S, which corresponds to the diversity estimation.

Algorithm 2.2 xQuAD algorithm

procedure xQuAD(q, R, λ, M)
  $S = \emptyset$
  while $|S| < M$ **do**
    Select document d^* as

$$d^* = \underset{d \in R}{\operatorname{argmax}} \left((1-\lambda) p(d|q) + \lambda p\left(d|\neg S, q\right) \right)$$

    $R = R \backslash \{d^*\}$
    $S = S \cup \{d^*\}$
  end while
  return S
end procedure

Given a set of sub-queries $\{q_1, \ldots, q_K\}$ for query Q, they estimate the probability $p(d, \neg S|q)$ by estimating the covering/redundancy over each sub-query.

$$p\left(d, \neg S|q\right) = \sum_{i=1}^{K} p(q_i|q) p\left(d, \neg S|q_i\right)$$

Santos et Al. use sub-queries for estimating user intentions with a similar sense than Zhai or Agrawall subtopics. As an approximation for sub-queries they use either query reformulations, document clusters or the usual query expansion techniques like Rocchio method. Document relevance for each sub-query is measured as a distribution $p(d|q_i)$ and the diversity term conditioned by the sub-query is given by

$$p\left(d, \neg S|q_i\right) = p(d|q_i) p\left(\neg S|q_i\right)$$

where $p(\neg S|q_i)$ denotes the probability of document set S not containing any relevant document for sub-query q_i. The latter is estimated assuming that relevance of a document $d_j \in S$ to a given sub-query q_i is independent of the relevance of any other document $d_l \in S$ to q_i. Therefore

$$p\left(\neg S|q_i\right) = p\left(\left(\neg d_1\right) \wedge \ldots \wedge \left(\neg d_{|S|}\right) \middle| \, q_i\right) = \prod_{i=1}^{|S|} p\left(\neg d_j|q_i\right) = \prod_{j=1}^{|S|} (1 - p(d_j|q_i))$$

Substituting in the general equation yields

$$p\left(d, \neg S|q\right) = (1-\lambda) p(d|q) + \lambda \sum_{i=1}^{K} p(q_i|q) p(d|q_i) \prod_{j=1}^{|S|} (1 - p(d_j|q_i))$$

If we compare its global objective with the one from IA-Select, we observe that IA-Select is just focused on aspect coverage and has the redundancy avoiding as a byproduct of demoting documents which have the already selected aspects. In

contrast, xQuAD aims to explicitly avoid redundancy, even after coverage has been achieved. In terms of re-ranking, this difference gets reflected in the way IA-Select and xQuAD behave after the documents covering the totality of the considered subtopics have been placed. In this situation, IA-Select essentially stops diversifying and retains the relative position of the subsequent elements. On the other hand, xQuAD continues to reorder the remaining documents, selecting those being less redundant with respect to the already selected ones.

Moreover, xQuAD λ parameter allows us to balance the trade-off between relevance and pure diversity. For $\lambda = 0$, the diversity score is neglected and documents are ranked from their baseline relevance probabilities $p(d|q)$. On the other hand, for $\lambda = 1$ yields a pure diversification, i which document relevance is not considered and final positions are only dependent on each document aspect distribution.

2.3 Aspect extraction

All the previously described strategies can be considered to be examples of explicit or intent-aware diversity approaches. Both metrics and algorithms are based on a choice of a certain classification space Z, capturing the different intentions that a user may had when posing the query q, i.e., the possible latent information needs underlying the literal expression of the query.

In the literature, the terms intent, aspect, subtopic or facet are usually used in the context if diversity tasks to reflect the different interpretations or intentions that a single explicit information token may have. They arise as a conceptual tool to fill the information gap between what a user really needs and how those needs are translated to a certain expression within the system interface.

For example, in search problems, a single query may be subject to different interpretations or reflect many retrieval subtasks that the user intends to do fulfill by that single query. In recommendation, where user profile traces acts as an implicit query, a single user may have different separate item tastes. For instance, a single user profile in a movie recommendation system may reflect completely different behavior when consuming films from different genres.

In general, we will denote intent space to the choice of any space or source of information whose elements represent each of the possible aspects of the information tokens treated by the system. The way of building this kind of abstract spaces is not obvious, as they attempt to capture unexpressed user needs. Several approaches have been taken in the literature. We will give an overview of some of the most relevant ones dividing them in two groups: methodologies relying on the use of external sources of information and approaches building implicit aspect spaces from the already known data. For additional information, an extensive comparison of the different diversity strategies and aspect building methodologies can be found in Santos et al. (2012).

2.3.1 External data approach

Explicit aspect spaces can be found if additional information is available about users behavior. In order to build proper aspect spaces it is not necessary to know

the explicit profile of each single user. It suffices to obtain a way of estimating the "possible" interest that the set of considered users may have.

The most simple example of an external source for intent information is given by TREC (Text REtrieval Conference) diversity task qrels (Clarke et al., 2010). Along years 2009 to 2012, an explicit sub-query structure was provided for an overall set of 200 queries and a set of subtopic-specific relevance judgments was provided.

Agrawal et al. (2009) use the Open Directory Project (ODP) to classify documents. ODP (www.dmoz.org) is a collaborative project for building a human-edited directory of the Web. Nowadays, more than 4 million sites have been classified in approximately one million stratified categories by a set of more than 90.000 editors. Agrawal et Al. use the first 15 top categories of the taxonomy to classify the content. Document relevance estimations and intents for queries are then obtained using the Amazon Mechanical Truk platform (www.mturk.com).

Santos et al. (2010), Radlinski and Dumais (2006) and Capannini et al. (2011) use query reformulations as a subtopic approximation. Sub-queries are found either by the use of query logs, analyzing patterns of query reformulations and extrapolating them to new observed data, or by traditional query expansion methods like Roccio. Santos et al. (2010) indicate additional sources of sub-query generations, such as the use of document clusters.

Additionally from ODP, Rafiei et al. (2010) use Wikipedia disambiguation pages to build an explicit intent space for a search experiment. They selected 50 of such pages from Wikipedia and used the ambiguous titles as queries, which were proposed to the system being evaluated. The results were led to human evaluators who decided which of the Wikipedia subtopics was the most suitable for each of the pages and they evaluated the system based on the global S-recall@α. A similar approach for subtopic generation was used by Welch et al. (2011).

2.3.2 Implicit aspect space building

Automatic aspect extraction algorithms represent a great challenge in diversity enhancement and evaluation. These algorithms aim to build a set of abstract aspects approximating user intents implicitly from the observed data. In contrast to the previously described methods, no additional sources of information are used to express user intentions.

Given a vectorial representation of data, matrix factorization algorithms have been used in the literature as a mean to compress the information into a low-dimensional summarizing space. Deerwester et al. (1990) propose the use of this technique in the context of text indexing to build a vector space that approximates latent semantic information underlying the literal content of documents. The method, called indexing by Latent Semantic Analysis, leads to spaces at the semantic level of information and, therefore, are candidates for building intent spaces. This procedure has been used in recommender systems diversity (Vargas et al., 2011) to build implicit user intent spaces suitable for diversification.

Finally, probabilistic topic models have been recently used to build aspect spaces. The use of methods like pLSA (Hofmann, 1999b,a, 2001) or LDA (Blei et al., 2003) provide useful tools to approximate the latent semantic ideas underlying a certain query or document term expression. Next chapter will be specifically devoted to

studying this kind of algorithms in depth.

Moreover, our main research objectives, exposed in chapters 4 and 5, focus on developing and optimizing new variants of these model-driven implicit aspect extraction algorithms.

Chapter 3

Topic Models

As we have seen in the last chapter, intent spaces are a powerful tool that allows us to represent and approximate the different real user needs of information that lie behind the expression of those needs provided to the system.

Nevertheless, whereas the abstract concepts of "aspect" or "intent space" provide a great theoretical contraption for building intent aware systems and enhancing the diversity of a search or recommendation engine, there exist a clear practical problem when it comes to build explicit intent spaces if a concrete set of features is not provided.

In the last part of the chapter we covered some methods used in the literature for automatic aspect extraction. Although some of the methods have been proved to be really effective in terms of the final diversity of the system, none of them is model driven. As a consequence, the resulting intent spaces can't be directly incorporated to the system model. Instead, at some point an heuristic approach must be made in most of the cases.

Probabilistic approaches have been proved to be among the most effective ones in most information retrieval tasks and have become the dominant paradigm. When describing a retrieval methodology, the ideal situation would be to prove that, given some fixed assumed hypothesis about the problem, a retrieval algorithm based on a retrieval model would be the most effective one among all approaches arising from the same hypothesis (Croft et al., 2010). While this kind of optimality proofs are usually intractable in the considered tasks, due to the complexity of formalizing human behavior, probabilistic methods provide a theoretical framework for controlling the inherent uncertainty of the problem.

In the context of language processing, the term "topic model" stands for a statistical latent variable language model describing a generative framework for abstract latent semantic topics that occur in a collection of documents (Blei, 2012). Topic models provide abstract low-dimensional spaces for text representation, such that documents are represented in "orthogonal" and independent semantic dimensions (Zhai, 2009). The low dimensionality and the orthogonality of the representations imply two desirable characteristics for semantic analysis:

- Similar words having a common meaning tend to be summarized in single abstract token

- A single word can be represented as a mixture of several topics, representing possible context depending meanings

While the first property is mainly desirable for text indexing systems, the second one allows us to retrieve semantic intent from texts an documents. This will allow us to use topic models to build intent spaces which can be used to measure and enhance the diversity of a set of retrieved documents.

In this chapter we will describe probabilistic methods to automatically build such intent spaces from a document corpus. In contrast to the previously described algorithms, these model topic techniques have a strong statistical theoretical basis. First of all, we will review some basic statistics notions on latent variable models, maximum likelihood estimators and some optimization techniques needed to solve the kind of parameter estimation problems that we will encounter later on. In particular, we will focus on the Expectation-Maximization (EM) algorithm and some of its variants. Then we will present two important model driven EM-based aspect extraction algorithms, Probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA). Both algorithms aim to surpass the lexical level of the documents language and extract the latent semantics behind it, building a space of latent semantic classes in the process. We will explore how these spaces are built and their application to diversity problems.

3.1 Previous concepts on maximum-likelihood estimators and convex optimization algorithms

Before introducing topic models themselves and the corresponding topic extractions algorithms, we need to present some of the main mathematical tools needed to develop them.

3.1.1 Latent Variables, parametric models and Maximum Likelihood estimators

Missing data arise in many applications of statistical analysis. Given a probability model, we can split the random variables between the set of observed and unobserved variables. In the literature, the term latent variable is usually reserved for unobserved variables that, even if they are not directly measured, can be inferred from the observed data.

We denote by latent variable models those in which the observed variables are modeled as completely defined and independent given a set of latent variables.

Thus, in a latent variable model we can distinguish between what we call the incomplete data model, or observed data model, that corresponds to the part of the model and the samples corresponding to the observed variables and the complete data model, which also involves the unknown value of the latent variables.

For example, quality of life, understood as the general well-being of individuals belonging to a certain social group, can be thought as a latent variable. While it can be estimated using quantitative measures like Human Development Index, quality of life is usually considered to be unmeasurable. Instead, evidences of high quality of life are observed from some related random variables. Examples of these

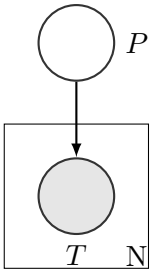
dependent include the proper Human Development Index, Social Progress Index, life expectancy, per-capita Gross domestic product, etc. An estimation of the overall quality of life can obtained as a combination of these observed variables.

Latent variable models usually depend on a set of parameters modeling the relation between the observed and unobserved variables, such as the ones defining the conditional distribution of the observed variables given the unobserved ones or the priori distribution of the latent variables. In this case, we say that the model is "parametric". Formally, a parametric model is a family of distributions modeling the complete data parametrized by a set of (usually real) parameters.

Both latent variables and parameters represent unknown information in the model. Nevertheless, they are not the same kind of statistical object. Latent variables are actual random variables, with its associated probability space of which we don't have (or we can't have) any sample. On the other hand, parameters are fixed (unknown) constants parametrizing the model.

As an example of parametric model, suppose that we toss a charged coin. The number of heads and tails after a certain number N of tosses can be modeled as a random variable which clearly depends on how much the coin was charged. As different tosses are independent we can just model the hole experiment as different samples from the following graphical model

Figure 3.1: Graphical model for charged coin toss



Where variables T_i stand for the result of the i-th toss of the coin. And variable P is the parameter representing the charge of the coin, in therms of the probability of getting heads. The complete data model is fully described by the graphical model and the conditional distribution

$$p(T = \text{“heads”}|P = p) = p$$

Given a parametric model like 3.1, we would like to infer some information about the parameters given the observed data. In particular, we aim to find the values of the parameters that maximize the probability of observing the given data. Those values are called the maximum-likelihood estimators for the parameters.

More precisely, suppose that we have a set X of observed variables modeled by a vector Θ of parameters. Let $x_1, \ldots, x_n$ be the samples of the variables in X that form the incomplete data.

Definition 3.1.1. *We define the likelihood of the incomplete data given a value $\Theta = \theta$ of the parameters to be the value of the functional*

$$\mathcal{L}(\theta) := p(X = x_1, \cdots, X = x_n|\Theta = \theta)$$

Usually, different samples from the same model are assumed to be independent. In that case, the likelihood is given by

$$\mathcal{L}(\theta) = \prod_{i=1}^{N} p(X = x_i | \Theta = \theta)$$

This product is not computationally well-behaved, so we will usually take the logarithm of this functional as a measure of the likelihood of the data

$$\log(\mathcal{L}(\theta)) = \sum_{i=1}^{N} \log(p(X = x_i | \Theta = \theta)) \tag{3.1.1}$$

We call this functional the log-likelihood of the data.

Definition 3.1.2. *The maximum-likelihood estimator (MLE) for the parameters Θ is given by*

$$MLE(\Theta) = \underset{\theta}{\operatorname{argmax}}(\mathcal{L}(\theta))$$

As the logarithm is strictly increasing, the MLE could be equivalently defined in terms of the log-likelihood.

In the coin toss example, we could compute the MLE for the charge of the coin. Suppose that we observe an list of results $t_1, \ldots, t_N$. For each $i = 1, \ldots, N$, we know that

$$P(T = t_i | P = p) = \begin{cases} p & t_i = \text{“heads”} \\ 1 - p & t_i = \text{“tails’} \end{cases}$$

Thus, we have that

$$\log(\mathcal{L}(p)) = \sum_{i=1}^{N} p(T = t_i | P = p) = (\#heads)\log(p) + (\#tails)\log(1 - p)$$

In order to obtain the MLE, we shall just compute the derivative of the functional and equal it to zero. Thus

$$0 = \frac{\partial \log(\mathcal{L}(p))}{\partial p} = \frac{\#heads}{p} - \frac{\#tails}{1 - p}$$

Solving for p we obtain the already expected result

$$p = \frac{\#heads}{\#heads + \#tails} = \frac{\#heads}{N}$$

Whereas this method seems easy to apply and clearly provides a way to uniquely define the MLE for the data, the last step may not be as straightforward as the one shown in the example. Depending on the form of the conditional distributions and the structure of the model, the resulting equations may lead to a strongly nonlinear system of equations with no explicit analytical solution. In this (rather typical) scenario, some optimization theory is needed in conjunction with the probability techniques in order to find the desired MLE.

3.1.2 Majorize-Minimization algorithm

The majorize-minimization algorithm is an optimization algorithm that allows us to obtain local minima of a convex function. It was developed by Ortega and Rheinboldt (1970) in the context of line search methods and since then it has been applied to multiple problems, such as multidimensional scaling (De Leeuw and Heiser, 1977), robust regression (Huber et al., 1981; Huber, 2004), quadratic lower bound principle (Böhning and Lindsay, 1988), medical imaging (De Pierro, 1995; Lange and Fessler, 1995), variable selection (Hunter and Li, 2005), discriminant analysis (Wu and Lange, 2010) and others (Hunter et al., 2000; Hunter and Lange, 2002; Sabati and Lange, 2002; Hunter et al., 2004).

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function, i.e., such that for each $x, y \in \mathbb{R}^n$ and every $t \in (0, 1)$, we have

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

Definition 3.1.3. *A function $g : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is said to majorize the function f at $y \in \mathbb{R}^n$ if*

$$f(y) = g(y, y) \tag{3.1.2}$$

$$f(x) \leq g(x, y) \quad \textit{for all } x \in \mathbb{R}^n \tag{3.1.3}$$

Dually, g is said to minorize f at y if $-g$ majorizes $-f$ at y.

The algorithm proceeds as follows.

Algorithm 3.1 Majorize-Minimization algorithm

1: **procedure** MM-ALGORITHM(f)
2: Start with a random point $x_0 \in \mathbb{R}^n$
3: **for** each n **do**
4: Select a function g_n that majorizes f at x_n
5: Select x_{n+1} as
$$x_{n+1} = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}}(g_n(x, x_n))$$
6: **end for**
7: The algorithm stops either after a maximum number of iterations or when $\|x_{n+1} - x_n\|$ is less than a certain threshold
8: **end procedure**

A dual version of this algorithm can be built in order to maximize f, just by applying the previous algorithm to $-f$. In this situation, the algorithm is called minorize-maximization algorith, because at each step instead of picking g_n that majorizes f, we pick g_n that minorizes f at x_n and choose x_{n+1} to be the arg-max of g_n instead of its minimum. Clearly, in order to apply this version, we usually ask for f to be concave instead of convex. In the literature, both the minorize-maximization and the majorize-minimization algorithms are usually denoted by MM-algorithm, and it is usually clear from the context which version are we applying.

Even without any assumption on f or g, the following lemma proves that the algorithm monotonically decreases the value of f.

Lemma 3.1.4. *Let $\{x_n\}$ be a set of points selected by Algorithm 3.1 for the function f. Then, for each n,*

$$f(x_{n+1}) \leq f(x_n)$$

Proof. For each n, g_n majorizes f at x_n, so by equation (3.1.2),

$$g_n(x_n, x_n) = f(x_n)$$

On the other hand, (3.1.3) implies that

$$f(x_{n+1}) \leq g_n(x_{n+1}, x_n)$$

Finally, by definition of x_{n+1} as the minimum of $g_n(\cdot, x_n)$, we have

$$g_n(x_{n+1}, x_n) \leq g_n(x_n, x_n)$$

□

It is worth noting that in this form, the algorithm is more like a meta-algorithm. For a generic f there is no guaranty of the existence of non-trivial minorizing functions g_n and even if they exist, the convergence to a local minimum of f is not guaranteed unless we suppose some additional hypothesis about f and g.

Also note that for lemma 3.1.4 to work we don't need x_{n+1} to be the minimum of $g_n(\cdot, x_n)$, but we only need x_{n+1} to be a point such that $g_n(x_{n+1}, x_n) \leq g_n(x_n, x_n)$. This leads us to a generalize version of the algorithm, called GMM algorithm in which in each step, we just take x_{n+1} to be any point that decreases the value of $g_n(\cdot, x_n)$. This may be useful in case that the minimum of the majorizing function is intractable, but approximable with certain precision. In this case, depending on the structure of both f and g, on most cases, the convergence properties of the GMM algorithm can be proved to be the same as that of the MM algorithm (Neal and Hinton, 1998).

(Dempster et al., 1977) proved that if f is convex, the algorithm converges to a local minimum.

Now we will review some basic and useful means of building the majorizing functions given that f has a certain form, for further reference, see Hunter et al. (2004) and Zhou and Lange (2010).

Suppose that

$$f(x_1, \ldots, x_n) = \phi\left(\sum_{i=1}^{n} x_i\right)$$

Where $\phi : \mathbb{R} \to \mathbb{R}$ is a convex function (and, therefore, f is convex). Then, applying Jensen inequality to function ϕ proves that for every positive $y_1, \ldots, y_n$

$$\phi\left(\sum_{i=1}^{n} x_i\right) \leq \sum_{i=1}^{n} \frac{y_i}{\sum_{j=1}^{n} y_j} \phi\left(\frac{\sum_{j=1}^{n} y_j}{y_i} x_i\right) \tag{3.1.4}$$

Taking $g(x, y)$ to be the right hand side of the inequality, we have that for every $x \in \mathbb{R}^n$, $f(x) \leq g(x, y)$ and clearly

$$g(y,y) = \sum_{i=1}^{n} \frac{y_i}{\sum_{j=1}^{n} y_j} \phi\left(\frac{\sum_{j=1}^{n} y_j}{y_i} y_i\right) = \sum_{i=1}^{n} \frac{y_i}{\sum_{j=1}^{n} y_j} \phi\left(\sum_{j=1}^{n} y_j\right) = f(y)$$

Thus, g majorizes f at every point $y = (y_1, \ldots, y_n)$ where $y_i > 0$ for $i = 1, \ldots, n$.

A second kind of majorization can be made in case f is concave. In this case, for every $y \in \mathbb{R}^n$, the supporting hyperplane of f at y lies over the graph of f and, by definition, it's tangent at y, so it gives us a majorizing function

$$g(x,y) = f(y) + \nabla f(y) \cdot (x - y)$$

If f is twice differentiable and has bounded curvature, Böhning and Lindsay (1988) proved that we can further improve the previous bound by considering the second order Taylor polynomial of f at a point y and majorizing its quadratic term. Explicitly, let H_f be the Hessian of f. If if we can find a possitive definite matrix M such that $M - H_f(x)$ is nonnegative definite for all x, then

$$f(x) \leq g(x,y) := f(y) + \nabla f(y) \cdot (x - y) + \frac{1}{2}(x - y)^t M (x - y)$$

And clearly, $f(y) = g(y,y)$, so $g(x,y)$ majorizes f at every $y \in \mathbb{R}^n$. Finally, suppose that $f(x) = \sum_{i=1}^{n} f_i(x)$ and that for each $i = 1, \ldots, n$ we have function g_i majorizing f_i at a fixed common point y. Then it is straightforward to see that $g(x,t) = \sum_{i=1}^{n} g_i(x,t)$ majorizes $f(x)$ at y, because as a consequence of g_i majorizing f at y

$$f(y) = \sum_{i=1}^{n} f_i(y) = \sum_{i=1}^{n} g_i(y,y) = g(y,y)$$

$$f(x) = \sum_{i=1}^{n} f_i(x) \leq \sum_{i=1}^{n} g_i(x,y) = g(x,y) \qquad \text{for all } x$$

3.1.3 Expectation-Maximization algorithm

Suppose that we have a latent variable model with observed variables X with samples $\{x_1, \ldots, x_n\}$ and unobserved variables Z parametrized by a vector of unknown parameters θ. In order to simplify the computations, in this section we will assume that considered random variables are categorical, but a continuous version can be equivalently derived (Dempster et al., 1977).

The objective of the Expectation-Maximization algorithm (from now on, EM algorithm) is to obtain a maximum likelihood estimator for the parameters θ given the observed data X. Marginalizing, the likelihood of the observed data is given by

$$\log(\mathcal{L}(\theta)) = \sum_{i=1}^{n} \log(p(x_i|\theta)) = \sum_{i=1}^{n} \log\left(\sum_{z \in Z} p(x_i, z|\theta)\right)$$

Trying to compute directly the minimum of the previous expression with respect to θ generally leads to heavily nonlinear equations on the parameters, due to the functional involving the logarithm of a sum of probabilities depending on θ.

The EM algorithm uses the MM algorithm to solve this problem. In order to build the minorizing function, we will use a version of equation (3.1.4). As $\log(\mathcal{L}(\theta))$ is a sum of functionals for every sample x_i of the observed variables, it's only necessarily to build a minorizing function for every sample x_i and sum them. Then, fixed a parameter vector θ_t

$$f_i(\theta) = \log\left(\sum_{z\in Z} p(x_i, z|\theta)\right) = \log\left(\sum_{z\in Z} p(z|x_i,\theta_t)\frac{p(x_i,z|\theta)}{p(z|x_i,\theta_t)}\right)$$

The logarithm is concave. As $p(z|x_i,\theta_t)$ are supposed to be positive (otherwise, restrict the sum) and $\sum_{z\in Z} p(z|x_i,\theta_t) = 1$, Jensen inequality implies that for any parameter vector θ

$$f_i(\theta) \geq \sum_{z\in Z} p(z|x_i,\theta_t)\log\left(\frac{p(x_i,z|\theta)}{p(z|x_i,\theta_t)}\right) = \sum_{z\in Z} p(z|x_i,\theta_t)\log(p(x_i,z|\theta)) -$$

$$\sum_{z\in Z} p(z|x_i,\theta_t)\log(p(z|x_i,\theta_t)) =: g_i(\theta,\theta_t)$$

with equality when $p(z|x_i,\theta_t) = p(z|x_i,\theta)$. In particular, $f_i(\theta_t) = g_i(\theta_t,\theta_t)$, so g_i minorizes the desired functional f_i.

The previous form of functional g_i can be expressed in a more representative way in terms of the expected value of the complete data likelihood and a certain entropy

$$g_i(\theta,\theta_t) = \mathbb{E}_{Z|x_i,\theta_t}[\log(p(x_i,Z|\theta))] + H(p(z|x_i,\theta_t)) \tag{3.1.5}$$

Following the MM-algorithm, we can find local maxima of the complete functional by successively maximizing $g(\theta,\theta_t) = \sum_{i=1}^n g_i(\theta,\theta_t)$. As each entropy term is constant, maximizing g with respect to θ is equivalent to maximizing the functional

$$Q(\theta,\theta_t) = \sum_{i=1}^n \mathbb{E}_{Z|x_i,\theta_t}[\log(p(x_i,Z|\theta))]$$

Thus, the MM-algorithm has the following explicit form

Algorithm 3.2 Expectation-Maximization algorithm

1: **procedure** EM-ALGORITHM($p(X,Z|\theta)$,$\{x_i\}$)
2: Start with a random vector parameter $\theta_0 \in \mathbb{R}^k$
3: **for** each t **do**
4: Step E: Compute $Q(\theta,\theta_t) = \sum_{i=1}^n \mathbb{E}_{Z|x_i,\theta_t}[\log(p(x_i,Z|\theta))]$
5: Step M: Select θ_{t+1} as

$$\theta_{t+1} = \underset{\theta\in\mathbb{R}^k}{\operatorname{argmax}}(Q(\theta,\theta_t))$$

6: **end for**
7: The algorithm stops either after a maximum number of iterations or when $\|\theta_{t+1} - \theta_t\|$ is less than a certain threshold
8: **end procedure**

Thus, the algorithm alternates between two steps, an "expectation" or "E" step in which the conditional expectation given by functional Q is computed given the previous value of the parameters θ_t and a "maximization" or "M" step, in which the maximum of such functional is found and set as the new value for the parameter, θ_{t+1}.

It is important to notice that the complete explicit form of the functional Q doesn't need to be computed during the E step but only the parameters depending on θ_t needed in order to find the maximum in step M later on. In particular, if both the observed and latent variables are discrete, it is usually enough to compute the conditional distributions $p(z|x_i, \theta_t)$ by Bayes rule.

Neal and Hinton (1998) formalize the EM algorithm in an equivalent way more related to this form of "computing" the E step. They prove that the E and M steps can be seen as both maximizing a common functional with respect to two different variables.

For each sample of the observe data x_i, let $q_i(z)$ be a distribution over the latent variables Z. For notation simplicity denote by $q = (q_1, \ldots, q_n)$. Consider the following functional over the q_i and the parameter vector Q

$$F(q, \theta) = \sum_{i=1}^{n} \left(\mathbb{E}_{q_i}[\log p(x_i, z|\theta)] + H(q_i) \right) \tag{3.1.6}$$

Applying Bayes rule, we have that $\log(p(x_i, z|\theta) = \log(p(z|x_i, \theta)) + \log(p(x_i|\theta))$. By linearity of the expected value and taking into account that the second term is constant in z, we have that for each i

$$\begin{aligned} \mathbb{E}_{q_i}[\log p(x_i, z|\theta)] + H(q_i) = \sum_{z \in Z} q_i(z) \log(p(z|x_i, \theta)) - \sum_{z \in Z} q_i(z) \log(q_i(z)) + \\ \log(p(x_i|\theta)) = -\operatorname{KLD}(q_i \| p(z|x_i, \theta)) + \log(p(x_i|\theta)) \end{aligned} \tag{3.1.7}$$

Summing in the samples we have

$$F(q, \theta) = -\sum_{i=1}^{n} \operatorname{KLD}(q_i \| p(z|x_i, \theta)) + \log(\mathcal{L}(\theta)) \tag{3.1.8}$$

First, notice that maximizing F with respect to q is the same as minimizing $\operatorname{KLD}(q_i \| p(z|x_i, \theta)$ with respect to q_i for each i. By Gibb's inequality, this quantity is non-negative, reaching zero if and only if $q_i = p(z|x_i, \theta)$. Thus, $F(\cdot, \theta)$ has a single minimum at $q_i = p(z|x_i, \theta)$. On the other hand, equation (3.1.6) implies that for $q_i = p(z|x_i, \theta_t)$

$$F(q, \theta) = g(\theta, \theta_t)$$

Therefore, given that $q_i = p(z|x_i, \theta)$

$$\operatorname*{argmax}_{\theta} F(q, \theta) = \operatorname*{argmax}_{\theta} g(\theta, \theta_t) = \operatorname*{argmax}_{\theta} Q(\theta, \theta_t)$$

This proves that the EM algorithm is equivalent to the following

Algorithm 3.3 Alternate form of the Expectation-Maximization algorithm

1: **procedure** EM-ALGORITHM($p(X, Z|\theta)$,$\{x_i\}$)
2: Start with a random vector parameter $\theta_0 \in \mathbb{R}^k$
3: **for** each t **do**
4: Step E: Select $q^{(t)} = \left(q_1^{(t)}(z), \cdots, q_n^{(t)}(z)\right)$ as

$$q^{(t)} = \underset{q}{\operatorname{argmax}}(F(q, \theta_t))$$

5: Step M: Select θ_{t+1} as

$$\theta_{t+1} = \underset{\theta \in \mathbb{R}^k}{\operatorname{argmax}}(F(q^{(t)}, \theta))$$

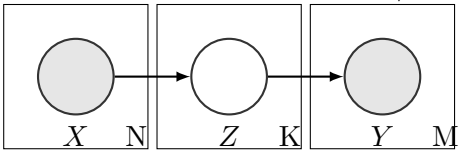
6: **end for**
7: The algorithm stops either after a maximum number of iterations or when $\|\theta_{t+1} - \theta_t\|$ is less than a certain threshold
8: **end procedure**

As the θ parameters usually describe the conditional and priori distributions $p(X|z, \theta)$ and $p(z|\theta)$, this version of the algorithm shows a more symmetric aspect of EM. E and M steps are both successively computing more precise approximations of the conditional distributions $p(z|X, \theta)$ and $p(X|z, \theta)$ respectively. This can be interpreted as the algorithm first estimating a distribution of the unobserved variables given the observed date with a preliminary guess of the parameters $p(z|X, \theta)$, and then using this distribution to compute the opposite conditional distribution $p(X|z, \theta)$ and refine the previous approximation of the parameters θ.

3.1.4 Helmholz functionals and tempering

One of the main applications of the EM algorithm comes from statistical learning. For example, suppose that we have a classification problem with input variables X and output variables Y modeled through a set of hidden variables Z with the following common generative model

Figure 3.2: Graphical model for classification/regression problem



Taking either the categorical distributions $p(Z_i|X)$ and $p(Y_i|Z)$ in the discrete scenario or its distribution parameters in the continuous counterpart as parameters of the model, hidden variables Z as unobserved variables and pairs of input-output samples (x_i, y_i) as observed data, the EM-algorithm gives a locally optimal estimation for the parameters of the model that maximize likelihood of the observed samples, thus obtaining a set of parameters that locally optimally fits the training

data.

This kind of approach has been proved to be really effective in training feedforward neural networks (Ma et al., 1977), Boltzmann machines (Byrne and Member, 1992) and multiclass classification tasks (Ng and McLachlan, 2004) improving both convergence rate and performance.

Alike any other learning algorithm, one of the main concerns while applying the EM-algorithm to statistical learning tasks is overfitting the training data. In order avoid it, annealing techniques can be incorporated to the classical EM algorithm. The deterministic annealing was developed by Rose et al. (1990) in his thesis (Rose, 1991) as a way to escape local optima while applying traditional clustering methods, weakening the dependency of the algorithm on the initial configuration in the process. The approach, motivated by the concepts of free energy and entropy in statistical mechanics was later on adapted by Ueda and Nakano (1998) as a modification of the EM algorithm in the frame of Generalized EM (GEM) methods.

GEM algorithms share the same philosophy of the EM algorithm. We find surrogate minorizing functions for the likelihood functional and then maximize the surrogates in order to increase the value of the total likelihood. The difference between GEM and EM is that, during the M step, instead of updating the parameters to an absolute maximum of the minimizing function, we just update to a point that increases the value of the functional. Obviously the convergence of these GEM algorithms depend heavily on how the updated parameters are chosen in the modified M step. A really successful strategy is to perturb the original surrogate functional through adding secondary minor terms that change the local dynamics of the algorithm without changing its global convergence rate.

Ueda and Nakano (1998) use this approach. In their work, Neal and Hinton (1998) notice that the functional F in equation (3.1.6) is analogous to the "variational free energy" of statistical physics, taking the values of the hidden variables Z for each sample x_i to be the states of the system and considering $-\log(p(z, x_i|\theta))$ as the "energy" of the state. With this interpretation, the free energy functional, corresponding to a statistical version of the Helmholtz free energy functional, for each sample x_i, set of parameters θ and state distribution $q_i(z)$ is the sum of the expected energy $\mathbb{E}_q[-\log(p(z, x_i|\theta))]$ minus the entropy of the system $H(q)$, which coincides precisely with the opposite of the functional F. With this physical interpretation, the EM algorithm successively decreases the total variational free energy of the system by alternately optimizing it with respect to the state distribution and the model parameters . Therefore, it corresponds to a grouped version of the coordinate ascent algorithm for the Helmholtz functional.

Ueda and Nakano introduce an "inverse computational temperature" in the Helmholtz functional. Instead of F, at each step the following functional is maximized given some $\beta \geq 0$

$$F_\beta(q, \theta) = \sum_{i=1}^{n} (\beta \mathbb{E}_{q_i}[\log p(x_i, z|\theta)] + H(q_i)) \tag{3.1.9}$$

The main motivation for the change is the "principle of entropy maximization" of statistical mechanics that specifies that among all probability distributions within the same energy level, the one with maximum entropy naturally arises. Jaynes

(1957) stated this principle in an article establishing the strong links existing between statistical mechanics and information theory. In the information theory scenario, the entropy of the distribution $p(Z|x_i, \theta)$ is a measure of the amount of information carried by the model. Thus, the distribution with maximal entropy corresponds to the most general one satisfying the constraints, in the sense that it is the one that models the most uncertain outcome of the random variable if we only know the given constraints. As an example, if no other condition on $p(Z|x_i, \theta)$ holds, the principle of entropy maximization would state that the most natural and general distribution to model the latent variable would be the uniform one.

In the clustering scenario, Rose (1991) and, posteriorly, Ueda and Nakano (1998) in the general context of EM algorithm both state that the latent distribution (cluster distribution, in the case of Rose's work)

Uneda and Nakano notice that the latent distribution $p(Z|x_i, \theta_t)$ being computed as the posterior depending on the parameter selection θ_t makes the EM algorithm depend excessively on the parameter estimation, and, in particular, on the original random starting estimation θ_0. If the parameter θ_t is far from optimal at one step, then $p(Z|x_i, \theta_t)$, computed directly as the posterior, can be far from the real distribution. Instead, they propose that the distribution should be computed from the parameters θ_t using the principle of entropy maximization, i.e., it should be taken as the probability distribution q_i with the maximum entropy within the ones with the same energy level $E_i = \mathbb{E}_{q_i}[-\log(p(x_i, z|\theta_t))]$.

Considering simultaneously all samples x_i, we are led to maximizing the total entropy $H(q) = \sum_{i=1}^{n} H(q_i)$ subject to the total energy being fixed, i.e.

$$\sum_{i=1}^{n} \mathbb{E}_{q_i}[-\log(p(x_i, z|\theta_t))] = E$$

The corresponding optimization problem with constraints can be solved using Lagrange multipliers. Letting β be the Lagrange multiplier for the restriction on E, we have to maximize

$$L(q, \beta) = \sum_{i=1}^{n} H(q_i) - \beta \left(\sum_{i=1}^{n} \mathbb{E}_{q_i}[-\log(p(x_i, z|\theta_t))] - E \right) \tag{3.1.10}$$

with respect to q and β. It is clear that differentiating with respect to $q_i(z)$ yields a parametric equation defining completely the distribution $q_i(z)$ depending on β but not on E. This fixes the structure of q_i up to the constant β, which is then determined by the total energy equation. Thus, we can reparamtrize the solution in terms of β. This allows us to take $E = 0$ in functional (3.1.10) and consider β not as a Lagrange multiplier anymore, but as a parameter playing the same roll E did. Thus, q is the maximum of functional

$$L(q, \beta)' = \sum_{i=1}^{n} H(q_i) - \beta \left(\sum_{i=1}^{n} \mathbb{E}_{q_i}[-\log(p(x_i, z|\theta_t))] \right) = F_\beta(q, \theta)$$

Once fixed the β parameter, the E and M steps now operate just as in EM version 3.3. It's worth saying that once the distributions q_i are computed, the M step takes exactly the same form as the traditional EM algorithm. Indeed, for $\beta = 1$

both algorithms are equivalent. The second important part of the algorithm is the choice of the β parameter. The following meta-algorithm is used

Algorithm 3.4 Tempered Expectation-Maximization algorithm

procedure TEM-ALGORITHM($p(X, Z|\theta)$,$\{x_i\}$,β_{min})
    Start with a random vector parameter $\theta_0 \in \mathbb{R}^k$
    Set $\beta = \beta_{min}$
    while $\beta \leq 1$ **do**
        for each t **do**
            Step E: Select $q^{(t)} = \left(q_1^{(t)}(z), \cdots, q_n^{(t)}s(z)\right)$ as

$$q^{(t)} = \underset{q}{\operatorname{argmax}}(F_\beta(q, \theta_t))$$

            Step M: Select θ_{t+1} as

$$\theta_{t+1} = \underset{\theta \in \mathbb{R}^k}{\operatorname{argmax}}(F_\beta(q^{(t)}, \theta))$$

        end for
        The loop stops either after a maximum number of iterations or when $\|\theta_{t+1} - \theta_t\|$ is less than a certain threshold
        Increase β
    end while
end procedure

Parameter β is increased so that the last β value is close to $\beta = 1$. This way, the las iteration is equivalent to an instance of EM for which the initial parameter θ_0 is chosen as the output parameter θ_t of the previous TEM iteration. Therefore, upon convergence, the final parameter θ_t would have converged to a local maximum of the likelihood functional $\mathcal{L}(\theta)$ given the observed data.

The idea is to start with high computational temperature, i.e. $0 < \beta_{min} << 1$. This makes $q^(0)$ become almost uniform and $F_\beta(q^(0), \theta)$ has a single global maximum to which the algorithm converges. By gradually increasing β, we are progressively perturbing the dynamics, weakening the starting global maximum and introducing the local maxima existing in the original EM functional F. Inside each step of the outer β loop, the last selected parameters, which converged to a local maximum for the old value of β, are now presumably out of a local maximum region, but near one (if the β change is little enough) to which the algorithms converges by the end of that β epoch.

While, as EM, this algorithm is not guaranteed to converge to a global maximum, the tempered version has been shown to effectively attain better local maxima. The basis behind this is that, if β grows slowly enough, the first local maxima of F to become relevant in the dynamics of F_β are the highest ones, thus making algorithm unconcerned of weak local maxima for low values of β. Therefore, unlike traditional EM, the global dynamics of TEM makes the algorithm able to scape easily from "lower" maxima and reach better local extrema.

While this may be a great strategy in fitting problems, reaching a higher maxima

than EM may make TEM overfit even more than expected with the classic algorithm. On the other side, we have proved that each iteration of the TEM algorithm for a fixed β leads to more entropic, thus general, distributions over $\mathcal{Z}$. Taking both things into account, Hofmann (1999a,b, 2001) proposes to use an "inverse" annealing. Instead of starting with a low β and increasing it, Hofmann proposes to start with an usual EM ($\beta = 1$) and then exponentially decrease the value of β, updating $\beta \leftarrow \eta\beta$. For each β, an early stop condition is implemented. TEM iterations stop when performance deteriorates on held-out data. Performance on unseen data is expected to increase on each β update. We stop to update β when this does not yield further improvements.

3.2 Probabilistic Latent Semantic Analysis

In information retrieval tasks, many algorithms are based on direct word/item matching strategies in order to rank relevance of documents or items with respect to a query. The possible ambivalence of the words used both in the documents and the query and, in general, the inherent lack of precision of natural language make these algorithms get stuck in the explicit literal expression of information tokens and incapable of abstracting in a precise way the abstract information behind them.

Latent Semantic Analysis (Deerwester et al., 1990) is an indexing method that seeks to solve this problem by introducing the concept of a latent semantic space. The elements of the latent semantic space represent the abstract information tokens behind a certain natural language expression. The basic assumption behind LSA is that words with similar meaning (also in terms of intention) tend to appear in similar pieces of text. LSA aims to capture each of the possible different meanings, represented as states in the latent space, and build a map between the words in a document and their corresponding latent semantic meaning. This means that, abstractly, synonyms would map to a single element in the latent space, while polysemic words would split in several different states in the latent space.

In practice, the LSA algorithm builds the latent space by applying a dimensionality reduction method to the document-term mappin. In particular, the latent space is build as a vector space of a prescribed dimension. Then Singular Value Decomposition (SVD) is applied to the document-term frequency matrix, thus finding the most suitable factorization of the document - term map through the latent space. The factors can be then interpreted as document-aspect and aspect-word maps and thus can be used to represent any document or phrase in the latent space.

Retrieval algorithms are then supposed to use the states of the latent semantic space instead of the term frequency vectors as a representation of a document of query. This strategy has been proved to successfully detect synonyms, treat polysemy and reduce noise in the samples, thus becoming a robust analysis tool with many applications (Deerwester et al., 1990; Foltz and Dumais, 1992; Dumais, 1995; Landauer and Dutnais, 1997; Bellegarda, 1998). Nevertheless, the classic LDA method lacks a probabilistic theoretical basis, which contrasts with the strong formalism of some of the most effective retrieval algorithms that are used in conjunction with the method. Indeed the use of SVD instead of other kind of factorization corresponds to a choice of a certain (euclidean) metric in the matrix space. This choice,

while natural and effective, is rather heuristic and independent of the data, and it does not link with any generative model of the retrieval problem.

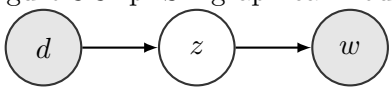
Probabilistic Latent Semantic Analysis (Hofmann, 1999a,b, 2001) is a probabilistic reformulation of the paradigm of the LDA method that allows us to build the latent space and the corresponding maps based on a generative model for the documents. Let $\mathcal{D} = \{d_1, \ldots, d_D\}$ be the corpus of documents and $\mathcal{W} = \{w_1, \ldots, w_W\}$ the complete vocabulary of the corpus. A "bag of words"-type document model is used. Thus, our observed data will consist on the pairs (w, d) such that the word w is observed in document d. For each $d \in \mathcal{D}$, and each $w \in \mathcal{W}$, we observe a pair (w, d) for each time that w appears in d. Let us denote by $n(w, d)$ the number of times that the word w appears in document d. The model is based suppose that there exist a set $\mathcal{Z} = \{z_1, \ldots, x_K\}$

An "aspect model" is used to describe the data. It is a latent variable model that assumes that for each observed pair (d, w) there exist an unobserved class variable $z \in \mathcal{Z} = \{x_1, \ldots, z_K\}$ modeling the underlying semantics and intention behind the instance of word d in the document d. The generative model for the complete data would be the following

- Select a document $d \in \mathcal{D}$ with probability $p(d)$
- Pick a latent semantic class $z \in \mathcal{Z}$ with probability $p(z|d)$
- Generate a word $w \in \mathcal{W}$ with probability $p(w|z)$

It is therefore assumed the word distribution is independent of the document once the latent semantic class is selected. This agrees with the LDA paradigm of our notion of the semantic occurrences of different words in a document are independent given Which corresponds to this graphical model

Figure 3.3: pLSA graphical model



This generative model induces a probability distribution over $\mathcal{W} \times \mathcal{D}$, given by

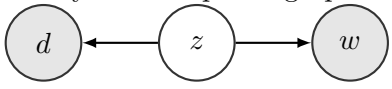
$$p(w, d) = p(d)p(w|d) = \sum_{z \in Z} p(w|z)p(z|d)p(d)$$

Applying Bayes, we get $p(z|d)p(d) = p(d|z)p(z)$, so we can refactor the previous equation in a symmetric form

$$p(w, d) = \sum_{z \in \mathcal{Z}} p(w|z)p(d|z)p(z) \tag{3.2.1}$$

Which corresponds to the following graphical model

Figure 3.4: Symmetric pLSA graphical model



In both cases, pLSA is presented as a parametric latent variable model, with observed variables w and d, latent variable z and parameters given by the categorical distributions $p(w|z)$, $p(z|d)$, $p(d)$ for the model 3.3 or categorical distributions $p(w|z)$, $p(d|z)$, $p(z)$ for the model 3.4. Both formulations are indeed equivalent parameterizations of the model, but the symmetry of the second one will simplify some computations, so we will use it instead.

The distributions are then selected as those maximizing the likelihood of the observed samples (w, d), i.e., the functional

$$\log(\mathcal{L}(p(w|z), p(d|z), p(z))) = \sum_{w\in\mathcal{W}, d\in\mathcal{D}} n(w,d) \log p(w,d) = \sum_{w\in\mathcal{W}, d\in\mathcal{D}} n(w,d) \log \sum_{z\in\mathcal{Z}} p(w|z)p(d|z)p(z) \quad (3.2.2)$$

As finding the maximum of (3.2.2) is intractable, EM algorithm is applied to the model instead. Using the original version (pseudo-code 3.2), the E step simply consist on computing the conditional distribution $p(z|w, d)$ through Bayes theorem

$$p(z|w,d) = \frac{p(w|z)p(d|z)p(z)}{\sum_{z\in\mathcal{Z}} p(w|z)p(d|z)p(z)}$$

On the other hand, the M step consists in maximizing the following functional with respect to the three distributions

$$Q(p(w|z), p(d|z), p(z)) = \sum_{d\in D} \sum_{w\in W} \sum_{z\in Z} n(w,d)p(z|w,d) \log(p(w|z)p(d|z)p(z))$$

subject to the restrictions arising from the parameters being, in fact, distributions, i.e., all being nonnegative and summing to one for each $z \in \mathcal{Z}$. The problem can be solved using Lagrange multipliers, checking afterwards that the obtained maxima correspond to nonnegative parameters. Thus, we have to maximize

$$\mathcal{F} = \sum_{d\in D} \sum_{w\in W} \sum_{z\in Z} n(w,d)p(z|w,d) \log(p(w|z)p(d|z)p(z)) - \sum_{z\in\mathcal{Z}} \lambda_z \left(\sum_{w\in\mathcal{W}} p(w|z) - 1 \right) - \sum_{z\in\mathcal{Z}} \mu_z \left(\sum_{d\in\mathcal{D}} p(d|z) - 1 \right) - \lambda \left(\sum_{z\in\mathcal{Z}} p(z) - 1 \right) \quad (3.2.3)$$

Setting the partial derivatives with respect to each parameter to zero yields

$$0 = \frac{\partial \mathcal{F}}{\partial p(\overline{w}|\overline{z})} = \sum_{d\in\mathcal{D}} n(\overline{w}, d)p(\overline{z}|\overline{w}, d) \frac{1}{p(\overline{w}|\overline{z})} - \lambda_z$$

$$0 = \frac{\partial \mathcal{F}}{\partial p(\overline{d}|\overline{z})} = \sum_{w\in\mathcal{W}} n(w, \overline{d})p(\overline{z}|w, \overline{d}) \frac{1}{p(\overline{d}|\overline{z})} - \mu_z$$

$$0 = \frac{\partial \mathcal{F}}{\partial p(\overline{z})} = \sum_{d\in\mathcal{D}} \sum_{w\in\mathcal{W}} n(w,d)p(\overline{z}|w,d) \frac{1}{p(\overline{z})} - \lambda$$

Solving each equation for the corresponding parameter yields

$$p(\overline{w}|\overline{z}) = \frac{1}{\lambda_z} \sum_{d \in \mathcal{D}} n(\overline{w}, d) p(\overline{z}|\overline{w}, d)$$

$$p(\overline{d}|\overline{z}) = \frac{1}{\mu_z} \sum_{w \in \mathcal{W}} n(w, \overline{d}) p(\overline{z}|w, \overline{d})$$

$$p(\overline{z}) = \frac{1}{\lambda} \sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} n(w, d) p(\overline{z}|w, d)$$

Summing first equation over $\overline{w} \in \mathcal{W}$ and taking into account that $\sum_{\overline{w} \in \mathcal{W}} p(\overline{w}|z) = 1$ we get that

$$\lambda_z = \sum_{w \in \mathcal{W}} \sum_{d \in \mathcal{D}} n(w, d) p(z|w, d)$$

Similarly, we get $\mu_z = \lambda_z$ and

$$\lambda = \sum_{z \in Z} \sum_{w \in W} \sum_{z \in \mathcal{D}} n(w, d) p(z|w, d)$$

Thus, $\lambda_z, \mu_z, \lambda$ are just normalization parameters for the already computed distributions. Substituting the values and simplifying the notation yields the following explicit M step for the algorithm

$$p(w|z) = \frac{\sum_{d \in \mathcal{D}} n(w, d) p(z|w, d)}{\sum_{w' \in \mathcal{W}} \sum_{d \in \mathcal{D}} n(w', d) p(z|w', d)}$$

$$p(d|z) = \frac{\sum_{w \in \mathcal{W}} n(w, d) p(z|w, d)}{\sum_{w \in \mathcal{W}} \sum_{d' \in \mathcal{D}} n(w, d') p(z|w, d')}$$

$$p(z) = \frac{\sum_{w \in \mathcal{W}} \sum_{d \in \mathcal{W}} n(w, d) p(z|w, d)}{\sum_{z' \in \mathcal{Z}} \sum_{w \in \mathcal{W}} \sum_{d \in \mathcal{D}} n(w, d) p(z|w, d)}$$

Finally, Hofmann (1999a,b, 2001) proposes the use of Tempered EM instead of EM as a way of increasing the perplexity. In this case, the M step is the same, but the equations for the E step are obtained by maximizing the functional (3.1.9), which takes the following explicit form

$$F_\beta = \beta \sum_{w \in \mathcal{W}} \sum_{d \in \mathcal{D}} \sum_{z \in \mathcal{Z}} n(w, d) p(z|w, d) \log(p(d|z) p(w|z) p(z)) - \sum_{w \in \mathcal{W}} \sum_{d \in \mathcal{D}} \sum_{z \in \mathcal{Z}} n(w, d) p(z|w, d) \log(p(z|w, d)) \quad (3.2.4)$$

With respect to $p(z|w, d)$ subject to the restrictions $\sum_{z \in \mathcal{Z}} p(z|w, d) = 1$ and $p(z|w, d) \geq 0$. Again, applying Lagrange multipliers, we have to maximize the functional

$$F'_\beta = F_\beta - \sum_{w \in \mathcal{W}} \sum_{d \in \mathcal{D}} \lambda_{wd} \left(\sum_{z \in \mathcal{Z}} p(z|w, d) - 1 \right)$$

Setting the partial derivatives to zero yields

$$0 = \frac{\partial F'_\beta}{\partial p(z|w,d)} = \beta n(w,d) \log(p(d|z)p(w|z)p(z)) - n(w,d) \left(\log(p(z|w,d)+1)\right) - \lambda_{wd}$$

Solving for $p(z|w,d)$

$$p(z|w,d) = (p(d|z)p(w|z)p(z))^\beta \exp\left(-\frac{\lambda_{wd}}{n(w,d)} - 1\right)$$

As $n(w,d)$ and λ_{wd} are constant in z, $p(z|w,d) \propto (p(d|z)p(w|z)p(z))^\beta$. The distribution has to be normalized, so we get the following tempered E step

$$p(z|w,d) = \frac{(p(d|z)p(w|z)p(z))^\beta}{\sum_{z\in\mathcal{Z}} (p(d|z)p(w|z)p(z))^\beta}$$

Then, we apply the "inverse deterministic annealing" described in the previous section, starting with $\beta = 1$ and exponentially decreasing its value as long as this improves the performance.

3.3 Latent Dirichlet Allocation

Following the probabilistic framework introduced by pLSA, LDA provides a generative latent semantic model for a text corpus based on the use of latent topics. Blei et al. (2003) notice that while the generative model provided by pLSA establishes a useful probabilistic modeling for text data, it does not present a generative model for the mixing proportions for the latent factors. In the pLSA model, any document d withing the corpus is essentially described by its probability mixing distribution $p(z|d)$. The algorithm effectively computes these distributions for the documents in the corpus, but does not provide a generative model for these numbers. Moreover, samples (w,d) are treated as observations from independent and identically distributed random variables over $\mathcal{W}\times\mathcal{D}$ with distribution $p(w,d) = \sum_{z\in\mathcal{Z}} p(w|z)p(d|z)p(z)$. This leads to some problems:

1. The length, and thus content, of a single document is not fully described. The model only predicts proportions $\frac{n(w,d)}{|d|} \sim p(w|d)$, but not the explicit number of occurrences $n(w,d)$.

2. The number of parameters $\{p(w|z), p(d|z), p(z)\}$ grows linearly with the number of documents, D. This increases both the convergence time and complexity of the model for big datasets, as well as the chances of overfitting.

3. Parameters $\{p(d|z)\}$ are only computed directly for documents in the training corpus. While Hofmann (1999b,a) describes a "fold in" strategy to infer the mixing distribution for an unobserved document, the procedure is not completely model driven. The lack of generative model makes it impossible to use a maximum-likelihood approach for the new distributions, leaving Bayesian inference from the already computed distributions as the closest, non-statistically-strong solution.

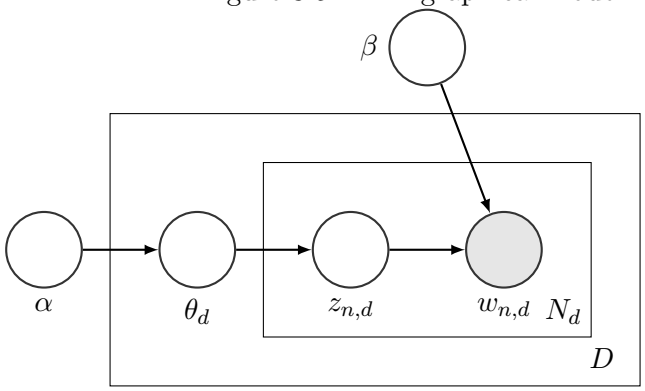
As a mean to solve this, LDA describes a generative process for both length of documents and mixing proportions in the following way:

For each document $d \in \mathcal{D}$

- Choose the length of the document $N_d \sim \text{Poisson}(\xi)$
- Choose a vector of mixing proportions $\theta_d = (\theta_{z,d})_{z \in \mathcal{Z}} \sim \text{Dir}(\alpha)$
- For each of the N_d words in document d, $w_{n,d}$
  - Pick a latent semantic class $z_{n,d} \in \mathcal{Z}$ from distribution $p(z|\theta_d) = \theta_{z,d}$
  - Generate a word $w_{n,d} \in \mathcal{W}$ with probability $p(w|z_{n,d}, \beta) = \beta_{w,z_{n,d}}$

A document d will be identified by the array of words $d \rightsquigarrow (w_{1,d}, \ldots, w_{N_d,d})$. Thus, contrary to pLSA, LDA models a set of arrays of words instead of a probability distribution over $\mathcal{W} \times \mathcal{D}$. In order to do so, three kind of parameters are to be estimated. $\xi > 0$ corresponds to the expected mean length of a document in the corpus. The length of each document is observed directly from the data and, once observed, the rest of the generative model doesn't depend on ξ. Thus, it can be estimated independently before starting the proper semantic analysis. For this reason, we will consider both the length and parameter ξ as constants from this point on. Parameter matrix $\beta = (\beta_{w,z})_{w \in \mathcal{W}, z \in \mathcal{Z}}$ parametrizes the categorical distribution $p(w|z) = \beta_{w,z}$ in the same way that it was done in pLSA. The important difference in the parametric model is the transformation of the family of parameters $p(z|d)$ into a random variable θ_d modeled as a Dirichlet distribution with parameter α. α is constant within the whole corpus $\mathcal{D}$ and, therefore, mixing proportions for every document θ_d are identically distributed, not only for documents in the corpus (for which this model is intended and adjusted) but also for any new document, allowing the "out of the training" inference that we were seeking. Globally, the graphical model for the corpus $\mathcal{D}$ is

Figure 3.5: LDA graphical model



Taking all this into account, the complete data distribution for a document $d = (w_{1,d}, \ldots, w_{N_d,d})$ is given by

$$p(z_d, \theta_d, d|\alpha, \beta) = p(\theta_d|\alpha) \prod_{n=1}^{N_d} p(z_n|\theta_d) p(w_n|z_n, \beta)$$

where

$$p(\theta|\alpha) = \frac{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\prod_{i=1}^{K}\theta_i^{\alpha_i-1}$$

Marginalizing over all possible outcomes of the latent variables θ and $z = (z_1, \ldots, z_{N_d})$ for each document, we get the following posterior for each document

$$p(d|\alpha,\beta) = \int p(\theta|\alpha)\prod_{n=1}^{N_d}\left(\sum_{z_n\in\mathcal{Z}} p(z_n|\theta)p(w_{n,d}|z_n,\beta)\right)d\theta$$

Substituting the value of the corresponding distribution yields

$$p(d|\alpha,\beta) = \frac{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\int\prod_{i=1}^{K}\theta_i^{\alpha_i-1}\prod_{n=1}^{N_d}\left(\sum_{z\in\mathcal{Z}}\theta_{z,d}\beta_{w_{n,d},z}\right)d\theta$$

Variables β and θ are heavily coupled, making the exact computation of the posterior intractable for general data. This also makes it impossible to perform the necessary computations needed to infer the exact conditional distribution of latent variables

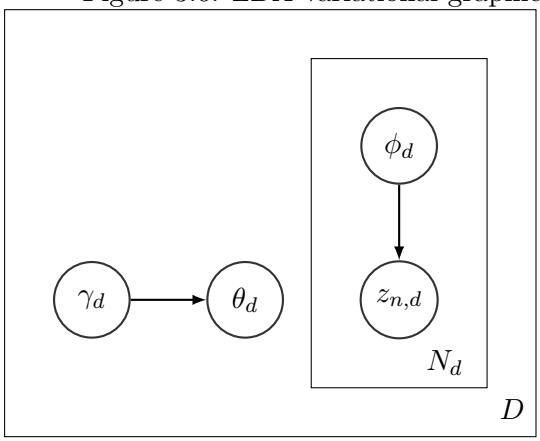
$$p(\theta, z|d,\alpha,\beta) = \frac{p(\theta,z,d|\alpha,\beta)}{p(d|\alpha,\beta)}$$

Our objective is to obtain a maximum likelihood estimation of the parameters α and β which define the mixing proportions distributions from the model. A priori, EM algorithm would fit our needs perfectly, but E step explicitly computes the conditional distribution $p(\theta, z|d,\alpha,\beta)$ for each document and the previous estimation of the parameters α, β. Thus, direct application of EM algorithm is impossible. Instead, approximation methods are needed in order to estimate these conditional distribution and being able to infer latent space distributions through the model once the training has ended.

Blei et al. (2003) propose the use of a variational method. Coupled parameters α and β come from the dependences between θ, Z and W in the graphical model. If these dependences didn't exist, the computation of E step of the EM algorithm would be straightforward. The suggested method consists precisely in substituting the exact distribution $p(\theta, z|d,\alpha,\beta)$ by a family of distributions parametrized through a decoupled version of the graphical model. For inference purposes, the new parameters are selected as those minimizing the divergence of the original distribution with respect to the variational one.

Explicitly, dependency between $\mathcal{Z}$ and θ is separated through a set of independent variational parameters $\phi_{n,d}$ and factor-term relations are suppressed from the model. Moreover, θ_d distributions from different documents, which depend on a single corpus parameter α, are decoupled, depending on a new set of document-specific parameters γ_d. The resulting Bayesian network is

Figure 3.6: LDA variational graphical model



Therefore, for each document d, the estimated family of distributions yield

$$p(\theta, z|\gamma_d, \phi_d) = p(\theta|\gamma_d) \prod_{n=1}^{N_d} p(z_n|\theta_{n,d})$$

The variational parameters are fixed as

$$(\gamma_d^*, \phi_d^*) = \underset{(\gamma_d, \phi_d)}{\operatorname{argmin}} \operatorname{KLD}\left(p(\theta, z|\gamma_d, \phi_d) \| p(\theta, z|d, \alpha, \beta)\right)$$

Optimal parameters can be found via an iterative fixed-point method, successively computing the following two update equations alternatively

$$\phi_{n,d,z} \propto \beta_{w_{n,d},z} \exp\left(\Psi(\gamma_d) - \Psi\left(\sum_{j=1}^{k} \gamma_j\right)\right)$$

$$\gamma_d = \alpha_d + \sum_{n=1}^{N_d} \phi_{n,d}$$

where Ψ is the first derivative of the $\log \Gamma$ function.

Once optimal variational parameters have been estimated for each document, distribution $p(\theta, z|\gamma_d^*, \phi_d^*)$ provides a suitable approximation for the desired conditional $p(\theta, z|d, \alpha, \beta)$.

Using this technique, we can adapt EM algorithm to work in the variational framework

Algorithm 3.5 LDA algorithm

procedure LDA($p(X, Z|\theta)$,$\{x_i\}$,β_{min})
  Start with random parameter vectors $\alpha^{(0)}, \beta^{(0)}, \gamma^{(0)}, \phi^{(0)}$
  for each t **do**
    Step E:
    for each $d \in \mathcal{D}$ **do**
      Take $(\gamma_d^{(t,0)}, \phi_d^{(t,0)}) = (\gamma_d^{(t-1)}, \phi_d^{(t-1)})$.
      for each s **do**

$$\phi_{n,d,z}^{(t,s)} \propto \beta_{w_{n,d},z}^{(t)} \exp\left(\Psi(\gamma_d^{(t,s-1)}) - \Psi\left(\sum_{j=1}^{k} \gamma_j^{(t,s-1)}\right)\right)$$

$$\gamma_d^{(t,s)} = \alpha_d^{(t)} + \sum_{n=1}^{N_d} \phi_{n,d}^{(t,s-1)}$$

      end for
      Stop when variational parameters have converged. Take

$$(\gamma_d^{(t)}, \phi_d^{(t)}) = (\gamma_d^{(t,s)}, \phi_d^{(t,s)})$$

      Step M:
      Select $\beta^{(t+1)}$ as

$$\beta_{w,z}^{(t+1)} \propto \sum_{\{(n,d)|d\in\mathcal{D}, 1\leq n\leq N_d, w_{n,d}=w\}} \phi_{n,d,z}^{(t)}$$

      Select $\alpha^{(t+1)}$ as

$$\alpha^{(t+1)} = \underset{\alpha}{\operatorname{argmax}} \sum_{d\in\mathcal{D}} \log\Gamma\left(\sum_{j=1}^{k}\alpha_j\right) - \sum_{j=1}^{k}\log\Gamma(\alpha_j) + \sum_{j=1}^{k}(\alpha_j - 1)\left(\Psi(\gamma_{d,j}^{(t)}) - \Psi\left(\sum_{i=1}^{k}\gamma_{d,i}^{(t)}\right)\right)$$

    end for
    The loop stops either after a maximum number of iterations or when parameters have converged
  end for
end procedure

α estimation can be computed in M step using Newton-Raphson algorithm. This variational framework has the advantage of allowing both parameter learning and inference approximations using at each step the closest possible approximation within the tractable ones.

An alternative to the variational approach is using Markov chain Monte Carlo for estimating the conditional distributions $p(\theta, z|\alpha, \beta)$. and produce inference on

the aspect space, in the form of distributions $p(z_j|z_{-j}, d)$ for each latent factor assignment given the observed factors for the other terms in the document.

3.4 Equivalence between pLSA and LDA

The latent semantic generative bases of pLSA and LDA are clearly analogous. In both cases, aspects are assumed to be drawn for each document and words are drawn from distributions depending on the selected aspect. Moreover, both models consider that the aspects are chosen from mixing distributions which are particular to each document.

The main difference between pLSA and LDA is the theoretical consideration of those mixing proportions $p(z|d)$. While pLSA considers them as parameters estimated during the execution of the EM algorithm, LDA takes them as a new set of latent random variables θ_d. These random variables are meant to follow a Dirichlet distribution $\mathrm{Dir}(\alpha)$ depending on a vector parameter α which is assumed to be constant for the whole corpus.

Therefore, LDA generative model can be understood as a Bayesian regularization of the mixing distributions $p(z|d)$ in the pLSA model. Parameters $p(z|d) = \theta_{z,d}$ are given a prior Dirichlet distribution that collapses all the information trough a narrower vector of parameters $\alpha = (\alpha_z)_{z \in Z}$.

This imposes a clear difference in the degrees of freedom of the system. LDA β parameters correspond to pLSA parameter distributions $p(w|z)$ unequivocally, thus contributing in WK independent parameters to both models. Nevertheless, LDA α parameter corresponds, in a certain sense, to fixing priors $p(z)$, in contrast to pLSA $p(z|d)$ parameters. Therefore, LDA adds K independent parameters to its system, while pLSA adds a whole new set of KD independent parameters. The main consequence of this parameter counting is that pLSA parameter space dimension depends on the size of the corpus, while LDA parameter space only grows with the vocabulary.

Despite these differences, the common ground generative model allows us to connect inference estimation in both pLSI and PLSA. Girolami and Kabán (2003) prove that pLSI can be recovered from a certain approximate inference method within the LDA framework, if a certain α parameter is fixed. In order to obtain this equivalence, we have to introduce MAP estimators as an alternative to the variational inference used in Blei et al. (2003).

During the parameter estimation process, a maximum likelihood estimation is used, which leads to the need of computing the intractable posterior $p(d|\alpha, \beta)$. Instead of computing this posterior, a maximum a posteriori – maximum likelihood approximation can be taken. It consists on estimating the maximum a posteriori for the latent variable θ for each document in the corpus. Then, these estimations substitute the original latent variable θ in the posterior approximation for each document. MAP estimators being decoupled from the rest of the model simplify parameter dependencies in the computation of the posterior $p(d|\alpha, \beta)$, making a maximum a posteriori of the rest of parameters (β) possible.

In particular, if we fix $\alpha_z = 1$ for all $z \in \mathcal{Z}$, thus forcing a uniform prior on θ, we obtain $p(\theta|d, \beta) \propto p(d|\theta, \beta)$. Therefore, the MAP estimator coincides with the

maximum likelihood estimator, corresponding to

$$\theta_d^{MAP} = \theta_d^{ML} = \operatorname*{argmax}_{\theta} \log p(d|\theta, \beta) = \operatorname*{argmax}_{\theta} \sum_{w \in \mathcal{W}} n(w,d) \log \left(\sum_{z \in \mathcal{Z}} p(w|z,\beta)\theta_z \right)$$

Once θ_d^{MAP} is fixed, we compute the beta parameters by maximum likelihood

$$\beta^{ML} = \operatorname*{argmax}_{\beta} \sum_{d \in \mathcal{D}} \log p(d|\theta_d^{MAP}, \beta) = \operatorname*{argmax}_{\beta} \sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} n(w,d) \log \left(\sum_{z \in \mathcal{Z}} \beta_{w,z} \theta_{z,d}^{MAP} \right)$$

Combining both steps, an estimation by MAP-ML of the latent variable θ and parameter β would correspond to setting

$$(\theta^{MAP}, \beta^{ML}) = \operatorname*{argmax}_{\theta,\beta} \sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} n(w,d) \log \left(\sum_{z \in \mathcal{Z}} \beta_{w,z} \theta_{z,d} \right)$$

Taking into account that $\theta_{z,d}$ and $\beta_{w,z}$ are the parameters defining $p(z|d)$ and $p(w|z)$ respectively, we obtain that $(\theta^{MAP}, \beta^{ML})$ is the solution to pLSA maximum likelihood problem. Therefore, pLSA corresponds to a MAP estimation of LDA if Dirichlet α parameter is taken as $(1, \ldots, 1)$. Direct computation from the LDA generative model using this choice of α justifies directly pLSA fold-in estimations.

3.5 Topic models in IR Diversity

In general terms, diversification algorithms aim to diversify a set of aspects or intents that are usually unknown to the system, corresponding to the different facets of the overall user information needs. Even if an explicit set of aspects is available (like ODP taxonomy in web search or movie genres in a recommendation task), they represent mere approximations to the real latent intentions of the user.

Latent aspect models seem to be natural tools to study this problem. The main idea consists in using topic models to abstract the semantics behind the particular information tokens available. As we saw, this latent semantics are encoded in an aspect space which is expected to approximate the space of abstract intents of the user.

Topic models extract latent semantic information from the observed data based on a robust probabilistic framework. Most of the aspect extraction methodologies described in the last chapter are either based on prior additional information about the observed data – like an ODP classification (Agrawal et al., 2009) or Wikipedia disambiguation pages (Rafiei et al., 2010; Welch et al., 2011) – or are based on an algebraic compression of information which is heavily dependent on the explicit representation of the data – like matrix factorization methods (Vargas et al., 2011).

While the latter algorithms are indeed really effective, one can't obviate the fact that the extracted aspects rely on a more or less heuristic choice of data representation. On the other hand, topic models describe the interaction between the observed variables and the latent semantic information in a generative intrinsic way, as far as the used generative model fits the data.

The explicit use of Latent Semantic Analysis in diversity tasks has been explored by diverse authors. As a possible continuation on their work on subtopic retrieval, Zhai et al. (2003) mention the possibility of using pLSA and LDA to model query subtopics. Carterette and Chandar (2009) propose the use of LDA to model query facets, or sub-queries. They use this estimated facet information to build rankings maximizing facet covering in the so called FM-LDA (Facet Model with LDA), therefore maximizing the overall diversity of the results. Among other diversification strategies, He et al. (2011) use FM-LDA together with clustering methods to develop query-specific diversified rankings. Krestel and Fankhauser (2012) propose the use latent factors computed by LDA to represent documents within the vector space of mixing proportions. He et al. (2012) uses Laplacian PLSA – a regularized generalization of pLSA due to Cai et al. (2008) – as a source of latent subtopics for a multi-source subtopic approach to diversity.

Vargas et al. (2012b) explicitly study the suitability of automatically extracted intent spaces as means of approximation of user intents for diversification purposes. They analyze the aspect space informativeness and use simulated data to explore the effect of the aspect space size in the overall diversity reached by the diversification system in terms of the ranking distance from the diversified results to the baseline, understood as a measure of the "room for change".

While the potential of latent factors for diversification has been proved, there exist several open questions about the optimization of the aspect extraction methodology in order to enhance the final diversity of the system. For example, Krestel and Fankhauser (2012) finds that latent factor representation of documents performs slightly worse than classic language models as a representation space for their model. On the other hand, they remark that topic models accomplish diversification with a significantly less amount of re-ranking and open the research objective of developing similar approaches to diversification that focus on the topical content of documents.

Through the next chapters, we will deepen in the objective of optimizing the process of aspect extraction for diversity enhancement. We will propose a new theoretical framework for generalizing pLSA model to incorporate relevance information to the dynamics of the intent space construction and provide a general probabilistic algorithm for incorporating arbitrary sources of information to the model, extending the potential application of pLSA to different kind of diversity problems. Moreover, further optimization methods will be studied for improving the obtained intent space, such as several aspect filtering methods and precise analytical estimations for the effect of the aspect space size to diversity.

Chapter 4

Latent Semantic aspects for Diversification

In the last chapter, we showed how automatically extracted intent spaces could be used to enhance search and recommendation diversity. Latent semantic information has been used in the literature to approximate user intents in diversity tasks. Probabilistic approaches to latent semantic extraction like pLSA have been proven to be really effective in ad-hoc tasks both as an indexing method or in recommendation tasks. Nevertheless, the use of probabilistically extracted latent semantic spaces as intent spaces for diversification has not been fully studied.

Our objective is to develop probabilistic aspect extraction models which are able to construct intent spaces that are more suitable for diversification. Starting from Hofmann's Probabilistica Latent Semantic Analysis, we aim to develop enriched probabilistic procedures that incorporate additional sources of information to the semantic model, such as document relevance, user profile information or additional features. This would generate more informative and precise semantic spaces which are expected to help diversification algorithms obtain more diverse results.

Focusing on the search diversity problem, we target to build spaces that capture better the relations between documents query and latent aspects by introducing baseline relevance information and query-specific information in the latent semantic model. Common diversification algorithms like IA-Select or xQuAD use these relations, in the form of distributions $p(z, d, q)$, to compute the diversified rankings. Therefore, refining aspect spaces with query-specific relevance information intuitively leads to more precise re-rankings, therefore improving the overall diversity.

In order to do so, we will introduce the concept of utility-biased likelihood estimator as a theoretical probabilistic framework for enhancing latent probabilistic models with a notion of "utility" for the observed data. This "utilitiy" will correspond generally to a measure of how relevant an observation is to the overall model. In our scenario, we will use it to make ground relevance impact the dynamics of the EM algorithm used to build the latent space.

An "utility-biased" Expectation Maximization algorithm is introduced, as a generalization of EM algorithm that allows us to incorporate the utility functional to the EM dynamics. We will use it to develop explicit Bayesian formulas for an au-

tomatic latent semantic space extraction algorithm in case the latent model can be described as a Bayesian network. In this scenario, an explicit tempered version will be developed, analogous to the one described for classic EM in the last chapter. Both variations combined will allow us to introduce arbitrary feature variables and an utility notion to the tempered pLSA model.

This general algorithm will be applied to different information retrieval tasks. We will analyze deeply some applications to search and recommendation diversity, and introduce some possible applications to other tasks like personalization or content based recommendation.

Finally, further aspect filtering optimization techniques and fold-in strategies will be explained and a geometric information-theoretical interpretation of the algorithm will be studied. Experimental results showing the effectiveness of the proposed methods for search and recommendation tasks will be exposed.

4.1 Approach

We propose to use latent factors obtained from a probabilistic latent semantic analysis as aspects for diversification. This approach was initially mentioned by Zhai et al. (2003), who noticed the possibility of modeling subtopics with LDA as a possible direction of future work. Some authors have recently explored the effectiveness of pLSA as an intent space builder for diversification (He et al., 2012; Vargas et al., 2012a,b), but a complete deep study has not been developed yet. In particular, the possibility of changing pLSA dynamics in order to obtain more suitable latent factors for diversification has not been addressed, and the dependency of the final diversity results on the properties of the aspect space has not been established completely.

On the other hand, the balance between outputting relevant or pure diverse results in retrieval systems has always been a main issue when enhancing the diversity of the results shown to user. Thus, it has been subject to multiple optimizations. While there have been advances in producing diversification algorithms that treat document relevance properly and produce resultsets that can be considered both as relevant and diverse by the final users (Santos et al., 2010; Vargas et al., 2012a), ground relevance of documents remain transparent to the construction process of the intent spaces which are used to retrieve these resultsets posteriorly.

Finally, query-specific latent information has been exploited to improve both retrieval and diversity. The use of query-specific cluster-based retrieval has long been proposed as a way to identify the top relevant documents for a given query, based on the idea that relevant documents for each query tend to be clustered together if query-wise clusters are built (Tombros et al., 2002). Moreover, He et al. (2011) proved that query-specific clusters can be used as an approximation for query subtopics. They proved that clustering documents within each single query separately led to an intent space that can be fed to a diversification algorithm in order to improve the overall diversity of the results.

Our approach aims to incorporate the previous ideas to a single unified theoretically-robust probabilistic framework for building enhanced model-driven latent semantic spaces. In the search scenario, we describe a relevance aware query-specific probabilistic latent semantic analysis (RapLSA) that is meant to produce aspect spaces

distinguishing the main latent semantic aspects of each query. The dynamics of the algorithm are altered from the ones of pLSA in a way that prioritizes adjusting the semantics of the top relevant documents and neglecting spam documents in the collection. In order to do so, the algorithm uses the baseline ranking and, if available, any kind of specific relevance information, to ponder the significance of each observed information token to the complete data model.

The resulting aspects are then interpreted as a query-specific intentions, or subqueries. Obtained document-aspect and query-aspect distributions are led to a diversification algorithm, like xQuAD or IA-Select, in order to obtain the final diversified results. Using RapLSA is shown to improve the overall diversity of the results in comparison with classic pLSA and query fold-in strategies.

In recommendation tasks, the theoretical framework is used to build user profiles that incorporate item relevance to the intent space construction dynamics. Although Hofmann (2003, 2004) has already incorporated explicit ratings (both categorical and Gaussian) to the pLSA recommendation model successfully, this is done in a way that does not alter the dynamics of the semantic part of the model significantly, as the topology and underlying relevance information of ratings are not transfered to the dynamics of the model. In the case of categorical ratings, rating values are symmetric and indistinguishable from the model point of view. There is no connection between the rating relevance information and its impact on the model. For example, if we consider the user-aspect distribution, the aspect with highest probability may not correspond to the one representing the items that the user liked the most, but, more likely, just the one representing the major class of observed items (even if the are all rated low). Nevertheless, if we used the latent factors for diversification, this kind of factor would be still prioritized.

The situation is analogous in the continuous case. Whereas ratings do retain their topology in the model, it remains restricted to the rating variable and it does not transfer the relevance information to the dynamics of the algorithm. On the other hand, RapLSA recommendation version is explicitly designed to incorporate relevance information and, in general, any additional item features to the model, prioritizing learning the ratings of more relevant items for each user. This way, high predicted rates are expected to be more accurate and, in general, the aspect space is expected to reflect better the user needs following a smoothed positive feedback strategy.

The general probabilistic model is shown to be suitable for other information retrieval tasks and diversity models, such as personalization, content based recommendation, or a variant of collaborative filtering recommendation models that infers an intent aware similarity function, augmenting the capacity of the system to develop topic-specific recommendations, thus improving the overall diversity when used in conjunction with a diversification algorithm.

Finally, some undesirable properties of the aspect distributions generated by the general model (including pLSA) are identified and a set of aspect filtering techniques is introduced to improve the diversity produced by common diversification algorithms that use these aspect spaces. In particular, we introduce a filter that increments the sparsity of aspect distributions, another one that neutralizes possible prior aspect biases and provides the diversification algorithm a “pure diversity”

aspect space, and a third one that decreases the entropy of the aspect distributions.

4.2 Relevance aware Probabilistic Latent Semantic Analysis

As we discussed through the exposition of the xQuAD algorithm, there exists an inherent trade-off between maximizing the diversity or novelty of a resultset and its pure relevance. While xQuAD effectively equilibrates this tradeoff, given a suitable intent space, none of the usual automatic aspect extraction algorithms (matrix factorization algorithms, pLSA, LDA, etc.) take into account the relevance of the documents while building the aspect space.

For example, it is known that both common standard training datasets and resultsets from commercial search engines present a certain non-negligible percentage of spam among the retrieved documents (Cormack et al., 2011). While applying algorithms like pLSA, the content of these documents is considered by the system at the same level as the content of the most relevant ones. This may produce a distortion in the formation of the semantic space, as spam documents can contain almost gibberish language (Crane and Trotman, 2012). For this reason, some authors (He et al., 2012) choose to filter spam before applying any kind of semantic algorithm.

In general, after applying an automatic intent space extraction algorithm, applying an algorithm like xQuAD will deprecate the semantic information of documents with low enough baseline score. The global score being a combination of the pure diversity score (probability of information being new given the already selected documents) and the low baseline score will usually make these documents sink below the top relevant documents, independently of their semantics. The algorithm may, indeed, produce some great permutations of documents in the lower part of the ranking, but the later has really little effect on the user experience and common diversity or relevance metrics based on cascade browsing models.

Therefore, the semantics of lower rank documents will have a lot of impact in the construction of the intent space in comparison to the fewer top documents (simply because all documents are considered at the same level and relevant documents are the less). In contrast the semantic of the lower documents will have almost no effect, from the user's perspective, in the final ranking of the lower rank documents. Thus, documents being considered at the same level independently on its relevance allow the semantics of the non-relevant documents to greatly perturb the semantic representation of the high relevance ones. At the end, the semantics that discriminate the novel information among the top documents is not that of the top documents. On the other hand, it depends heavily on the interaction between them and the other, irrelevant, documents in the corpus.

In order to overcome this effect, we present a version of pLSA that takes into account the baseline relevance information of the documents and the semantic information of the query in order to build a query-specific latent semantic space that allows diversification algorithms to maximize the diversity of the top part of the ranking, introducing pseudo-relevance feedback effects in the overall re-ranking process.

4.2.1 Abstract Probabilistic Model

We will start proposing a general probabilistic modification of the EM algorithm that will allow us to introduce the relevance in the model. Suppose that we have a parametric latent variable model, with observed variables X, unobserved variables Z and parameter vector Θ. Let us suppose that we have a set of observations $\overline{X} = \{x_1, \ldots, x_n\}$ with frequencies $f_1, \ldots, f_n$.

Depending on the background generative model, we may be able to predict that there might exist a set of statistically relevant observations (high frequency ones) which are either not useful for our model or, for some reason, are better to be filtered out before fitting its parameters. The aim of this model is to make the EM algorithm be able to take this a priori information into account in order to improve its effectiveness.

Let us suppose that together with the observed data, we are given an utility function $f : \overline{X} \to [0, 1]$, obtained from a priori knowledge of the observed data, which associates each observation a score that measures how relevant it is. We will further assume that it is normalized i.e., that $\sum_{i=1}^{n} f(x_i) = 1$.

From (3.1.1), the log-likelihood of the observed data would be given by

$$\log(\mathcal{L}(\theta)) = \sum_{i=1}^{n} f_i \log(p(x_i|\theta))$$

In the case that the samples are enhanced with the additional utility data f, it makes sense to perturb the previous functional in order to increase the impact of the high utility observations and weak the effect of the low utility ones. We propose the use of the following utility-biased log-likelihood functional notion instead

Definition 4.2.1. *Let $(\overline{X}, f)$ be a pair consisting on a set of observations $\overline{X}$ together with a normalized utility function $f : \overline{X} \to [0, 1]$. We denote the functional*

$$\mathcal{L}_f(\theta) = \sum_{i=1}^{n} f(x_i) \log(p(x_i|\theta))$$

both as the likelihood of the pair $(\overline{X}, f)$ or as the f-likelihood of the observed data $\overline{X}$.

Definition 4.2.2. *The maximum-likelihood estimator for the parameters Θ given the enhance observed data $(\overline{X}, f)$ is given by*

$$MLE_f(\Theta) = \operatorname*{argmax}_{\theta}(\mathcal{L}_f(\theta))$$

It is clear that taking $f(x_i) = \frac{f_i}{\sum_{i=1}^{n} f_i}$, we have $\mathcal{L} = \mathcal{L}_f$ and, thus, $MLE(\Theta) = MLE_f(\Theta)$. Therefore, this notion of likelihood functional is a generalization of the usual likelihood $\log(p(\overline{X}|\theta))$. This biased estimators take a really meaningful form in case that the utility function f can be given a model-driven probability interpretation. Suppose that there exists a priori probability distribution $\tilde{x}_i$ over

the observed data X for which f is an approximation over the samples, in the sense that $\tilde{p}(x_i) \sim f(x_i)$. Then

$$\mathcal{L}_f(\theta) \sim \sum_{i=1}^{n} \tilde{p}(x_i) \log(p(x_i|\theta)) = H(\tilde{p}) - \mathrm{KLD}\left(\tilde{p} \| p(\cdot|\theta)\right) \tag{4.2.1}$$

Thus, we get

$$MLE_f(\Theta) = \operatorname*{argmax}_{\theta} \left(H(\tilde{p}) - \mathrm{KLD}\left(\tilde{p} \| p(\cdot|\theta)\right)\right) = \operatorname*{argmin}_{\theta} \mathrm{KLD}\left(\tilde{p} \| p(\cdot|\theta)\right)$$

This way, we get that the biased maximum likelihood estimator for Θ is the vector θ for which the distribution $p(\cdot|\theta)$ is the closest possible to the prior $\tilde{p}$ in the Kullback-Leibler pseudo-metric. In other words, the distribution $p(\cdot|MLE_f(\Theta))$ is the best approximation of $\tilde{p}$ among all probability distributions of the observed variables that factor through the given parametric latent model.

Using this theoretical framework, we can describe a general abstract procedure in order to build relevance aware algorithms for automatic intent space construction. The general method is to perturb pLSA retrieval models (either, search, recommendation, personalization, etc.) by introducing relevance in the form of an utility function and then applying EM or TEM algorithm to build an intent space. As an example (to be developed in a next section), we will consider the classic pLSA algorithm (Hofmann, 1999a) as used for building intent spaces for search diversity. A possible application of this method would be to introduce a query variable in the generative method, represented in a probability space $\mathcal{Q}$. We pass from a latent model over $\mathcal{W} \times \mathcal{D}$ to a latent model over $\mathcal{W} \times \mathcal{D} \times \mathcal{Q}$. This can be done in several ways that we will consider later on.

The observed data in classic pLSA correspond to pairs (w, d), so we have a frequency scheme of this pairs given by number of occurrences $n(w, d)$. From this frequencies we can estimate a conditional a priori distribution $p(w|d)$, given by

$$\tilde{p}(w|d) = \frac{n(w, d)}{\sum_{w \in \mathcal{W}} n(w, d)}$$

On the other hand, we can estimate a probability distribution $\tilde{p}(d|q)$ from the search baseline. This can be done, for example, using a discount function of the ranking of the document for the given query. Depending on the structure of the search engine, it is possible that it even outputs such a probability distribution (or another equivalent one, like $\tilde{p}(q|d)$) as a model driven document score (for example, if it is using language models). Either of these ways allow us to compute a distribution $\tilde{p}(d, q)$ (taking uniform distribution over $\mathcal{Q}$, for example, but the method allows more representative utility functionals, perhaps involving inherent query ambiguity, for instance) and, finally, a priori distribution for the observed data

$$\tilde{p}(w, d, q) = \tilde{p}(w|d)\tilde{p}(d, q)$$

This distribution will act as an utility function of the samples (w, d, q) of triples such that the word w belongs to document d and document d appears in the baseline ranking for query q. Now, we can use EM (and even TEM) algorithm to approximate the maximum likelihood estimator for the observed $((w, d, q), \tilde{p})$. The choice of

distribution $\tilde{p}(d, q)$ makes observations related to more relevant documents for each query to be taking in further consideration when it comes to building the corresponding intent space. As we mentioned, this scenario will be treated in depth later on, but it is a clear representative on how using utility-biased maximum likelihood instead of the regular one can solve our initial problem, making EM algorithm capable of using the priori relevance information, which was omitted in classic pLSA, and making us capable of building more discriminant and effective intent spaces oriented for diversification.

4.2.2 Latent factor estimator via the EM algorithm

Once we have selected an utility function (that we will assume, from this point on, to have the form of a probability distribution $\tilde{p}$ of the observed data), we get an utility-based maximum likelihood estimator problem for the parameters of the latent variable model. We can adapt the EM and TEM algorithms in order to work with this kind of biased functionals, thus obtaining iterative algorithms which can find locally optimal approximations for the parameters.

In this section we will obtain explicit, yet general, equations for E and M steps of this modified EM algorithm in terms of the graph structure of the complete data graphical model. We will see that both steps acquire a meaningful Bayesian form when the utility functional comes from a model-driven probability distribution.

Let $S = X \cup Z$ be the random variables of the complete data model. Let us assume that the complete model for S is a Bayesian network. For each node $S_i \in S$, let $S_{\pi(i)}$ denote the set of parents of node S_i in the graphical model. The probability of the complete data is then given by

$$p(X, Z) = \prod_{i=1}^{|S|} p(S_i | S_{\pi(i)})$$

By a convenient abuse of notation, giving a sample (x, z) of the variables X, Z we will denote by $S_i(x, z)$ and $S_{\pi(i)}(x, z)$ respectively the projection of sample (x, z) to variable S_i and variables in $S_{\pi(i)}$ respectively. Therefore

$$p(x, z) = \prod_{i=1}^{|S|} p(S_i(x, z) | S_{\pi(i)}(x, z))$$

Let us suppose that we have samples $\overline{X} = \{x_1, \ldots, x_n\}$ of observed variables X, enhanced with a prior utility functional, in the form of a given prior distribution $\tilde{p}(X)$ supported over $\overline{X}$. The enhanced observed data is then modeled by a parametric latent variable model, with latent variables Z and parameter vector Θ corresponding to the unknown categorical distributions $p(S_i | S_{\pi(i)})$ for each node S_i in the graphical model. The utility biased maximum likelihood of the parameter vector $\Theta = \theta$ given $(\overline{X}, \tilde{p})$ is then given by

$$\mathcal{L}_{\tilde{p}}(\theta) = \sum_{i=1}^{n} \tilde{p}(x_i) \log p(x_i | \theta) = \sum_{i=1}^{n} \tilde{p}(x_i) \log \left(\sum_{z \in Z} \prod_{j=1}^{|S|} p(S_j(x_i, z) | S_{\pi(j)}(x_i, z), \theta) \right)$$

Maximizing the likelihood directly is clearly intractable. We will use the same approach that EM, using $p(z|x_i, \theta_t)$ as coefficients for applying Jensen inequality in order to find a minorizing function. The calculation for each x_i is exactly the same as that of (3.1.5). Using linearity of minorizing functions proved in last chapter yields the following minorizing functional

$$g_{\tilde{p}}(\theta, \theta_t) = \sum_{i=1}^{n} \tilde{p}(x_i) \mathbb{E}_{Z|x_i,\theta_t}[\log p(x_i, Z|\theta)] + \sum_{i=1}^{n} \tilde{p}(x_i) H(p(Z|x_i, \theta_t))$$

As the last factor is constant in θ, this is equivalent to maximizing the functional

$$Q_{\tilde{p}}(\theta, \theta_t) = \sum_{i=1}^{n} \sum_{z \in Z} p(z|x_i, \theta_t) \tilde{p}(x_i) \sum_{j=1}^{|S|} \log p(S_j(x_i, z)|S_{\pi(j)}(x_i, z))$$

subject to the usual normalization constraints, $\sum_{s \in S_i} p(s|v) = 1$ for all $i = 1, \ldots, n$ and every $v \in S_{\pi(i)}$. Taking a Lagrange multiplier $\lambda_{i,v}$ for each i and v, M step correspond to maximizing the functional

$$\mathcal{F} = \sum_{i=1}^{n} \sum_{z \in Z} p(z|x_i, \theta_t) \tilde{p}(x_i) \sum_{j=1}^{|S|} \log p(S_j(x_i, z)|S_{\pi(j)}(x_i, z)) - \sum_{j=1}^{|S|} \sum_{v \in S_{\pi(j)}} \lambda_{j,v} \left(\sum_{s \in S_j} p(s|v) - 1 \right) \quad (4.2.2)$$

Deriving with respect to each parameter $p(s|v)$, with $s \in S_j$, $v \in S_{\pi(j)}$, we have

$$0 = \frac{\partial \mathcal{F}}{\partial p(s|v)} = \frac{1}{p(s|v)} \sum_{\{(x,z) \in \overline{X} \times Z | S_j(x,z) = s, S_{\pi(j)}(x,z) = v\}} p(z|x, \theta_t) \tilde{p}(x) - \lambda_{i,v}$$

Therefore,

$$p(s|v) \propto \sum_{\{(x,z) \in \overline{X} \times Z | S_j(x,z) = s, S_{\pi(j)}(x,z) = v\}} p(z|x, \theta_t) \tilde{p}(x)$$

and normalization constraint yields the explicit form of the M step

$$p(s|v, \theta_{t+1}) = \frac{\sum_{\{(x,z) \in \overline{X} \times Z | S_j(x,z) = s, S_{\pi(j)}(x,z) = v\}} p(z|x, \theta_t) \tilde{p}(x)}{\sum_{s \in S_j} \sum_{\{(x,z) \in \overline{X} \times Z | S_j(x,z) = s, S_{\pi(j)}(x,z) = v\}} p(z|x, \theta_t) \tilde{p}(x)} \quad (4.2.3)$$

On the other hand, the E step is obtained directly by Bayesian inference from distributions $p(S_j|S_{\pi(j)})$ computed in the last step

$$p(z|x_i, \theta_t) = \frac{\prod_{j=1}^{|S|} p(S_j(x_i, z)|S_{\pi(j)}(x_i, z), \theta_t)}{\sum_{z \in Z} \prod_{j=1}^{|S|} p(S_j(x_i, z)|S_{\pi(j)}(x_i, z), \theta_t)} \quad (4.2.4)$$

Therefore, we observe that both E and M steps correspond to Bayesian estimation of the corresponding distributions. E step computes the aspect distributions

$p(z|x_i, \theta_t)$ from the Bayesian network assuming the parameter estimation for conditional distributions $p(S_j|S_{\pi(j)}, \theta_t)$ computed in the last step. On the other hand, M step computes the distributions $p(S_j|S_{\pi(j)}, \theta_{t+1})$ from the aspect distribution $p(z|x_i, \theta_t)$ and the prior utility distribution $\tilde{p}(\theta_i)$, applying Bayes theorem to obtain a distribution $p(x_i, z|\theta_t) = p(z|x_i, \theta_t)\tilde{p}(x_i)$ and marginalizing with respect to the observed variables. Putting this together, we obtain the following general algorithm

Algorithm 4.1 Utility-biased Expectation-Maximization algorithm

procedure EM-ALGORITHM($p(X, Z|\theta)$,$\{x_i\}$,$\tilde{p}(x_i)$)
  Start with a random parameter vector θ_0, corresponding to random normalized probability distributions $p(S_i|S_{\pi(i)}, \theta_0)$
  for each t **do**
    Step E:
    for each $x_i \in \overline{X}$ **do**
      Compute

$$p(z|x_i, \theta_t) = \frac{\prod_{j=1}^{|S|} p(S_j(x_i, z)|S_{\pi(j)}(x_i, z), \theta_t)}{\sum_{z \in Z} \prod_{j=1}^{|S|} p(S_j(x_i, z)|S_{\pi(j)}(x_i, z), \theta_t)}$$

    end for
    Step M: Select θ_{t+1} as
    for each $i = 1$ to $|S|$ **do**
      Compute

$$p(s|v, \theta_{t+1}) = \frac{\sum_{\{(x,z) \in \overline{X} \times Z | S_j(x,z)=s, S_{\pi(j)}(x,z)=v\}} p(z|x, \theta_t)\tilde{p}(x)}{\sum_{s \in S_j} \sum_{\{(x,z) \in \overline{X} \times Z | S_j(x,z)=s, S_{\pi(j)}(x,z)=v\}} p(z|x, \theta_t)\tilde{p}(x)}$$

    end for
  end for
  The algorithm stops either after a maximum number of iterations or when $\|\theta_{t+1} - \theta_t\|$ is less than a certain threshold
end procedure

It is worth noticing that step M becoming a direct Bayesian computation allows us to apply any of the classical Bayesian network exact inference algorithms (variable elimination, clique tree propagation, etc.) to compute the M step faster than direct calculation. The use of an approximate inference algorithm (loopy belief propagation, generalized belief propagation, variational inference, etc.) in order to compute approximations for the M step parameters may be possible if it is shown to increase functional $Q_{\tilde{p}}(\theta, \theta_t)$. In this case, a GEM-like algorithm would be being used instead of an EM algorithm. Increasing of the global likelihood is warranted at each step, but convergence properties are to be analyzed in each scenario.

The previous equations approximate all conditional distributions $p(S_j|S_{\pi(j)})$. If any of these distributions only involves observed variables, it can be estimated with the incomplete data model alone. In this case, the algorithm can be provided with some of the probability distributions $\tilde{p}(S_j|S_{\pi(j)})$. We may substitute the param-

eters $p(S_j|S_{\pi(j)})$ by the corresponding known distributions in functional (4.2.2). This distributions don't need to be estimated anymore during the M step and E step equations clearly correspond to substituting the parameters by the distributions $\tilde{p}(S_j|S_{\pi(j)})$. This is essentially equivalent, yet faster, to executing the full EM algorithm. In order to prove this, let S_j be an observed variable such that $S_{\pi(j)}$ only contains observed variables. Then $S_j(x,z)$ and $S_{\pi(j)}(x,z)$ are constant in z. Denoting this projections as $S_j(x)$ and $S_{\pi(j)}(x)$ respectively, we get

$$\{(x,z) \in \overline{X} \times Z | S_j(x,z) = s, S_{\pi(j)}(x,z) = v\} = \{x \in \overline{X} | S_j(x) = s, S_{\pi(j)}(x) = v\} \times Z$$

Therefore, as $p(z|x,\theta_t)$ is normalized for every $x \in \overline{X}$ and every step t, M step approximation yields

$$p(s|v,\theta_{t+1}) = \frac{\sum_{\{(x,z)\in\overline{X}\times Z|S_j(x,z)=s,S_{\pi(j)}(x,z)=v\}} p(z|x,\theta_t)\tilde{p}(x)}{\sum_{s\in S_j}\sum_{\{(x,z)\in\overline{X}\times Z|S_j(x,z)=s,S_{\pi(j)}(x,z)=v\}} p(z|x,\theta_t)\tilde{p}(x)} =$$
$$\frac{\sum_{\{x\in\overline{X}|S_j(x)=s,S_{\pi(j)}(x)=v\}}\sum_{z\in Z} p(z|x,\theta_t)\tilde{p}(x)}{\sum_{s\in S_j}\sum_{\{x\in\overline{X}|S_j(x)=s,S_{\pi(j)}(x)=v\}}\sum_{z\in Z} p(z|x,\theta_t)\tilde{p}(x)} =$$
$$\frac{\sum_{\{x\in\overline{X}|S_j(x)=s,S_{\pi(j)}(x)=v\}} \tilde{p}(x)}{\sum_{s\in S_j}\sum_{\{x\in\overline{X}|S_j(x)=s,S_{\pi(j)}(x)=v\}} \tilde{p}(x)}$$

Therefore, $p(s|v,\theta_{t+1})$ does not depend on $p(z|x,\theta_t)$, it's constant through all iterations and it's equal to the distribution $\tilde{p}(s|v)$ computed directly with the incomplete data model. The only difference between both forms of computation is the first E step. If distributions $\tilde{p}(s|v)$ are used, this first step uses these exact distributions for computing the first approximation $p(z|x,\theta_0)$. Otherwise, it uses random distributions, and starts using the real ones after the second E step. As the rest of the parameters remain random at first approximation, the effect on convergence and effectiveness can be considered negligible. Finally, this version of EM can also be equivalently stated following Neal and Hinton (1998) structure. Let $q = (q_1(z), \ldots, q_n(z))$ represent an array of n distributions, one for each observation in $\overline{X}$. Let us consider the functional

$$F_{\tilde{p}}(q,\theta) = \sum_{i=1}^{n} \tilde{p}(x_i)\left(\mathbb{E}_{q_i}[p(z,x_i|\theta)] + H(q_i)\right) \tag{4.2.5}$$

We will prove that E and M steps are equivalent to alternatively maximizing functional $F_{\tilde{p}}$ with respect to q and θ. Using equation (3.1.7), we obtain an analogous correspondence to that of (3.1.8)

$$F_{\tilde{p}}(q,\theta) = -\sum_{i=1}^{n} \tilde{p}(x_i)\,\mathrm{KLD}\left(q_i \| p(z|x_i,\theta)\right) + \sum_{i=1}^{n} \tilde{p}(x_i)\log p(x_i|\theta) =$$
$$-\sum_{i=1}^{n} \tilde{p}(x_i)\,\mathrm{KLD}\left(q_i \| p(z|x_i,\theta)\right) + \mathcal{L}_{\tilde{p}}(\theta) \tag{4.2.6}$$

As the last term is constant in q, Gibb's inequality implies that $F_{\tilde{p}}(\cdot,\theta)$ attains a unique global maximum at $q_i(z) = p(z|x_i,\theta)$. Thus, computation of E step is

equivalent to maximizing $F_{\tilde{p}}$ with respect to q. On the other hand, it is obvious that $F_{\tilde{p}}(p(z|x_i,\theta_t),\theta) = g(\theta,\theta_t)$, so maximizing $F(p(z|x_i,\theta_t),\theta)$ with respect to θ is completely equivalent to step M. Then, we get the following algorithm

Algorithm 4.2 Alternate form of the utility-biased EM algorithm

1: **procedure** EM-ALGORITHM($p(X,Z|\theta)$,$\{x_i\}$,$\tilde{p}(x_i)$)
2: Start with a random vector parameter θ_0
3: **for** each t **do**
4: Step E: Select $q^{(t)} = \left(q_1^{(t)}(z), \cdots, q_n^{(t)}(z)\right)$ as

$$q^{(t)} = \operatorname*{argmax}_q(F_{\tilde{p}}(q,\theta_t))$$

5: Step M: Select θ_{t+1} as

$$\theta_{t+1} = \operatorname*{argmax}_\theta(F_{\tilde{p}}(q^{(t)},\theta))$$

6: **end for**
7: The algorithm stops either after a maximum number of iterations or when $\|\theta_{t+1} - \theta_t\|$ is less than a certain threshold
8: **end procedure**

4.2.3 Tempering

Following the ideas of Rose et al. (1990), Rose (1991) and Ueda and Nakano (1998), discussed in last chapter, we can perturb the previous EM model with a tempering, parametrized by an inverse computational temperature β, in order to improve its effectiveness. If we follow the tempering methodology exposed by Ueda and Nakano (1998), we expect this to allow the algorithm to converge to better local maxima of the likelihood functional. On the other hand, if we adapt the Hofmann (1999b) "inverse" tempering strategy, more general distributions are expected to arise from the algorithm, thus decreasing the chances of overfitting the data. Both strategies are based on applying the maximum entropy principle on the basic Helmholtz energy functional of the algorithm.

We will start by interpreting our utility-biased maximum likelihood in the Helmholtz energy framework. As in the last section, we will suppose that the utility function is given by a probability distribution $\tilde{p}$ over X supported over $\overline{X}$. In classic TEM, for each sample $x_i \in \overline{X}$, latent variables are considered as states of a probabilistic system with state distribution $q_i(z)$, such that the energy of each state is given by $-\log p(z,x_i|\theta)$. TEM step E computes state distributions $q_i(z)$ with maximum entropy among those with the same energy and M step computes the parameter vector θ that correspond to the minimum global energy given the state distributions $q_i(z)$.

In contrast, we will consider a single probabilistic physical system, composed of $|\overline{X}|$ mixed overlapped subsystems, each one with state space Z. Each system $x_i \in \overline{X}$ occurs with probability $\tilde{p}(x)$ and, as in the EM case, they have state distribution $q_i(z)$ and energy $-\log p(z,x_i|\theta)$, computed through the complete data model. Thus,

the total energy of the complete system is given by

$$E(q,\theta) = \sum_{i=1}^{n} \tilde{p}(x_i)\mathbb{E}_{q_i}\left[-\log p(z,x_i|\theta)\right]$$

On the other hand, the total entropy of the system is not given as the sum of the entropies of probabilities q_i. In this case, the expected total entropy is given by

$$H(q) = \mathbb{E}_{\tilde{p}}\left[H(q_i)\right] = \sum_{i=1}^{n} \tilde{p}(x_i)H(q_i)$$

Following the maximum entropy principle, the modified E step is now stated as maximizing the expected total entropy within the same fixed expected total energy $E(q,\theta_t) = E$. Using a Lagrange multiplier β for the energy and Lagrange multipliers λ_i for each observation x_i, this is equivalent to maximizing the functional

$$\begin{aligned} L(q,\beta) = \sum_{i=1}^{n} \tilde{p}(x_i)H(q_i) - \beta\left(\sum_{i=1}^{n} \tilde{p}(x_i)\mathbb{E}_{q_i}\left[-\log p(z,x_i|\theta_t)\right] - E\right) - \\ \sum_{i=1}^{n} \lambda_i\left(\sum_{z\in Z} q_i(z) - 1\right) = -\sum_{i=1}^{n} \tilde{p}(x_i)\sum_{z\in Z} q_i(z)\log q_i(z) + \\ \beta\sum_{i=1}^{n}\sum_{z\in Z} \tilde{p}(x_i)q_i(z)\log p(z,x_i|\theta_t) - E\beta - \sum_{i=1}^{n}\lambda_i\left(\sum_{z\in Z} q_i(z) - 1\right) \end{aligned} \tag{4.2.7}$$

Differentiating with respect to $q_i(z)$ yields

$$0 = \frac{\partial L(q,\theta)}{\partial q_i(z)} = -\tilde{p}(x_i)\left(\log(q_i(z)+1) + \beta\tilde{p}(x_i)\log p(z,x_i|\theta) - \lambda_i\right.$$

Solving for $q_i(z)$ yields

$$q_i(z) = p(z,x_i|\theta_t)^{\beta}\exp\left(-\frac{\lambda_i}{\tilde{p}(x_i)} - 1\right)$$

so $q_i(z) \propto p(z,x_i|\theta_t)^{\beta}$. As the distribution is normalized we get the tempered E step

$$q_i^{(t)}(z) = \frac{p(z,x_i|\theta_t)^{\beta}}{\sum_{z\in Z} p(z,x_i|\theta_t)^{\beta}} = \frac{\left(\prod_{j=1}^{|S|} p(S_j(x_i,z)|S_{\pi(j)}(x_i,z),\theta_t)\right)^{\beta}}{\sum_{z\in Z}\left(\prod_{j=1}^{|S|} p(S_j(x_i,z)|S_{\pi(j)}(x_i,z),\theta_t)\right)^{\beta}} \tag{4.2.8}$$

The same discussion about the choice of the β parameter made in the last chapter clearly holds for this utility-biased EM. The value of the β parameter would be determined by equation $E(q,\theta_t) = E$ as a function of the energy level E. As the choice of E is essentially arbitrary, we can directly reparametrize the equations as a function of β and fix it as a parameter of the algorithm.

This way, utility-biased TEM algorithm correspond to alternatively maximizing functional

$$F_{\tilde{p},\beta}(q,\theta) = \beta\sum_{i=1}^{n} \tilde{p}(x_i)\mathbb{E}_{q_i}[\log p(z,x_i|\theta)] + \sum_{i=1}^{n} \tilde{p}(x_i)H(q_i) \tag{4.2.9}$$

with respect to distributions q_i (E step) and parameter vector θ (M step). we can refactor functional F_β in terms of the untempered EM functional $F(q, \theta)$

$$F_{\tilde{p},\beta}(q,\theta) = \beta \sum_{i=1}^{n} \tilde{p}(x_i) \left(\mathbb{E}_{q_i}[\log p(x_i, z|\theta)] + H(q_i)\right) + (1-\beta) \sum_{i=1}^{n} \tilde{p}(x_i) H(q_i) = \beta F_{\tilde{p}}(q,\theta) + (1-\beta) \sum_{i=1}^{n} \tilde{p}(x_i) H(q_i) \quad (4.2.10)$$

At each step, we are identifying probability distributions $q_i(z)$ as approximations of the distribution $p(z|x_i, \theta)$. Thus, taking into account the development of equation (3.1.6) exposed in last chapter we have that the TEM algorithm finds a local maximum of the functional

$$\mathcal{F}_{\tilde{p},\beta} = \beta \mathcal{L}_{\tilde{p}}(\theta) + (1-\beta) \sum_{i=1}^{n} \tilde{p}(x_i) H(p(z|x_i, \theta)) \quad (4.2.11)$$

If we substitute equation (4.2.1), we get

$$\mathcal{F}_{\tilde{p},\beta} = \beta H(\tilde{p}) - \beta \,\mathrm{KLD}\,(\tilde{p} \| p(\cdot|\theta)) + (1-\beta) \sum_{i=1}^{n} \tilde{p}(x_i) H(p(z|x_i, \theta))$$

Dropping the first constant entropy term, we can interpret the expected topic entropy as a regularization term for the EM learning algorithm. EM finds the distribution $p(\cdot|\theta)$ that factors through the complete data model that is (locally) closest to the prior distribution $\tilde{p}$ in the KLD pseudo-metric. TEM seeks for this nearest distribution while maxing the expected entropy of the topic distribution $p(z|x, \theta)$.

4.2.4 Algorithm convergence

The interpretation of the algorithm as a coordinate ascent algorithm proves that each E and M step successively increase the value of functional $F_{\tilde{p},\beta}$. This functional is bounded from above, as $F_{\tilde{p}}(q, \theta) \leq \mathcal{L}_{\tilde{p}}(\theta) \leq H(\tilde{p})$ and $H(q_i) \leq H(u)$ for all i, where u is a uniform distribution over $\mathcal{Z}$. Thus,

$$F_{\tilde{p},\beta}(q,\theta) \leq \beta H(\tilde{p}) + (1-\beta) H(u)$$

Then, $\{F_{\tilde{p},\beta}(q^{(t)}, \theta_t)\}_{t\geq 0}$ is a crescent sequence bounded from above, so it converges. Once $F_{\tilde{p},\beta}(q^{(t)}, \theta_t)$ is close enough to the sequence limit, either $(q^{(t)}, \theta_t)$ converges or it moves describing an $F_{\tilde{p},\beta}$-almost-constant path. From this point, following an analogous argument to the one given by Dempster et al. (1977), a convergence proof based on the curvature of functional $F_{\tilde{p},\beta}$ being bounded could be straightforward. Dempster et al. (1977) curvature computations would held simply changing the frequency parameters to the prior distributions.

As this proof goes beyond the scope of this work, we will present an intuitive idea of why the algorithm should computationally converge and why it should not stay on slowly decaying cycles. From Dempster et al. (1977) proof, we know that classic EM converges for any starting set of parameters. In particular, we know that the

algorithm convergence for any observed samples, and thus, for any frequency vectors $n(x)$ over the observed data $\overline{X}$. Executing EM algorithm with frequency vectors $n(x)$ is equivalent to executing the utility-biased algorithm with prior $\tilde{p}(x) = \frac{n(x)}{\sum_{x \in \overline{X}} n(x)}$.

We know that the EM algorithm converges for any choice of the frequency vectors $n(x)$, so, in particular, we know that the utility-biased EM converges for any $\tilde{p}$ with rational values, i.e., such that $\tilde{p}(x) \in \mathbb{Q}$ for all $x \in \overline{X}$. As any real number is computationally represented as a rational number, utility-biased EM can always be simulated in a computer as an EM execution with the adequate choice of frequency vectors. Therefore, the algorithm will computationally converge for any choice of $\tilde{p}$.

Experimental results on the evolution of the Kullback-Leibler divergence between the prior and the estimated distribution in search diversity tasks show that the algorithm converges at a similar rate than classic EM.

4.2.5 Applications to search diversity

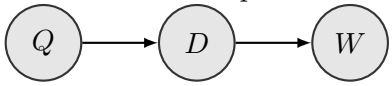
Deepening in the initial example, we will describe several models able to incorporate query-specific document relevance to the pLSA algorithm. We will work in the theoretical utility-biased likelihood framework described in the previous sections, using the explicit utility-biased EM algorithm 4.1.

Let $\mathcal{Q}$ be the set of queries in the corpus. The proposed models are be based in extending the word-document base space of pLSA to a latent model over $\mathcal{W} \times \mathcal{D} \times \mathcal{Q}$. We will consider that the observed data consist of triples (w, d, q), where the word w belongs to document d and document d is retrieved by the baseline search system for the query q. We can consider that d is retrieved by q if the baseline score is higher than a threshold o simply if it belongs to the top ranking for a certain fixed cut. In our experiments we will select documents that appear in the top 100 results for each query. The set of retrieved documents for the query q will be denoted by $\mathcal{D}_q$.

Observed triples (w, d, q) are enhanced with a priori utility distribution $\tilde{p}$ as follows. In pLSA, distribution $p(w, d)$ corresponds to the probability of observing the pair (w, d) in the corpus. In our case, distribution $p(w, d, q)$ would be interpreted as the probability of observing the pair (w, d) and document d being relevant for query q. Using the baseline search system information, we can approximate a prior distribution $\tilde{p}(w, d, q)$.

If the baseline provides a full language model, the desired distribution may be obtained directly and take into account relevance of word w for query q apart form relevance of document d, resulting in the distribution being more informative about the query semantic information need. Nevertheless, we will consider the search system as a black box and use only baseline rankings for each query. In this case, we will use the usual independence hypothesis of words being independent from the query once the document is observed.

Figure 4.1: Prior incomplete data model



Thus, we approximate

$$\tilde{p}(w,d,q) = \tilde{p}(w|d)\tilde{p}(d|q)\tilde{p}(q)$$

Distribution $\tilde{p}(w|d)$ can be estimated from word count in the classical way

$$\tilde{p}(w|d) = \frac{n(w,d)}{\sum_{w\in\mathcal{W}} n(w,d)}$$

Jelinek-Mercer, Dirichlet or any other kind of smoothing would be possible, but incidence matrix sparsity is crucial to the algorithm efficiency both in memory usage and execution time, as the observed incidence population is several orders of magnitude smaller than $|\mathcal{W}| \cdot |\mathcal{D}|$. We will take distribution $\tilde{p}(q)$ simply as uniform over $\mathcal{Q}$. A priori there is no reason to treat information from different queries as more or less valuable, but an utility functional can be used instead if we want to incorporate parameters like query ambiguity or difficulty to the model.

Finally, we can estimate distribution $\tilde{p}(d|q)$ either form baseline score or from ranking position. As before, further information about probabilistic meaning of the baseline score (for example, knowing that it corresponds to a PRP system or as the result of a document likelihood language model) would lead to more informative distribution, but if we only consider ranking information, it is natural to consider the distribution as proportional to a discount functional on the ranking position. Let $\tau(d,q)$ be the position (starting in 0) of document d in the baseline ranking of query q, and let $s : \mathbb{N} \to \mathbb{R}^+$ a discount function. We estimate

$$\tilde{p}(d|q) \sim \frac{s(\tau(d,q))}{\sum_{d\in\mathcal{D}_q} s(\tau(d,q))}$$

Different discount functions can be used in order to regulate the impact of the low rank documents in the algorithm. In our search diversity task, we are not interested in penalizing too much documents with mid-low ranking, as they might have useful diverse information that might be promoted by the diversification algorithm, but a non-incisive discount is yet recommended to be applied. We propose to use a simple linear discount

$$s(\tau) = 1 - \frac{\tau}{|\mathcal{D}_q|}$$

Moreover, a Jelinek-Mercer smoothing can be applied, resulting in

$$\tilde{p}(d|q) \sim \lambda\frac{s(\tau(d,q))}{\sum_{d\in\mathcal{D}_q} s(\tau(d,q))} + (1-\lambda)\frac{1}{|\mathcal{D}_q|}$$

Once the incomplete data model is set, we need to extend pLSA latent variable model to include the new query variable Q. Properties of the latent space will strongly depend on the generative model used. The closer distribution $\tilde{p}$ is to a distribution $p(w,d,q)$ representable in the selected data model, the better the latent distributions will be able to fit the data. Of course, as in every machine learning algorithm, more complex models allow to fit better the data, but at the cost of a bigger parameter space, and thus, more chances of overfitting.

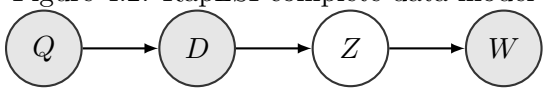
As an example, we will provide a model that aims to extend Hofmann's probabilistic Latent Semantic Indexing algorithm (Hofmann, 1999b) so that it takes

document relevance into account. We will consider the following utility-biased generative model for the enhanced observed data (i.e., the observed triples (w, d, q))

- Choose a random query $q \in \mathcal{Q}$ with probability $\tilde{p}(q)$ (usually $\mathrm{Unif}_{\mathcal{Q}}(q)$)
- Select a document $d \in \mathcal{D}_q$ with probability $\tilde{p}(d|q)$
- Pick a latent semantic class $z \in \mathcal{Z}$ with probability $p(z|d)$
- Generate a word $w \in \mathcal{W}$ with probability $p(w|z)$

corresponding to the following complete data Bayesian network

Figure 4.2: RapLSI complete data model



As the aspects and words are independent to the query once the document is selected and none of the parameters of the model depend on the query, we can collapse that variable in the model marginalizing and get exactly the same utility-based MLE for the parameters. Thus, we arrive to the asymmetrical graphical model of the original pLSA (3.3), in which the prior distribution for documents $p(d)$ has been replaced by

$$\tilde{p}(d) = \sum_{q \in \mathcal{Q}} \tilde{p}(d|q)\tilde{p}(q)$$

where $\tilde{p}(d|q) = 0$ if $d \notin \mathcal{D}_q$. As in pLSA, we can reparametrize the problem in the symmetric form (3.4). Substituting step E by the tempered version (4.2.8) in algorithm 4.1 leads to the following explicit algorithm for the selected Bayesian network

Algorithm 4.3 Relevance aware pLSI

procedure RapLSI($\tilde{p}(w,d)$,$\overline{X} = \{(w,d)\}$,β)
  Start with a random distributions $p(w|z), p(d|z), p(z)$
  for each t **do**
    Step E:
    for each $(w,d) \in \overline{X}$ and every $z \in \mathcal{Z}$ **do**

$$p(z|w,d) = \frac{(p(d|z)p(w|z)p(z))^{\beta}}{\sum_{z\in\mathcal{Z}} (p(d|z)p(w|z)p(z))^{\beta}}$$

    end for
    Step M:
    for each $z \in \mathcal{Z}$, $d \in \mathcal{D}$ and $w \in \mathcal{W}$ **do**

$$p(d|z) = \frac{\sum_{(w',d)\in\overline{X}} p(z|w',d)\tilde{p}(w',d)}{\sum_{(w',d')\in\overline{X}} p(z|w',d')\tilde{p}(w',d')}$$

$$p(w|z) = \frac{\sum_{(w,d')\in\overline{X}} p(z|w,d')\tilde{p}(w,d')}{\sum_{(w',d')\in\overline{X}} p(z|w',d')\tilde{p}(w',d')}$$

$$p(z) = \frac{\sum_{(w',d')\in\overline{X}} p(z|w',d')\tilde{p}(w',d')}{\sum_{z\in\mathcal{Z}}\sum_{(w',d')\in\overline{X}} p(z|w',d')\tilde{p}(w',d')}$$

    end for
  end for
  The algorithm stops either after a maximum number of iterations or when distributions $p(d|z), p(w|z), p(z)$ have converged
end procedure

Once the algorithm has converged, distributions $p(z|q)$ can be computed through Bayesian fold in from the computed parameters, for example, taking $p(z|q) = \sum_{d\in\mathcal{D}_q} p(z|d)\tilde{p}(d|q)$, where, applying Bayes theorem, $p(z|d) = \frac{p(d|z)p(z)}{\sum_{z\in\mathcal{Z}} p(d|z)p(z)}$. The observed data distribution approximation is given by

$$p(w,d) = \sum_{z\in\mathcal{Z}} p(w|z)p(d|z)p(z)$$

The algorithm converges to a local maximum of the tempered functional (4.2.11), i.e., it maximizes locally

$$\mathcal{F}_{\tilde{p},\beta} = \beta H(\tilde{p}) - \beta\,\mathrm{KLD}\,(\tilde{p}\|p) + (1-\beta)\sum_{(w,d)\in\overline{X}} \tilde{p}(w,d)H(p(z|w,d)) \qquad (4.2.12)$$

Dropping the entropy regularization and focusing on the main term (i.e., taking $\beta = 1$), we can analyze the divergence term, factorizing it trough the incomplete

data model as

$$\mathcal{L}_{\tilde{p}} = \sum_{w\in\mathcal{W}}\sum_{d\in\mathcal{D}} \tilde{p}(w,d)\log p(w,d) = \sum_{w\in\mathcal{W}}\sum_{d\in\mathcal{D}} \tilde{p}(w|d)\tilde{p}(d)\left(\log p(w|d) + \log p(d)\right) =$$
$$\sum_{d\in\mathcal{D}} \tilde{p}(d)\log p(d) + \sum_{d\in\mathcal{D}} \tilde{p}(d)\sum_{w\in\mathcal{W}} \tilde{p}(w|d)\log p(w|d) = H(\tilde{p}(d)) - \mathrm{KLD}\left(\tilde{p}(d)\|p(d)\right) +$$
$$\sum_{d\in\mathcal{D}} \tilde{p}(d)\left(H(\tilde{p}(w|d)) - \mathrm{KLD}\left(\tilde{p}(w|d)\|p(w|d)\right)\right)$$

Dropping the constant entropy of the priori terms, we obtain that RapLSA minimizes

$$\mathrm{KLD}\left(\tilde{p}(d)\|p(d)\right) + \mathbb{E}_{\tilde{p}(d)}\left[\mathrm{KLD}\left(\tilde{p}(w|d)\|p(w|d)\right)\right] \tag{4.2.13}$$

In the incomplete data model, $\tilde{p}(w,d)$ splits in $\tilde{p}(w|d)$ and $\tilde{p}(d)$. Distributions $\tilde{p}(w|d)$ contain the semantic information of the data, while priori $\tilde{p}(d)$ contain the relevance information. Looking at equation (4.2.13) we observe that RapLSA minimizes a combination of two divergences. Minimizing the first term, $\mathrm{KLD}\left(\tilde{p}(d)\|p(d)\right)$ makes the algorithm learn the relevance information of the baseline, $\tilde{p}(d)$. The second term is a weighted mean of divergences between prior word distributions $\tilde{p}(w|d)$ and estimated distributions $p(w|d)$, using $\tilde{p}(d)$ as coefficients. By minimizing the second term, the algorithm learns the term distribution of all the documents in the corpus, prioritizing those with higher baseline relevance $\tilde{p}(d)$.

Taking $0 < \beta < 1$ keeps this behavior, but introduces an entropic regularization term for the aspect distributions $p(z|w,d)$, leading to more "general" aspect distributions in the sense explained in the last section. It is worth noticing that taking $\tilde{p}(d) = \tilde{p}_{pLSI}(d) := \frac{\sum_{w\in\mathcal{W}} n(w,d)}{\sum_{w\in\mathcal{W},d'\in\mathcal{D}} n(w,d')}$ leads to Hofmann's pLSI algorithm. This allows us to analyze theoretically the difference in the objective functional between classic pLSI and the proposed RapLSI.

Equation (4.2.13) hold for pLSI setting$\tilde{p} = \tilde{p}_{pLSI}$. Therefore, Hofmann's pLSI minimizes a combination of $\mathrm{KLD}\left(\tilde{p}_{pLSI}(d)\|p(d)\right)$ and a weighted combination of divergences $\mathrm{KLD}\left(\tilde{p}(w|d)\|p(w|d)\right)$ for each document. In RapLSI, the first divergence effectively captures the relevance information provided by priori $\tilde{p}$, but for classic pLSI, distribution $\tilde{p}_{pLSI}$ is just proportional to the length of the document, so $pLSI$ "learns" document length. On the other hand, document distribution $\tilde{p}(d)$ sets the weight coefficient for every document divergence $\mathrm{KLD}\left(\tilde{p}(w|d)\|p(w|d)\right)$. In case of using classic pLSI, the algorithm gives more importance to learning the term distribution of the largest documents.

Taking these two properties into account, theoretically, the RapLSI resulting aspect space would cover the same abstraction needs of pLSI (synonym detection, polysemy disambiguation, etc.), but, depending on the $\tilde{p}$ distribution choice, it is supposed to become more robust than classic pLSI when dealing with noise and spam in samples.

Now, we will focus on the task of building intent spaces optimized for their use by diversification algorithms. As before, we will consider incomplete data models over $\mathcal{W}\times\mathcal{D}\times\mathcal{Q}$ with the same prior distribution $\tilde{p}(w,d,q) = \tilde{p}(w|d)\tilde{p}(d|q)\tilde{p}(q)$. Term $\tilde{p}(w|d)$ in the priori equation contains the semantic information of the data, while the term $\tilde{p}(d|q)$ hold the pseudo-relevance feedback information. As the Bayesian

abstract model minimizes the KLD between the priori and the estimated distribution of $\tilde{p}(w, d, q)$, the resultant EM balances the influence of the semantic and relevance information in a way that maximizes the likelihood of the a priori knowledge of the data.

This equilibrium fits perfectly with the problem of IR diversification, in which the mixture of pure diversity and relevance is critic. The semantic information acts as a pure diversity source. The relevance part of the priori modulates the aspect distributions of each document, promoting the election of high-rated documents among those which share a certain aspect. The algorithm controls the equilibrium while extracting the aspects, so it can lead to a more precise treatment of the relevance for each specific document.

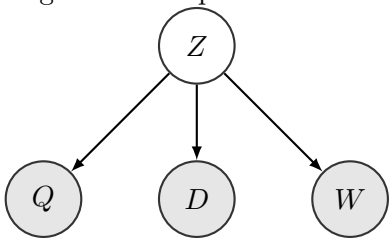
While the previous graphical model (4.2) appears to be natural for the indexing task and it can indeed be used to build an intent space for diversification by means of folding-in techniques, the resulting aspect space parametrizes global semantic categories, similar to the ones provided by ODP. The algorithm acts essentially as a clustering algorithm in the KLD pseudo-metric, so the size of the aspect space K modules the "depth" of the categories, in the same way that we can use different levels of categories in the ODP classification.

Instead, we want to extract query-specific intent spaces that are able to distinguish and isolate any subtle aspect of the possible information needs behind the query expression. In order to do so, we need to use a complete data model which, at least, makes the aspect variable depend on the query.

Several generative models may fit our needs

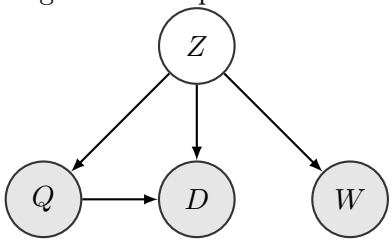
1. **Symmetric model:** We select an information need $z \in \mathcal{Z}$ from distribution $p(z)$ and then choose $w \in \mathcal{W}$, $d \in \mathcal{D}$ and $q \in \mathcal{Q}$ from distributions $p(w|z)$, $p(d|z)$ and $p(q|z)$ respectively. These three distributions are interpreted as parametrizing probability of taking a word expressing the aspect z, a document covering the aspect z or a query that express the information need z respectively. Words, documents and queries are considered mutually independent given an aspect.

Figure 4.3: RapLSA model 1



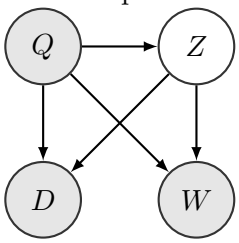
2. **Symmetric linked:** Similar to model 1, but document selection is supposed to depend both on the chosen aspect and query, thus linking explicitly latent semantics and relevance information in a single parameter distribution.

Figure 4.4: RapLSA model 2



3. **Space of pairs** (z, q)**:** Documents and words are supposed to be independent given a pair (z, q). Aspects are selected for each query from a probability distribution $p(z|q)$. Documents and words are drawn from distributions $p(d|z, q)$ and $p(w|z, q)$ respectively, depending on the selected pair (z, q) but independent from each other.

Figure 4.5: RapLSA model 3



Experimental results show that model 3 performs significantly better than the others under diversity metrics benchmarking. As explicit E and M steps equations are easily derived from each graphical model using pseudo-code 4.1, we will only analyze model 3 equations and its differences from classic pLSA algorithm. In particular, the explicit algorithm runs as follows

Algorithm 4.4 Relevance aware pLSA model 3

procedure RapLSA($\tilde{p}(w,d,q)$,$\overline{X} = \{(w,d,q)\}$,$\beta$)
  Start with a random distributions $p(w|z,q), p(d|z,q), p(z|q)$
  for each t **do**
    Step E:
    for each $(w,d,q) \in \overline{X}$ and every $z \in \mathcal{Z}$ **do**

$$p(z|w,d,q) = \frac{(p(d|z,q)p(w|z,q)p(z|q)\tilde{p}(q))^{\beta}}{\sum_{z\in\mathcal{Z}}(p(d|z,q)p(w|z,q)p(z|q)\tilde{p}(q))^{\beta}}$$

    end for
    Step M:
    for each $z \in \mathcal{Z}$, $q \in \mathcal{Q}$, $d \in \mathcal{D}_q$ and $w \in \mathcal{W}$ **do**

$$p(d|z,q) = \frac{\sum_{(w',d,q)\in\overline{X}} p(z|w',d,q)\tilde{p}(w',d,q)}{\sum_{(w',d',q)\in\overline{X}} p(z|w',d',q)\tilde{p}(w',d',q)}$$

$$p(w|z,q) = \frac{\sum_{(w,d',q)\in\overline{X}} p(z|w,d',q)\tilde{p}(w,d',q)}{\sum_{(w',d',q)\in\overline{X}} p(z|w',d',q)\tilde{p}(w',d',q)}$$

$$p(z|q) = \frac{\sum_{(w',d',q)\in\overline{X}} p(z|w',d',q)\tilde{p}(w',d',q)}{\sum_{z\in\mathcal{Z}}\sum_{(w',d',q)\in\overline{X}} p(z|w',d',q)\tilde{p}(w',d',q)}$$

    end for
  end for
  The algorithm stops either after a maximum number of iterations or when distributions $p(d|z,q), p(w|z,q), p(z|q)$ have converged
end procedure

If we take, $\tilde{p}(q) \sim \text{Unif}_{\mathcal{Q}}(q)$, as $\tilde{p}(w,d,q) = \tilde{p}(w,d|q)\tilde{p}(q)$, $\tilde{p}(q)$ factors cancel in each fraction both in E and M steps in algorithm 4.4. Therefore, in this case, model 3 is equivalent to conditioning RapLSI model by q for each $q \in \mathcal{Q}$ in terms of the generative model and the complete data model. This would be therefore equivalent to executing a RapLSI instance for each query independently, using $\tilde{p}(d|q)$ as document prior for each query.

Therefore, the algorithm minimizes the combination of divergences

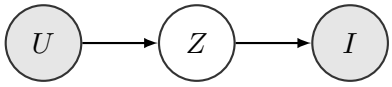
$$\text{KLD}\left(\tilde{p}(d|q)\|p(d|q)\right) + \mathbb{E}_{\tilde{p}(d|q)}[\text{KLD}\left(\tilde{p}(w|d)\|p(w|d,q)\right)]$$

4.2.6 Applications to recommenders diversity

Recommendation tasks can be set in the framework of a generative model similar to the one used by pLSA for search tasks. In terms of information recovery, user profile would act as an implicit query and items would take the same roll of documents. Nevertheless, unless a set of item content features is available, thus enabling content-based recommendation strategies, in recommendation tasks there is no clear information token taking the same role as document words.

Instead, Hofmann (2004) uses items as implicit user features. Let $\mathcal{U}$ be the set of users and $\mathcal{I}$ be the set of items. Let us consider that a set of user-item pairs is observed $\overline{X} = \{(u, i)\}$. Let $n(u, i)$ denote the number of observed instances of the pair (u, i) in the corpus. Depending on the recommendation scenario, $n(u, i)$ may correspond to a number of times user u "accesses" or "uses" item i, evaluated by click counts, reproductions or other methods. Then, users take a similar role to the one documents had in the search framework and items observed for each user take the roll of words within a document. A latent semantic model is then assumed for the set of observed pairs (u, i), $u \in \mathcal{U}$, $i \in \mathcal{I}$.

Figure 4.6: Latent model for recommendation



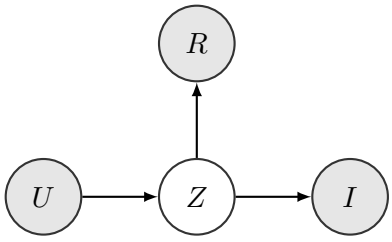
Hofmann (2004) applies EM algorithm to get distributions $p(z|u)$ and $p(i|z)$ that maximize the likelihood of the observed pairs

$$\mathcal{L} = \frac{1}{U} \sum_{(u,i)\in\overline{X}} n(u, i) \log p(i|u) = \frac{1}{U} \sum_{(u,i)\in\overline{X}} n(u, i) \log \sum_{z\in\mathcal{Z}} p(i|z)p(z|u)$$

Hofmann also proposes a method for introducing explicit ratings information into a collaborative filtering model (Hofmann, 2003, 2004). Let $\mathcal{R}$ denote the set of possible ratings. Hofmann describes algorithms for treating both discrete and continuous sets of ratings. The latter is parameterized as a Gaussian distribution whose parameters are estimated in step M of the algorithm. In order to focus on the theoretical properties of the model and avoid additional computations, we will restrict our ongoing analysis to a discrete set of ratings.

Hofmann (2004) considers various graphical models that can extend classic pLSA model 4.6 to incorporate the rating variable. As we are interested in the construction of an intent space for diversification purposes, we will use the so called *categorized* model, in which items only impact the prediction through the aspect space $\mathcal{Z}$

Figure 4.7: Latent model for recommendation with ratings



Incidence triples $\overline{X} = \{(u, i, r)\}$ are then used as observed data, and EM algorithm is used to maximize the functional

$$\mathcal{L} = \frac{1}{U} \sum_{(u,i,r)\in\overline{X}} \log \sum_{z\in\mathcal{Z}} p(i|z)p(r|z)p(r|u)$$

We propose to use an utility-biased version of this algorithm, in the framework of the general procedure previously described, using either explicit or estimated user ratings as a source of relevance for both the observed pairs (u, i) and triplets (r, u, i).

Let us assume that we have a baseline recommender system that allows us to estimate a relevance score $s : \mathcal{U} \times \mathcal{I} \to \mathbb{R}^+$. The main assumption of our model is that we can estimate the relevance of an observation (which may not correspond exactly to the item rating) from the item relevance score. In a first approximation, we will assume that the higher an item has been rated, the more informative it is for the user profile. This would essentially correspond to a smoothed positive feedback personalization strategy, in which information from non-relevant elements is discarded (or, in this case, given a much lower impact on the profile estimation). Normalizing the score, we obtain a normalized utility function which can be interpreted as a probability of observing a user-item pair, assuming that more relevant items are more likely to be drawn for each user.

$$\tilde{p}(u, i) \sim \frac{s(u, i)}{\sum_{u \in \mathcal{U}, i \in \mathcal{I}} s(u, i)}$$

If an explicit set of ratings is considered in the model, for each observed pair $(u.i)$, a prior distribution can be estimated over the rating values, $\tilde{p}(r|u, i)$. As a simple approximation, if multiple ratings are available for a single pair (u, i), the distribution can be taken simply as proportional to the observed frequency. Usually, this is not the case, and users only rate items once. Then, triples (r, u, i) are in bijection to observed pairs (u, i) and we can build a function $r : \overline{X} \subseteq \mathcal{U} \times \mathcal{I} \to \mathcal{R}$ identifying the observed rating of each pair. Therefore, as a particular case, we can take

$$\tilde{p}(r|u, i) \sim \begin{cases} 1 & r = r(u, i) \\ 0 & r \neq r(u, i) \end{cases}$$

As usual, if more baseline information is available (such as a discovery-rating model or prior information about user or item biases), it can be incorporated to single utility functions $\tilde{p}(u, i)$ or $\tilde{p}(r, u, i)$ depending on ratings being considered or not in the observed data model.

Once the enhanced observed data is modeled, complete data models are described from pLSA generative models 4.6 and 4.7 using a symmetric parameterization. Both observed and complete data models together allow us to apply utility-biased EM 4.1, resulting in the following explicit algorithms

Algorithm 4.5 Relevance aware pLSA recommendation model

procedure RapLSA($\tilde{p}(u,i)$,$\overline{X} = \{(u,i)\}$,β)
  Start with a random distributions $p(u|z), p(i|z), p(z)$
  for each t **do**
    Step E:
    for each $(u,i) \in \overline{X}$ and every $z \in \mathcal{Z}$ **do**

$$p(z|u,i) = \frac{(p(u|z)p(i|z)p(z))^{\beta}}{\sum_{z \in \mathcal{Z}} (p(u|z)p(i|z)p(z))^{\beta}}$$

    end for
    Step M:
    for each $z \in \mathcal{Z}$, $u \in \mathcal{U}$ and $i \in \mathcal{I}$ **do**

$$p(u|z) = \frac{\sum_{(u',i) \in \overline{X}} p(z|u',i)\tilde{p}(u',i)}{\sum_{(u',i') \in \overline{X}} p(z|u',i')\tilde{p}(u',i')}$$

$$p(i|z) = \frac{\sum_{(u,i') \in \overline{X}} p(z|u,i')\tilde{p}(u,i')}{\sum_{(u',i') \in \overline{X}} p(z|u',i')\tilde{p}(u',i')}$$

$$p(z|q) = \frac{\sum_{(u',i') \in \overline{X}} p(z|u',i')\tilde{p}(u',i')}{\sum_{z \in \mathcal{Z}} \sum_{(u',i') \in \overline{X}} p(z|u',i')\tilde{p}(u',i')}$$

    end for
  end for
  The algorithm stops either after a maximum number of iterations or when distributions $p(u|z), p(i|z), p(z)$ have converged
end procedure

Utility-biased likelihood equation (4.2.1) imply that the untempered algorithm minimizes locally KLD $(\tilde{p}(u,i) \| p(u,i))$. Therefore, the algorithm effectively learns the baseline relevance information provided to the system and builds aspect spaces that prioritize approximating correctly the most relevant items for each user.

Classic tempered pLSA model can be recovered from algorithm 4.5 simply taking

$$\tilde{p}(u,i) \sim \frac{n(u,i)}{\sum_{u \in \mathcal{U}, i \in \mathcal{I}} n(u,i)}$$

Therefore, the proposed method generalizes Hofmann's pLSA.

Algorithm 4.6 Relevance aware pLSA recommendation model with explicit ratings

procedure RapLSA($\tilde{p}(u,i)$,$\overline{X} = \{(r,u,i)\}$,$\beta$)
  Start with a random distributions $p(u|z), p(i|z), p(r|z), p(z)$
  for each t **do**
    Step E:
    for each $(r,u,i) \in \overline{X}$ and every $z \in \mathcal{Z}$ **do**

$$p(z|u,i,r) = \frac{(p(u|z)p(i|z)p(r|z)p(z))^{\beta}}{\sum_{z\in\mathcal{Z}} (p(u|z)p(i|z)p(r|z)p(z))^{\beta}}$$

    end for
    Step M:
    for each $z \in \mathcal{Z}$, $u \in \mathcal{U}$ and $i \in \mathcal{I}$ **do**

$$p(u|z) = \frac{\sum_{(r',u',i)\in\overline{X}} p(z|r',u',i)\tilde{p}(r',u',i)}{\sum_{(r',u',i')\in\overline{X}} p(z|r',u',i')\tilde{p}(r',u',i')}$$

$$p(i|z) = \frac{\sum_{(r',u,i')\in\overline{X}} p(z|r',u,i')\tilde{p}(r',u,i')}{\sum_{(r',u',i')\in\overline{X}} p(z|r',u',i')\tilde{p}(r',u',i')}$$

$$p(r|z) = \frac{\sum_{(r,u',i')\in\overline{X}} p(z|r,u',i')\tilde{p}(r,u',i')}{\sum_{(r',u',i')\in\overline{X}} p(z|r',u',i')\tilde{p}(r',u',i')}$$

$$p(z|q) = \frac{\sum_{(r',u',i')\in\overline{X}} p(z|r',u',i')\tilde{p}(r',u',i')}{\sum_{z\in\mathcal{Z}}\sum_{(r',u',i')\in\overline{X}} p(z|r',u',i')\tilde{p}(r',u',i')}$$

    end for
  end for
  The algorithm stops either after a maximum number of iterations or when distributions $p(u|z), p(i|z), p(r|z), p(z)$ have converged
end procedure

If ratings are unique for every observed pair (u,i) step M equations can be slightly simplified, as summation over $\mathcal{R}$ is no longer necessary (there is only one factor).

Either case, analogous computations to the ones done for RapLSA prove that the untempered algorithm minimizes locally

$$\mathrm{KLD}\left(\tilde{p}(u,i)\|p(u,i)\right) + \sum_{(u,i)\in\overline{X}} \tilde{p}(u,i)\,\mathrm{KLD}\left(\tilde{p}(r|u,i)\|p(r|u,i)\right)$$

The first factor corresponds exactly to the same divergence between the relevance prior $\tilde{p}(u,i)$ and the estimated distribution $p(u,i)$, minimized by the previous version. The second factor corresponds to a weighted mean of the divergences between the real rating distribution $\tilde{p}(r|u,i)$ and the predicted rating $p(r|u,i)$, using the prior utility distribution values as mixing factors.

Thus, the algorithm learns the relevance information and, at the same time, it learns the explicit ratings registered for the observed pairs user-item, prioritizing

learning items with higher relevance.

While it may seem redundant to enhance the rating observations (r, u, i) with an estimated relevance distribution $\tilde{p}(r, u, i)$ obtained from the rating itself, we have to notice that categorical ratings $\mathcal{R}$ don't hold the same topological information as the distribution $\tilde{p}(r, u, i)$. Even if we used a continuous set of ratings that held the original rating order, classic pLSA does never allow rating values to be transfered to the EM dynamics. Nevertheless, previous divergence computations prove that the proposed new models intrinsically learn this relevance information, therefore adapting successfully the original algorithm to the desired smoothed positive feedback strategy.

Finally, we notice that depending on the baseline system, there might exist user an item biases that would get immediately propagated to the EM dynamics. In particular, the profiles of users (or items) with overall higher ratings would be analyzed through observed pairs (u, i) with high relevance $\tilde{p}(u, i)$. Therefore, learning its profile would become a priority for the system, over learning profiles with lower ratings. There are several ways of mitigating this effect. The first and simplest one would consist in normalizing the ratings/scores before applying the algorithm. As the explicit rating value doesn't affect the dynamics, categorical ratings may be left unmodified, but relevance scores are highly recommended to be normalized among users/items.

On the other hand, a model driven strategy might be applied. Taking into account that

$$\mathrm{KLD}\left(\tilde{p}(u,i)\|p(u,i)\right) = \mathrm{KLD}\left(\tilde{p}(u)\|p(u)\right) + \sum_{u\in\mathcal{U}} \mathrm{KLD}\left(\tilde{p}(i|u)\|p(i|u)\right) = \\ \mathrm{KLD}\left(\tilde{p}(i)\|p(i)\right) + \sum_{i\in\mathcal{I}} \mathrm{KLD}\left(\tilde{p}(u|i)\|p(u|i)\right)$$

we can compensate any relevance bias by perturbing user or item priors, therefore absorbing any existing biases in the selected prior.

4.2.7 Further applications

General algorithm 4.1 can be applied to any pair consisting of an utility-enhanced observed data and a complete latent variable model in the form of a Bayesian network, for which unknown distributions are assumed to be categorical (i.e., whose parameters correspond to categorical distributions).

Generative models described in the last sections represent just a few examples of application of the general theoretical framework. As far as a suitable generative model can be described for the complete data of an information retrieval problem and a proper utility distribution can be inferred for the observed data that reflects the relevance of the observation, the algorithm allows us to build automatically intent spaces that priorize learning of the more relevant pieces of information, while, at the same time, learning the utility distribution itself.

In this section we will describe some information retrieval tasks for which these models can be theoretically described and thus, for which RapLSA algorithm can be applied. Once prior distribution and complete data model are fixed, the explicit

form of E and M equations comes straightforward from direct computation in general algorithm 4.1, so some of them will be omitted. Tempering is also available for all models, applying equation (4.2.8).

Personalized search

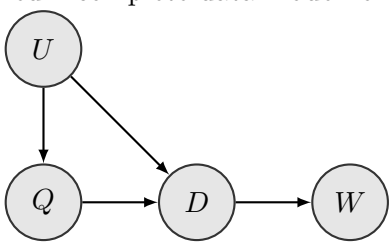
RapLSA models describe in the las section transfer almost directly to the personalization scenario by introducing a random variable over the set of users $\mathcal{U}$ acting similar to the query. We will assume that we have a baseline user profile source and we will distinguish two scenarios depending on the baseline being sensible or not to the user profile.

In the first simplest one, we will suppose that our baseline search engine is already personalized, so that retrieved documents for each query q also depend on the user u. Denoting by $\mathcal{D}_{q,u}$ the set of retrieved documents (top 100 retrieved, for example, just as in RapLSI), we can build a probability distribution $\tilde{p}(d|q,u)$ over $\mathcal{D}_{q,u}$. Similarly to RapLSI, this distribution can be estimated precisely from the baseline score if suitable, or it can be estimated from the ranking $\tau(d,q,u)$ using a discount function

$$\tilde{p}(d|q,u) \sim \frac{s(\tau(d,q,u))}{\sum_{d\in\mathcal{D}_{q.u}} s(\tau(d,q,u))}$$

Then, we can use this incomplete data model

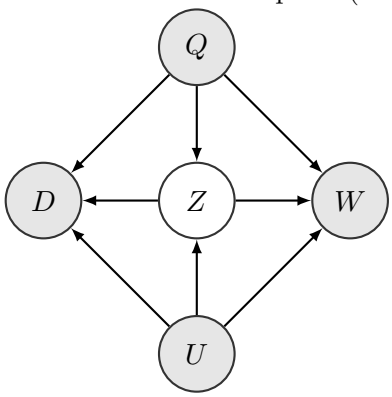
Figure 4.8: Personalized incomplete data model for RapLSA (scenario I)



to estimate $\tilde{p}(w,d,q,u) = \tilde{p}(w|d)\tilde{p}(d|q,u)\tilde{p}(q,u)$ for some, typically uniform distribution $\tilde{p}(q,u)$.

Then, we can use the following graphical model for the complete data with coupled (z,q,u) triples

Figure 4.9: Personalized RapLSA(model 3)



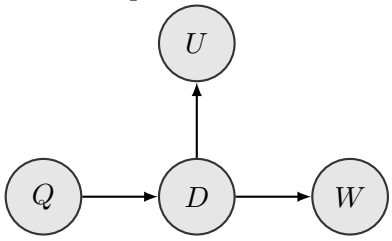
As with RapLSA model 3 (4.5), the resulting algorithm will be equivalent to executing RapLSI for each query and user, taking $\tilde{p}(d|q,u)$ as document priori. This version of the algorithm would minimize, for each user and query,

$$\mathrm{KLD}\left(\tilde{p}(d|q,u)\|p(d|q,u)\right) + \mathbb{E}_{\tilde{p}(d|q,u)}[\mathrm{KLD}\left(\tilde{p}(w|d)\|p(w|d,q,u)\right)]$$

so it will learn personal document relevance for each query (the provided personalized relevance information) at the same time as the term distribution of each document, prioritizing those expected to be more relevant to the user for the query.

The second and more complex scenario would correspond to having ad-hoc baseline separated from personal profiles. In this case, we can use the following simplified version of model 4.8, in which personal information is (initially) supposed to be independent of the query given a document (i.e., personal bias is content-related, independently of the query).

Figure 4.10: Personalized incomplete data model for RapLSA (scenario II)



to estimate $\tilde{p}(w,d,q,u) = \tilde{p}(w|d)\tilde{p}(u|d)\tilde{p}(d|q)\tilde{p}(q)$ for some, typically uniform distribution $\tilde{p}(q)$. $\tilde{p}(u|d)$ can be estimated using Bayes theorem from $\tilde{p}(d|u)$, assuming a known prior distribution $\tilde{p}(u)$ over $\mathcal{U}$ (for example, $\tilde{p}(u) \sim \mathrm{Unif}_{\mathcal{U}}(u)$) taking

$$\tilde{p}(u|d) = \frac{\tilde{p}(d|u)\tilde{p}(u)}{\sum_{d\in\mathcal{D}} \tilde{p}(d|u)\tilde{p}(u)}$$

where $\tilde{p}(d|u)$ can be easily estimated directly from user profile data depending on the explicit kind of profile available (document history, tags, personalized language model, etc.). In particular, if $\tilde{p}(u)$ is uniform, we get $\tilde{p}(u|d) = \tilde{p}(d|u)$. Previous complete data model (4.9) can still be applied.

If $\tilde{p}(q)$ is taken as uniform, analogous computations to the ones done for RapLSA model 3 show that distribution $\tilde{p}(q)$ cancels in E and M step equations. Priori and estimated distributions for the incomplete data model factors as

$$\tilde{p}(w, d, u|q) = \tilde{p}(w|d)\tilde{p}(d|q)\tilde{p}(u|d)$$

$$p(w|d, q, u) = p(w|d, q, u)p(d|q, u)\tilde{p}(u)$$

Thus, for each query $q \in \mathcal{Q}$, the (untempered) algorithm minimizes

$$\mathrm{KLD}\left(\tilde{p}(d, u|q)\|p(d, u|q)\right) + \mathbb{E}_{\tilde{p}(d,u|q)}[\mathrm{KLD}\left(\tilde{p}(w|d)\|p(w|d, q, u)\right)]$$

or, equivalently, it minimizes

$$\mathbb{E}_{\tilde{p}(u|q)}[\mathrm{KLD}\left(\tilde{p}(d|q, u)\|p(d|q, u)\right)] + \mathbb{E}_{\tilde{p}(d,u|q)}[\mathrm{KLD}\left(\tilde{p}(w|d)\|p(w|d, q, u)\right)]$$

Therefore, the algorithm learns both personalized relevance (distributions $\tilde{p}(d|q, u)$) and term distributions $\tilde{p}(w|d)$, priorizing learning the profile of users that are more likely to find information provided by query q relevant, i.e., such that relevant documents for query q are overall considered relevant for user u (high $\tilde{p}(u|q) = \sum_{d\in\mathcal{D}_q} \tilde{p}(u|d)\tilde{p}(d|q)$).

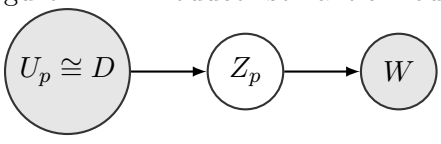
Producer-consumer search model

General methodology behind utility-biased EM can be applied additively when the utility-biased likelihood of more than one set of observed variables is considered. We will use this to obtain a RapLSA-like model that takes into account all information available for the system at the same time: baseline relevance scores, term frequencies for each document and a query language model (which, in the simplest scenario, would correspond to smoothed query terms).

The basis of the model is to decouple the generative model of the *production* of a document from the point of view of its creator (or producer user) from the generative model describing how (consumer) users *consume* the information, in terms of how they form a query expressing an abstract need of information. Both producer and consumer generative models are describe within the pLSA latent semantic framework.

Producer user generates documents expressing a series of abstract ideas/topics $\mathcal{Z}$. For each topic, a certain word distribution $p(w|z_p)$ from a universal vocabulary $\mathcal{W}$ is used to select the literal expression of those ideas. Identifying each document with its producer user, we get classic pLSI generative model

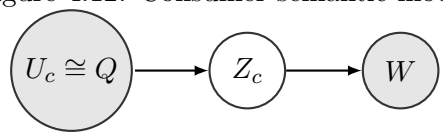
Figure 4.11: Producer semantic model



for which we observe the final term distribution for each document, $\tilde{p}(w|d)$.

On the other hand, consumer user wants to search for a set of abstract information tokens/topics $\mathcal{Z}_c$, summarized selecting a few words in a query from a distribution $p(w|z_c)$. Identifying each query to its consumer user, we get

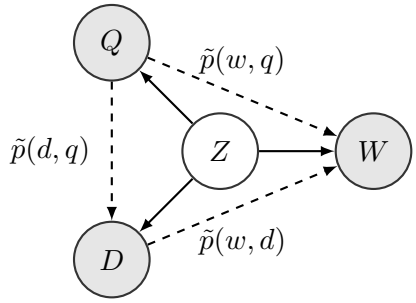
Figure 4.12: Consumer semantic model



for which we observe (or estimate from a query language model) the final query term distribution $\tilde{p}(w|q)$.

The main assumption of the producer-consumer model is that aspects $\mathcal{Z}_c$ and $\mathcal{Z}_q$ both lie within a big universal latent topic space $\mathcal{Z}$. Aspect distributions over $\mathcal{Z}_c$ and $\mathcal{Z}_q$ can be understood as distributions over $\mathcal{Z}$ supported over $\mathcal{Z}_c$ and $\mathcal{Z}_q$ respectively, so both parts of generative model are linked by the common aspect space $\mathcal{Z}$. Moreover, relation between $\mathcal{U}_p$ and $\mathcal{U}_c$, identified with $\mathcal{D}$ and $\mathcal{Q}$ respectively, can be inferred from the baseline relevance (baseline search system score or ranking for the given query). Using a symmetric reparametrization of models 4.11 and 4.12 and identifying latent variable spaces, we obtain the following complete data model

Figure 4.13: Producer-consumer semantic model



where the dashed lines over the graphical model indicate the known priors in the observed data model. Distributions $\tilde{p}(w,d)$, $\tilde{p}(w,q)$ and $\tilde{p}(d,q)$ are obtained from the described priors fixing suitable $\tilde{p}(q)$ (for, example, uniform) and $\tilde{p}(d)$. The algorithm then maximizes the combination of the utility-biased likelihoods of the parameters using all three prior distributions,

$$\mathcal{L} = \sum_{w\in\mathcal{W},d\in\mathcal{D}} \tilde{p}(w,d)\log\sum_{z\in\mathcal{Z}} p(w|z)p(d|z)p(z) +$$
$$\sum_{w\in\mathcal{W},q\in\mathcal{Q}} \tilde{p}(w,q)\log\sum_{z\in\mathcal{Z}} p(w|z)p(q|z)p(z) + \sum_{d\in\mathcal{D},q\in\mathcal{Q}} \tilde{p}(d,q)\log\sum_{z\in\mathcal{Z}} p(d|z)p(q|z)p(z)$$

In order to do so, we will simultaneously execute three utility-biased EM iterations coupled by a common set of parameters, in the form of distributions $p(w|z)$, $p(d|z)$, $p(q|z)$ and $p(z)$.

In chapter one, we proved that minorizing relation is additive, so

$$g(\theta,\theta_t) = g_{\tilde{p}(w,d)}(\theta,\theta_t) + g_{\tilde{p}(w,q)}(\theta,\theta_t) + g_{\tilde{p}(d,q)}(\theta,\theta_t)$$

minorizes $\mathcal{L}$. E step comes from direct Bayes computation for each pair of coupled variables

$$p(z|w,d) = \frac{p(w|z)p(d|z)p(z)}{\sum_{z\in\mathcal{Z}} p(w|z)p(d|z)p(z)}$$

$$p(z|d,q) = \frac{p(d|z)p(q|z)p(z)}{\sum_{z\in\mathcal{Z}} p(d|z)p(q|z)p(z)}$$

$$p(z|w,q) = \frac{p(w|z)p(q|z)p(z)}{\sum_{z\in\mathcal{Z}} p(w|z)p(q|z)p(z)} \tag{4.2.14}$$

M step $Q_{\tilde{p}}$ functional equations and Lagrange multiplier functional (4.2.7) are additive over the three combined minorizing functionals. Therefore, direct computation yields

$$p(w|z) \propto \sum_{d'\in\mathcal{D}} \tilde{p}(w,d')p(z|w,d') + \sum_{q'\in\mathcal{Q}} \tilde{p}(w,q')p(z|w,q')$$

$$p(d|z) \propto \sum_{w'\in\mathcal{W}} \tilde{p}(w',d)p(z|w',d) + \sum_{q'\in\mathcal{Q}} \tilde{p}(d,q')p(z|d,q')$$

$$p(q|z) \propto \sum_{w'\in\mathcal{W}} \tilde{p}(w',q)p(z|w',q) + \sum_{w'\in\mathcal{W}} \tilde{p}(w',q)p(z|w',q)$$

$$p(z) \propto \sum_{w'\in\mathcal{W},d'\in\mathcal{D}} \tilde{p}(w',d')p(z|w',d') + \sum_{w'\in\mathcal{W},q'\in\mathcal{Q}} \tilde{p}(w',q')p(z|w',q') + \sum_{d'\in\mathcal{D},q'\in\mathcal{Q}} \tilde{p}(d',q')p(z|d',q') \tag{4.2.15}$$

The resulting aspect space would theoretically capture at the same time document semantics, query latent intents and prior document relevance. If we compare this algorithm to RapLSA, we observe that this would be already possible if we were able to build a prior $\tilde{p}(w,d,q)$ that represented at the same time all three sources of information. Nevertheless, this can be difficult to estimate unless our baseline system gives us access to a complete language model that fits both documents and queries.

The observed data prior proposed for RapLSA uses a simplified generative model for the incomplete data that would make it almost impossible to obtain this kind of information. Estimating distributions $\tilde{p}(w,d)$, $\tilde{p}(w,q)$ and $\tilde{p}(w,d)$ doesn't allow us to infer a complete distribution $\tilde{p}(w,d,q)$ (known distributions impose less than $WD+WQ+DQ+1$ independent equations over WDQ variables). Trying to do so would result in a more restrictive system. If such a distribution is fixed, RapLSA would obtain the (locally) closest distribution factoring through the complete data model, while this new proposed algorithm would obtain distributions (locally) closest to set of distributions marginalizing to the given ones. Therefore, the obtained distributions would potentially be closer to the primal priors, while RapLSA over an estimated mixture would just be closer to that approximated joint distribution.

Content based recommender models

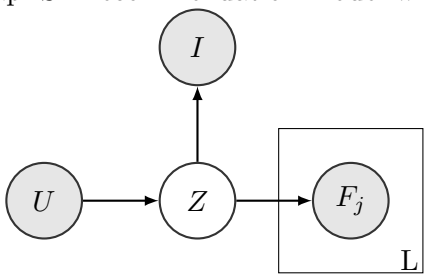
In our previous exposition about the similarities between pLSA models for recommendation and search, we described how, while user profiles seem to take the role of implicit queries and items seem to represent documents, there didn't exist a clear substitute for the role of words within documents. We will describe a simpler, yet general, alternative for the previous model when there exist an explicit set of information tokens that can take that content-describing role, thus leading to a content based model.

Let us suppose that there exist a set $\mathcal{F} = \{\mathcal{F}_1, \cdots, \mathcal{F}_L\}$ of item feature spaces, such that for each item $i \in \mathcal{I}$ an associated vector of prior feature distributions can be estimated $p(\tilde{f}|i) = \tilde{p}(f_1, \ldots, f_L|i)$ for each $f_j \in \mathcal{F}_j$.

We will assume that triples (u, i, f) are observed following the next generative model

- A random user $u \in \mathcal{U}$ is drawn from a prior distribution $\tilde{p}(u)$.
- A latent user feature $z \in \mathcal{Z}$ is selected from a distribution $p(z|u)$
- An item $i \in \mathcal{I}$ is picked from distribution $p(i|z)$
- For each feature $\mathcal{F}_j \in \mathcal{F}$, pick a feature value $f_j \in \mathcal{F}_j$ from a distribution $p(f_j|z)$

Figure 4.14: RapLSA recommendation model with item features



There exist several kind of feature spaces for which the model can be applied. As an example, let us consider film recommendation. Suitable features would include

- **Title and plot summary word distributions:** The combination of title and plot summary gives us explicitly a plain text description of the algorithm, completely analogous to a text document in search tasks.
- **The director:** binary prior that takes value 1 for the film director.
- **List of principal actors:** distribution among the set of actors, with higher probability the more important its role was or the more popular the actor is
- **Producer companies:** uniform among set of producers involved in the movie
- **Explicit known genres:** if a set of genres is available, each film can be given a distribution over them, for example, taking it as uniform supported over the genres to which the film belong

- **Country:** Adding a variable that ranges over the full existing countries would be computationally inefficient. Instead, a set of the most ones plus an "other" option can be used. Prior distribution

As a natural assumption, in order to build a distribution over the full observed data $\overline{X} \subseteq \mathcal{U} \times \mathcal{I} \times \mathcal{F}$ we will suppose that feature distributions only depend on the item, thus making all feature variables $\mathcal{F}_j$ independent from the user $\mathcal{U}$ given an item $i \in \mathcal{I}$. On the other hand, a distribution $\tilde{p}(u, i)$ can be estimated from a baseline relevance source as with the usual RapLSA, leading to a prior utility distribution

$$\tilde{p}(u, i, f) = \tilde{p}(u, i)\tilde{p}(f|i)$$

Usually, in order to build the prior, we can further consider that features are independent from each other given a the item, so

$$\tilde{p}(u, i, f) \sim \tilde{p}(u, i) \prod_{j=1}^{L} \tilde{p}(f_j|i)$$

It becomes clear that the availability of any of these features would be useful for building a more precise user profile. Resulting latent intent space would then be expected to be more refined, capturing the semantic information simultaneously from user similarities and item features.

As usual, we will consider the equivalent reparameterized symmetric version. Instantiating algorithm 4.1 with the modified tempered E step (4.2.8) yields

Algorithm 4.7 Relevance aware pLSA recommendation model

procedure RapLSA($\tilde{p}(u,i,f)$,$\overline{X} = \{(u,i,f)\}$,$\beta$)
  Start with a random distributions $p(u|z), p(i|z), p(f_j|z), p(z)$
  for each t **do**
    Step E:
    for each $(u,i,f) \in \overline{X}$ and every $z \in \mathcal{Z}$ **do**

$$p(z|u,i,f) = \frac{\left(p(u|z)p(i|z)p(z)\prod_{j=1}^{L} p(f_j|z)\right)^{\beta}}{\sum_{z\in\mathcal{Z}}\left(p(u|z)p(i|z)p(z)\prod_{j=1}^{L} p(f_j|z)\right)^{\beta}}$$

    end for
    Step M:
    for each $z \in \mathcal{Z}$, $u \in \mathcal{U}$ and $i \in \mathcal{I}$ **do**

$$p(u|z) = \frac{\sum_{(u',i,f')\in\overline{X}} p(z|u',i,f')\tilde{p}(u',i,f')}{\sum_{(u',i',f')\in\overline{X}} p(z|u',i',f')\tilde{p}(u',i',f')}$$

$$p(i|z) = \frac{\sum_{(u,i',f')\in\overline{X}} p(z|u,i',f')\tilde{p}(u,i',f')}{\sum_{(u',i',f')\in\overline{X}} p(z|u',i',f')\tilde{p}(u',i',f')}$$

$$p(f_j|z) = \frac{\sum_{\{(u,i',f')\in\overline{X}|f'_j=f_j\}} p(z|u,i')\tilde{p}(u,i')}{\sum_{(u',i',f')\in\overline{X}} p(z|u',i',f')\tilde{p}(u',i',f')}$$

$$p(z) = \frac{\sum_{(u',i',f')\in\overline{X}} p(z|u',i',f')\tilde{p}(u',i',f')}{\sum_{z\in\mathcal{Z}}\sum_{(u',i',f')\in\overline{X}} p(z|u',i',f')\tilde{p}(u',i',f')}$$

    end for
  end for
  The algorithm stops either after a maximum number of iterations or when distributions $p(u|z), p(i|z), p(z)$ have converged
end procedure

In this case, the minimized function would correspond to

$$\mathrm{KLD}\left(\tilde{p}(u,i)\|p(u,i)\right) + \sum_{u\in\mathcal{U},i\in\mathcal{I}} \tilde{p}(u,i)\,\mathrm{KLD}\left(\tilde{p}(f|i)\|p(f|u,i)\right)$$

Therefore, as usual, both relevance information $\tilde{p}(u,i)$ and feature distributions $\tilde{p}(f|i)$ are learned, prioritizing the approximation of the features of the most relevant items for each user.

Collaborative filtering based recommender models

The previously described recommender models (4.5 and 4.6) can be used independently on the kind of baseline relevance source used. This includes any kind of content based, collaborative or hybrid filtering methods. In this section, we will explore an alternative procedure for the collaborative filtering scenario.

A common procedure for a collaborative filtering method diversification would be the following

1. A user similarity matrix $sim(u, v)$ is estimated using a certain similarity computing algorithm $\mathcal{A}_{sim}$

2. Apply a collaborative filtering recommendation algorithm $\mathcal{A}_{CF}$ that computes the objective function using the previous similarity. For example, if we want to uniformize user biases, we can take

$$s(u, i) = \mu(r) + \sigma(r) \sum_{v \in N_u} sim(u, v)\tilde{r}(v, i)$$

   where $\tilde{r}(v, i)$ is the normalized rating

$$\tilde{r}(v, i) = \frac{r(v, i) - \mu_v(r)}{\sigma_v r}$$

3. PLSA is applied over the observed pairs $\overline{X} = \{(u, i)\}$. If our recommendation models are used, RapLSA is applied using $\tilde{p}(u, i) \propto s(u, i)$ in order to obtain a latent variable space that represents latent interests of users. A complete distribution $p(u, i, z)$ is approximated.

4. A diversification algorithm $\mathcal{A}_{div}$ like xQuAD or IA-Select is used to diversify $\mathcal{A}_{CF}$ results based on the latent factor space extracted by RapLSA, the estimated distribution $p(u, i, z)$ and the baseline score $s(u, i)$.

In this kind of approaches, user similarity is computed in absolute terms, not taking into account that users might be similar to each other in some aspects of their profiles bot not in others. For example, let us consider two users A and B in a movie recommendation task.

- User A is mainly interested in science fiction and terror movies.

- User B is interested in science fiction and comedy movies. Moreover, he likes the same kind of science fiction movies than A.

The overall similarity between user profiles A and B can be very low if A watches more terror film than science fiction ones and B watches more comedies than science fiction. Nevertheless, if we restrict ourselves to science fiction recommendation, user A and B are very similar. If we want to recommend user A a science fiction film it would indeed be interesting to analyze user B tastes in science fiction films.

Therefore, in the context of latent aspects, it is interesting to analyze user similarity conditioned by a certain aspect. We propose to use RapLSA to extract an aspect space from both item ratings and baseline user similarities. User baseline similarity is not necessarily restricted to evaluate common item rating profiles. It can incorporate other sources of information, such as user popularity or social data. In this way, we can incorporate both ground relevance and inter-user information to the pLSA model.

Let us take $\tilde{p}(v|u) \propto sim(u, v)$, where we will assume that the similarity function is positive and bounded. We will interpret $\tilde{p}(v|u)$ as the probability that v has the

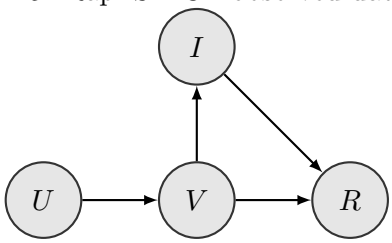
same user model/profile than u for a fixed u. Utility distribution $\tilde{p}(i|u)$ can be estimated through item scores, taking $\tilde{p}(i|u) \propto s(u,i)$. Moreover, we can incorporate explicit ratings taking binary $\tilde{p}(r|i,u)$. Finally, a user prior $\tilde{p}(u)$ is assumed. This can be taken as uniform or through a popularity model.

As observed data, we will consider quadruples (u,v,i,r) consisting of a triple (v,i,r) such that rating r is registered for user v and item i, and a user u such that u and v might share a common latent intent profile. A priori, user u would range over all $\mathcal{U}$ for all triple (v,i,r), and expected probability of u and v sharing profile is regulated by $\tilde{p}(v|u)$. Observed data is enhanced with distribution

$$\tilde{p}(r,v,i,u) = \tilde{p}(r|i,v)\tilde{p}(i|v)\tilde{p}(v|u)\tilde{p}(u)$$

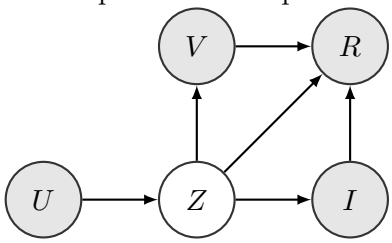
following the Bayesian model

Figure 4.15: RapLSA CF observed data model



Observed items and ratings can be considered independent from user u given user v, because u and v are assumed to share a common latent profile. We consider the following latent variable model

Figure 4.16: RapLSA CF complete data model



Subgraph $\{R,U,I,Z\}$ can be refactorized using any of the pLSA rating recommendation models used by Hofmann (2004). The final estimated user-item-rating distribution is taken as

$$p(r,i,u) = \sum_{z\in\mathcal{Z}} \sum_{v\in\mathcal{V}} p(r|i,v,z)p(i|z)p(v|z)p(z|u)$$

We observe that, similarly to classic collaborative filtering algorithms, item ratings are computed as a mixture of item ratings for other users, pondered by the similarity between the original user and the others.

If $\tilde{p}(u)$ is considered as uniform, the algorithm is equivalent to considering every distribution as conditioned by u. Then, the algorithm minimizes, for each user $u \in \mathcal{U}$

$$\mathrm{KLD}\left(\tilde{p}(v|u)\|p(v|u)\right) + \sum_{v\in\mathcal{U}} \tilde{p}(v|u)\,\mathrm{KLD}\left(\tilde{p}(i|v)\|p(i|v,u)\right) + \sum_{i\in\mathcal{I},v\in\mathcal{U}} \tilde{p}(i,v|u)\,\mathrm{KLD}\left(\tilde{p}(r|i,v)\|p(r|i,v,u)\right)$$

First factor makes algorithm learn the explicit baseline user similarity. Second factor makes algorithm learn item distribution for each user. As $\tilde{p}(i|v)$ is independent of u, the algorithm tries to make $p(i|v,u)$ the closest possible to be independent of u, prioritizing for each user, learning the item distribution of its closest neighbours. It is worth noticing that this is done in both directions, ranging over u and v, as the more close two users are, the more similar $\tilde{p}(i|u)$ and $\tilde{p}(i|v)$ are supposed to be, and the algorithm would make $p(i|u,v)$ and $p(i|v,u)$ be both very similar to each other and similar to $p(i|v)$ in the Kullback-Leibler divergence sense. Finally, the third factor makes the algorithm learn the ratings for each of the observed users and items, overall prioritizing learning those having higher probability to being relevant to user u.

User similarity $sim(u,v)$ can be estimated using any available method, but a probabilistic method would link better with the rest of the theoretical framework. In particular, conditional exponential model for user similarity proposed by Jin et al. (2004) fits both the global probabilistic formulation and the maximum entropy principle applied in EM tempering. Taking into account the previous divergence factorization, $\mathrm{KLD}\left(\tilde{p}(v|u)\|p(v|u)\right)$ being minimized and $\tilde{p}(v|u)$ being obtained by the maximum entropy principle makes $p(v|u)$ estimation tend to be of maximum entropy possible.

Therefore, the obtained distribution $p(v|u)$ is an approximation by the maximum entropy principle. If we add the entropic regularization term of tempered EM (4.2.8), the whole model correspond to the distribution covering the observed data with maximum entropy.

4.2.8 Geometric interpretation of the algorithm

The utility biased Expectation-Maximization algorithm can be given an interesting theoretical interpretation in the framework of information geometry.

Let us consider a complete data model consisting on a set of observed variables $X = \{X_1, \ldots, X_n\}$ and a set of latent variables $Z = \{Z_1, \ldots, Z_k\}$, represented by a Bayesian graph G. Let $\mathcal{P}(X)$ be the manifold with boundary parameterizing all possible categorical probability distributions over the observed variables X. As a differential manifold, it is defined as the submanifold of $\mathbb{R}^{|X_1|\cdots|X_n|}$ corresponding to the nonnegative coordinates with sum one, i.e.

$$\mathcal{P}(X) = \{p(x)|x \in X, p(x) \geq 0, \sum_{x\in X} p(x) = 1\}$$

the manifold can be given a Riemannian structure using Fisher information metric. In the case of a space of categorical distributions, this metric takes the following

explicit form in the adapted coordinates $\left\{\frac{\partial}{\partial p(x)}\right\}_{x\in X}$

$$g|_p = \sum_{x\in X} p(x) d(\log p(x)) \otimes d(\log p(x)) = \sum_{x\in X} \frac{1}{p(x)} d(p(x)) \otimes d(p(x))$$

Reparameterizing $p(x) = f_x^2$, $\mathcal{P}(X)$ is diffeomorphic to the possitive quadrant of the sphere $S^{|X_1|\cdots|X_n|-1} \subset \mathbb{R}^{|X_1|\cdots|X_n|}$. If we pull back the fisher information, we obtain that it is isometric to the positive quadrant of the euclidean sphere or radius 2, i.e., the sphere with the induced ambient euclidean metric, as

$$g_c|_f = \sum_{x\in X} df_x \otimes df_x = \sum_{x\in X} d(\sqrt{p(x)}) \otimes d(\sqrt{p(x)}) = \frac{1}{4} \sum_{x\in X} \frac{d(p(x)) \otimes d(p(x))}{p(x)} = \frac{1}{4} g|_p$$

Fisher information metric is heavily linked to Kullback-Leibler divergence. For each distribution p, metric matrix at point p, g_p, correspond to the Hessian of the Kullback-Leibler divergence with respect to p, $\mathrm{KLD}\,(p\|\cdot)$.

Now, let us consider the space of distribution over X and Z, $\mathcal{P}(X, Z)$. This manifold parameterizes all possible distributions of the complete data. Marginalizing the latent variables allow us to project $\mathcal{P}(X, Z)$ into $\mathcal{P}(X)$ taking

$$\begin{array}{ccc} \mathcal{P}(X,Z) & \xhookrightarrow{\pi} & \mathcal{P}(X) \\ p(x,z) & \longmapsto & \sum_{z\in Z} p(x,z) \end{array}$$

Let $\mathcal{P}_0(X)$ denote the interior of the manifold, corresponding to distributions that are nowhere zero (its support is all X). On the other hand, let us suppose that a Bayesian network is given over the complete data, represented by a directed acyclic graph G. Let S_j be a node in G and let us denote by $S_{\pi(j)}$ the set of parents of node S_j. We will denote by $\mathcal{P}(S_j|S_{\pi(j)})$ the manifold parameterizing the set of conditional distributions of variable S_j given the value of $S_{\pi(j)}$, i.e.

$$\mathcal{P}(S_j|S_{\pi(j)}) = \{p(s|v) | s \in S_j, v \in S_{\pi(j)}, p(s|v) \geq 0, \sum_{s\in S_j} p(s,v) = 1 \forall v \in S_{\pi(j)}\}$$

From the definition it becomes clear that $\mathcal{P}(S_j|S_{\pi(j)}) \cong \mathcal{P}(S_j)^{|S_{\pi(j)}|}$. Moreover, for any couple of variable sets X, Y, Bayes theorem provides us an isomorphism between the interior of the manifolds

$$\begin{array}{ccc} \mathcal{P}_0(X|Y) \times \mathcal{P}_0(Y) & \xleftrightarrow{\sim} & \mathcal{P}_0(X,Y) := \mathcal{P}_0(X \cup Y) \\ (p(x|z), p(z)) & \longmapsto & p(x|z)p(z) \\ \left(\frac{p(x,z)}{\sum_{x\in X} p(x,z)}, \sum_{x\in X} p(x,z)\right) & \longmapsto & p(x|z)p(z) \end{array}$$

If we drop the nonzero condition, we still have a projection $\mathcal{P}(X|Y) \times \mathcal{P}(Y) \to \mathcal{P}(X, Y)$. This correspondence can be clearly extended to an arbitrary number of factors following Bayes rule. This motivates the following definition. Let $\mathcal{P}_G(X, Z)$

be the subvariety of $\mathcal{P}(X,Z)$ corresponding to distributions factorizing through the graphical model G. This corresponds to the image of the map

$$\begin{array}{c} \prod_{j=1}^{n} \mathcal{P}(S_j|S_{\pi(j)}) \xrightarrow{\qquad\varphi_G\qquad} \mathcal{P}_0(X,Y) := \mathcal{P}_0(X \cup Y) \\ (p(S_j(x,z)|S_{\pi(j)}(x,z))) \longmapsto p(x,z) = \prod_{j=1}^{n} p(S_j(x,z)|S_{\pi(j)}(x,z)) \end{array}$$

Let $\mathcal{P}_G(X)$ be the projection of $\mathcal{P}_G(X,Y)$ through π, i.e., the manifold consisting on the marginalized distributions that factor through G.

$$\mathcal{P}_G(X) := \pi(\mathcal{P}_G(X)) = (\pi \circ \varphi_G)\left(\prod_{j=1}^{n} \mathcal{P}(S_j|S_{\pi(j)})\right) = \\ = \left\{ p(x) \in \mathcal{P}(X) | \exists p(x,z) \in \mathcal{P}_G(X,Z)\, p(x) = \sum_{z \in Z} p(x,z) \right\}$$

Now, let us consider some utility enhanced data $(\overline{X}, \tilde{p})$ over which utility-biased EM is applied. Clearly, $\tilde{p} \in \mathcal{P}(X)$. On the other hand, as the estimated distribution factorizes through the complete data model, $p(x,z) \in \mathcal{P}_G(X,Z)$. Equation (4.2.1) proves that the utility-biased maximum likelihood estimator corresponds to the set of parameters minimizing $\mathrm{KLD}\,(\tilde{p}\|p)$. As considered distributions are categorical and the set of parameters is taken to be the full set of values of the distributions $p(S_j|S_{\pi(j)})$, the utility-biased maximum estimator correspond to finding the distribution $p(x,z) \in \mathcal{P}_G(X,Z)$ such that $\mathrm{KLD}\,(\tilde{p}\|p)$ is minimum.

Utility-biased EM algorithm locally minimizes this divergence. Instead of considering the corpus-dependent complex local dynamics, we will focus on the global objective dynamics and on comparing utility-biased EM with classic EM. This will give a taste on the qualitative differences between both methodologies from the information-theoretical point of view.

For a given $\tilde{p}$ fixed, given another point $q \in \mathcal{P}(X,Z)$, $\mathrm{KLD}\,(\tilde{p}\|q)$ gives the square of the length of the geodesic $\gamma_t(x) = \frac{p^{1-t}(x)q^t(x)}{\sum_{x' \in X} p^{1-t}(x)q^t(x)}$ from p to q. Therefore, for a fixed point $\tilde{p}$, minimizing the divergence of any point of $\mathcal{P}_G(X)$ with respect to $\tilde{p}$ is equivalent to finding the projection of $\tilde{p}$ to the manifold $\mathcal{P}_G(X)$.

We proved that classic EM correspond to taking $\tilde{p}(x) \propto n(x)$. Let us denote that canonical prior distribution by $\tilde{p}_c(x)$. Then changing from EM to utility-based EM corresponds to rotating the projection point in the space of distributions $\mathcal{P}(X)$.

In order to appreciate better the geometric effect of this change, let us particularize the previous manifolds for a pLSI model executed over a single query q (model 4.2). Classic Hofmann prior is

$$\tilde{p}_c(w,d) = \tilde{p}_c(w|d)\tilde{p}_c(d) = \frac{n(w,d)}{|d|}\frac{|d|}{\sum_{d \in D}|d|}$$

while proposed prior would be

$$\tilde{p}(w,d) = \tilde{p}_c(w|d)\tilde{p}(d)$$

where, as we considering a single query q, $\tilde{p}(d) = \tilde{p}(d|q)$. $\mathcal{P}_G(W, D)$ can be explicitly described as the image of $\mathcal{P}(W|Z) \times \mathcal{P}(D|Z) \times \mathcal{P}(Z)$ by $\pi \circ \varphi_G$. Moreover, if we naturally suppose that samples are available for all documents within the corpus, we can restrict the model to the interior of $\mathcal{P}(D)$ and consider the equivalent problem in $\mathcal{P}(W|D) \times \mathcal{P}_0(D)$. Pulling back Kullback-Leibler divergence from $\mathcal{P}(W, D)$ induces the following divergence in $\mathcal{P}(W|D) \times \mathcal{P}_0(D)$

$$\begin{aligned}
&\mathrm{KLD}\left(p(w,d)\|q(w,d)\right) = \\
&\qquad \sum_{d\in D}\sum_{w\in W} p(w|d)p(d)\left(\log p(d) - \log q(d) + \log p(w|d) - \log q(w|d)\right) = \\
&\sum_{d\in D} p(d)\left(\log p(d) - \log q(d)\right) + \sum_{d\in D} p(d) \sum_{w\in W} p(w|d)\left(\log p(w|d) - \log q(w|d)\right) = \\
&\qquad\qquad \mathrm{KLD}\left(p(d)\|q(d)\right) + \sum_{d\in D} p(d)\,\mathrm{KLD}\left(p(w|d)\|q(w|d)\right)
\end{aligned}$$

This divergence comes as the sum of the divergences of the projections of $p(w, d)$ and $q(w, d)$ to both spaces $\mathcal{P}_0(D)$ and $\mathcal{P}(W|D)$. In the first space, the divergence is the usual Kullback-Leibler divergence. Using the isomorphism $\mathcal{P}(W|D) \cong \mathcal{P}(W)^{|D|}$, the second divergence corresponds to a pondered sum of divergences of each factor, where each term is pondered by the corresponding $p(d)$.

Taking this consideration into account, back to the original pLSI scenario, we can translate the projection problem into the factorized space $\mathcal{P}_0(D) \times \mathcal{P}(W)^{|D|}$, projecting $\mathcal{P}_G(W, D)$ into both factors through Bayes theorem. Prior $\tilde{p}(w, d)$ decomposes in the classical priors $\tilde{p}_c(w|d)$ and the variable document prior $\tilde{p}(d)$.

Geometrically, distributions $\tilde{p}(w|d)$ being fixed, if we restrict ourselves to $\mathcal{P}(W|D)$ we can understand the change of prior $\tilde{p}(d)$ not as a movement of the projection point, but as a dilatation of the space in certain directions. Considering the image of $\mathcal{P}_G(W, D)$ in $\mathcal{P}_0(D) \times \mathcal{P}(W)^{|D|}$. Finding the closest point to $(\tilde{p}(d|q), \tilde{p}_c(w|d))$ would be equivalent to finding the closest point to $(\tilde{p}_c(d), \tilde{p}_c(w|d))$ if we dilated each space $\mathcal{P}(W)$ corresponding to $D = d$ by the factor

$$\lambda_d = \frac{\tilde{p}(d|q)}{\tilde{p}_c(d)}$$

and rotate $\tilde{p}(d)$ in the space $\mathcal{P}_0(D)$

Therefore, we prove that geometrically, changing the prior from the Hofmann pLSA one to the relevance $\tilde{p}(d|q)$ corresponds to compressing the information variety along term distribution spaces $\mathcal{P}(W)$ corresponding to less relevant documents and stretching it along distribution spaces $\mathcal{P}(W)$ corresponding to relevant documents, while rotating the projection point in the space $\mathcal{P}_0(D)$.

In particular, this gives a geometric proof that the algorithm takes into account relevant documents information more than the standard one, neglects information from irrelevant documents and optimizes pure relevance information, as by changing $\tilde{p}(d)$ to become $\tilde{p}(d|q)$, the algorithm is forced to obtain a projected point whose marginal to the $\mathcal{P}_0(D)$ factor lies close to the relevance distribution $\tilde{p}(d|q)$.

4.3 Aspect filtering

Experiments in both search and recommendation diversity show that aspect distributions obtained by the application of EM and TEM algorithms may have some undesirable characteristics when they are fed directly to diversification algorithms as intent spaces. Experimental data shows that aspect distributions tend to be highly entropic, specially when applying TEM with lower values of β, leading to documents sometimes having really similar mixing proportions, therefore obstructing diversification algorithms to discriminate the novel semantic information among documents. Resulting distributions also tend to be noisy, with long heavy tails composed of topics which documents don't really belong to. In order to develop more suitable distributions which lead to improved diversity performance we have developed some post-processing filters for the distributions. The filters can be applied to any algorithm derived from the previously described abstract probabilistic model, and they are shown to improve the results both in classic pLSA and new models.

As the notation is very different from the search to the recommendation tasks, but filters are completely analogous in both cases, we will only describe them in the search diversity context. This will simplify and shorten the necessary mathematical derivations while summarizing the main ideas.

4.3.1 Cutoff filter

The asymptotic convergence of the EM algorithm leads to the absence of 0-probability aspects for every document or query if initialized in a random way. In order to prove it, we just have to notice that the initial random parameters are almost never zero (in the probabilistic sense). If the parameters are nonzero for a given step of the algorithm, equations (4.2.4), (4.2.3) and, in general, EM equations for any model similar to the ones already described imply that each parameter is updated in E and M steps as a rational function with positive coefficients of the parameters of the previous iteration, therefore leading to positive new parameters.

On the other hand, the dimensional difference between $\mathcal{W}$ and $\mathcal{Z}$ makes EM algorithm converge to distributions that minimize the common information between term distributions of each semantic aspect. Therefore, aspect-document distributions naturally tend to mainly concentrate on some aspects for each document and, in general, they present a really high difference between the probability of a few aspects and the rest of them. For this reason, one can experimentally check that the distribution matrix is full of nearly-zero values.

When retrieving the documents after diversification, these non-zero values produce noise that may affect the process. In order to clear that noise we suggest applying a filter consisting in truncating the tail of the aspect distributions and then renormalizing the obtained probability vectors. This transform the previous dense aspect distribution matrices y sparse ones, with sparsity degree depending on the way the extreme of the cut tail is selected for each document.

The previous general filter can be implemented in several ways. For example, for each document d we select aspects z such that

- $p(z|d) \geq \xi$ for some fixed threshold $\xi > 0$.

- z belongs to the top x percentile, for some x.
- $p(z|d)$ is not a bottom outlier of $\{p(z|d)\}$ at ξ-sigma significance for some fix $\xi > 0$, i.e.

$$p(z|d) >= \frac{1}{K} - \xi\sqrt{\mathrm{Var}(\{p(z|d)\})}$$

In the experiments we have chosen to take an intermediate approach between the first and third approach, dropping those aspects whose probability is under the uniform distribution, i.e., those such that

$$p(z|d) < \frac{1}{K}$$

This corresponds to taking $\xi = \frac{1}{K}$ in the first case or $\xi = 0$ in the third one and has the advantage of being an auto-adjusting threshold when varying the size of the latent space.

4.3.2 Uniform aspect filter

This kind of ad-hoc filter corresponds to changing aspect query distributions $p(z|q)$ to uniform distributions over $\mathcal{Z}$ before passing the information to the diversifier. The main idea behind this is that there exist a difference between query aspects and user intentions. RapLSA infers the first, but can't precisely deduce the later unless explicit user information is available. In this situation, obtained aspects can be considered as a good semantic approximation of user abstract information needs, but relevance information, codified in distributions $p(z|q)$, may be biased, as it is approximated from the baseline relevance estimation.

As an example illustrating the differences between the RapLSA-estimated and real relevance to the user, let us consider a query with a "main" aspect and "secondary" aspects. For example, executing query "Java" in most commercial search systems will mainly produce "Java programming language" results in the first positions. Query "Java" has another possible intention, related to "Java island". While results corresponding to this other meaning still appear in the baseline ranking, they are clearly considered by the system as less relevant. Now let us suppose that we execute RapLSA algorithm for this query q and such a typical relevance prior $\tilde{p}(d|q)$.

RapLSA introduces the query in order to build query-specific distributions. Document and term distributions $p(d|z,q)$ and $p(w|z,q)$ are inferred prioritizing information contained in the most relevant documents. As island-related documents term distribution are be completely different from the programming-related ones and island pages don't have a neglectable relevance prior (they have lower prior than programming ones, but are not spam documents), one or many aspects (depending on the aspect space size) would likely capture this sense. For simplicity, let us suppose that there exist only one such aspect $\overline{z}$. Distributions $p(d|\overline{z},q)$ and $p(w|\overline{z},q)$ are expected to effectively "select" documents referring to the island, so the semantics are correctly extracted, allowing the diversifier to work properly.

Nevertheless, $p(\overline{z}|q)$ and $p(z|q)$ for $z \neq \overline{z}$ can become a problem for diversity enhancing. As all "island" documents are systematically considered as less relevant from the baseline point of view, $p(\overline{z}|q)$ would be little, compared with other aspect

probabilities. Therefore, when the diversity algorithm is applied, it will prioritize covering the other aspects before covering $\overline{z}$, thus degrading the original "secondary" island aspect of the query.

This effect only depends on how much the baseline score penalizes island documents and not on real user intentions (unless the baseline is personalized). Setting $p(z|q)$ as uniform before using the diversifier "resets" all baseline biases towards any of the aspects as far as of the intent space. Baseline relevance can be introduced again into the diversifier if an algorithm like xQuAD is used. The difference is that, in this case, we are provided a "balanced" latent space that provides "pure" diversity information and the impact of prior relevance can be regulated with the own algorithm parameters.

4.3.3 α-means filter

We have observed that the aspect distributions become sometimes too uniform, in the sense that some documents have a large amount of aspects with nearly the same probability. Although the probabilities are different, the high entropy of the distribution makes discriminating among the aspects difficult for the diversification algorithms.

Moreover, tempering is based on selecting distributions $p(z|x)$ with maximum entropy among those with similar "energy". Taking into account the explicit form of the total functional $\mathcal{F}_{\tilde{p},\beta}$ locally maximized by TEM (4.2.11), for $0 < \beta < 1$, TEM maximizes a convex combination of likelihood and the entropy of the distributions $p(z|x_i, \theta)$, so it is expected, in general, to find distributions with higher entropy $H(p(z|x_i, \theta))$.

In order to show the effect of feeding highly entropic distributions to a diversifier, let us consider a simplified example. Suppose that we obtain a distribution with high $H(p(z|x_i, \theta)$ for each sample x_i. This is not really far from reality, as we have just proved that TEM maximizes the sum of all of them and, while experimental results in search diversity experiments prove that not all of the documents will have extremely highly entropic aspect distributions, they show that a great majority of them tend to be almost uniform, thus attaining nearly maximum entropy.

Let us consider two samples x_i, x_j with high entropy. Using parallelogram identity for the Kullback-Leibler divergence (Csiszar, 1975), we can find a bound for the Jensen-Shannon divergence between their topic distributions. In order to simplify the notation, let $p_i(z) = p(z|x_i, \theta)$, and let us denote the uniform distribution over

$\mathcal{Z}$ as $u(z) = \frac{1}{K}$. Then

$$2\,\mathrm{JSD}\left(p_i \| p_j\right) = \mathrm{KLD}\left(p_i \middle\| \frac{p_i + p_j}{2}\right) + \mathrm{KLD}\left(p_j \middle\| \frac{p_i + p_j}{2}\right) =$$

$$\mathrm{KLD}\left(p_i \| u\right) + \mathrm{KLD}\left(p_j \| u\right) - 2\,\mathrm{KLD}\left(\frac{p_i + p_j}{2} \middle\| u\right) \leq$$

$$\mathrm{KLD}\left(p_i \| u\right) + \mathrm{KLD}\left(p_j \| u\right) = \sum_{z \in \mathcal{Z}} p_i(z) \log p_i(z) - \sum_{z \in \mathcal{Z}} p_i(z) \log u(z) +$$

$$\sum_{z \in \mathcal{Z}} p_j(z) \log p_j(z) - \sum_{i=1}^{n} p_j(z) \log u(z) = -H(p_i) - H(p_j) +$$

$$\log(K) \sum_{z \in Z} \left(p_i(z) + p_j(z)\right) = -H(p_i) - H(p_j) + 2\log(K) \quad (4.3.1)$$

In particular, suppose that the entropy of p_i and p_j are near enough to the top possible entropy, i.e., that for a certain $\epsilon \geq 0$ yields

$$H(p_i) \geq \max_p H(p) - \epsilon$$

concavity of function $f(x) = -x \log x$ and Jensen inequality implies that the maximum entropy is attained for the uniform distribution, so that

$$H(p_i) \geq H(u) - \epsilon = \sum_{z \in \mathcal{Z}} \frac{1}{K} \log K - \epsilon = \log K - \epsilon$$

If the last bound holds, substituting in equation (4.3.1) yields

$$\mathrm{JSD}\left(p_i \| p_j\right) \leq \frac{-H(p_i) - H(p_j) + 2\log K}{2} \leq \epsilon$$

Thus, if $H(p_i) \geq H(u) - \epsilon$ for some ϵ and all samples $i = 1, \ldots, n$, then we can assure that the divergence between any two of the distributions is less than ϵ. In our diversification scenario, this means that if EM results in high entropy results, then the diversifying algorithm will be given really close aspect distributions for every pair of documents. In particular, using Pinsker-Csiszár-Kullback inequality (Pinsker, 1964; Csiszár, 1967; Kullback, 2006), the following bound in terms of the Jensen-Shannon divergence holds (Yamano, 2014)

$$\sum_{z \in Z} |p_i(z) - p_j(z)| \leq \sqrt{8\,\mathrm{JSD}\left(p_i \| p_j\right)} \leq \sqrt{8\epsilon}$$

In algorithms like xQuAD this would make the variability of diversity scores among the corpus to be negligible in comparison to the variability of the baseline score, so that the final score ranking will become essentially the baseline score ranking, maybe with some minor modifications.

Of course, this effect can be mitigated by selecting high enough values of the λ parameter in xQuAD, but this would be a simple patch to fix a defective choice of the intent space. Anyway, it becomes clear that more discriminant aspect distributions

would allow xQuAD to use the latent semantics more effectively in order to increase the overall diversity of the system.

For this reason, we introduce a family of filters intended to increase the overall entropy of the aspect distributions.

Definition 4.3.1. *Let $t_1, \ldots, t_n$ be real positive numbers. Let $\alpha \in \mathbb{R}\backslash\{0\}$. We define the α-mean of numbers $t_1, \ldots, t_n$ to be*

$$M_\alpha(t_i) = \left(\frac{\sum_{i=1}^n t_i^\alpha}{n}\right)^{\frac{1}{\alpha}}$$

For $\alpha \in \{0, \pm\infty\}$, we define

$$M_\alpha(t_i) = \lim_{\alpha' \to \alpha} M_\alpha(t_i)$$

Explicitly, we get

$$M_0(t_i) = \sqrt[n]{\prod_{i=1}^n t_i}$$

$$M_{-\infty}(t_i) = \min\{t_i\}$$

$$M_\infty(t_i) = \max\{t_i\}$$

This notion of generalized mean covers some of the most usual "means"

- Arithmetic mean : $\alpha = 0$
- Geometric mean: $\alpha = 1$
- Quadratic mean: $\alpha = 2$
- Harmonic mean: $\alpha = -1$

Taking logarithms and applying Jensen inequality to the logarithm function (which is concave) yields that if $\alpha < \alpha'$ then for every $\{t_i\}$

$$M_\alpha(t_i) \leq M_{\alpha'}(t_i)$$

with equality only when $t_1 = \cdots = t_n$. If $\alpha > 0$, M_α has sense for non-negative t_i. Taking limits when some of the t_i tend to zero, this theorem also holds for any nonnegative choice of t_i.

In particular, we will be interested in taking t_i as a probability distribution over $\mathcal{Z}$. In this case

$$M_1(p(z)) = \frac{1}{K}$$

$$M_\infty(p(z)) \leq 1$$

Thus, for every α and every distribution p, $\frac{1}{K} \leq M_\alpha(p(z)) \leq 1$. Karamata inequality then tells us that for $\alpha > 1$, if a distribution p majorizes a certain distribution q, then $M_\alpha(p) \geq M_\alpha(q)$. This allows us to control the "shape" of the distribution through the value of its α-mean. The higher the value, the more "far" from the uniform and "close" to a one-point distribution ($p(z_i) = \delta_{ij}$ for some j) it is.

Following this idea, we propose the following filter

Definition 4.3.2. *Let $p(z)$ be a probability distribution over $\mathcal{Z}$ and let $\alpha > 0$. We define the α-filter of p, $F_\alpha(p)$, as the probability distribution over $\mathcal{Z}$ given by*

$$F_\alpha(p)(z) = \frac{p(z)^\alpha}{M_\alpha(p(z))^\alpha}$$

It is clear from the definition that $F_\alpha(p)(z) \geq 0$ for all $z \in \mathcal{Z}$ and

$$\sum_{z \in \mathcal{Z}} F_\alpha(p)(z) = \frac{\sum_{z \in \mathcal{Z}} p(z)^\alpha}{M_\alpha(p(z))^\alpha} = 1$$

Thus, $F_\alpha(p)$ is, indeed, a probability distribution over $\mathcal{Z}$. The behavior of the filter depends on whether α is greater or less than 1. For $\alpha = 1$, clearly $F_1(p) = p$. Let us consider $\alpha > 1$. Then, as the function $f(x) = x^\alpha$ is a convex and crescent bijection of the interval $[0, 1]$, the filter must "lower" even more the lowest values of $p(z)$ and increase its higher peaks.

Let us consider a basic example. Let $K = 5$ and consider the following, rather typical, probability distribution

Table 4.1: α-means filter distribution example

z	$p(z)$
1	0.1
2	0.4
3	0.1
4	0.1
5	0.3

In the context of intent spaces, this distribution would typically arise during EM from a document belonging to two different topics (2 and 5). As topic classification is fuzzy and EM can't attain zero values, the aspect distribution retains traces of the other three aspects (1, 3 and 4). We will show the approximate effect of applying the F_2 and F_3 filters, together with the basic cutoff filter F previously described with threshold $\xi = \frac{1}{K} = 0.2$.

Table 4.2: α-mens filter effect example

z	$p(z)$	$F(p)(z)$	$F_2(p)(z)$	$(F \circ F_2)(p)(z)$	$F_3(p)(z)$	$(F \circ F_3)(p)(z)$
1	0.1	0	0.036	0	0.01	0
2	0.4	0.58	0.57	0.64	0.68	0.7
3	0.1	0	0.036	0	0.01	0
4	0.1	0	0.036	0	0.01	0
5	0.3	0.42	0.32	0.36	0.29	0.3

As expected, we observe that for higher α values the filter F_α becomes more incisive, dropping the bottom aspects to almost zero probability and spacing the distribution of the top factors. The example shows how, in contrast to the cutoff

filter, the α-filter redistributes the mass lost by the bottom aspects to the top ones in an exponentially proportional way to the previous value of the distribution. For high α, the most probable aspects receive much more mass than the successive ones, thus increasing the gaps between aspects and remarking the semantic differences between different documents.

The fact that the lower aspects are drop to having almost zero probability makes it natural to apply a cutoff filter after using the α-mean filter.

Finally, we will see that this kind of filter effectively increases the entropy of the distributions, while moving among distributions that adjust the original observed data exactly as well as the original distribution, linking it to tempering.

In TEM algorithm, the last instance of $p(z|x_i, \theta_t)$ is computed as

$$p(z|x_i, \theta_t) = \frac{p(z, x_i|\theta_t)^\beta}{\sum_{z\in\mathcal{Z}} p(z, x_i|\theta_t)^\beta}$$

Let us denote this distribution as p_β. We can write this distribution in terms of the p_1 distribution as follows

$$F_\beta(p_1(z|x_i, \theta_t)) = \frac{p_1(z|x_i, \theta_t)^\beta}{\sum_{z\in\mathcal{Z}} p_1(z|x_i, \theta_t)^\beta} \propto p_1(z|x_i, \theta_t)^\beta = \frac{p(z, x_i|\theta_t)^\beta}{\left(\sum_{z\in\mathcal{Z}} p(z, x_i|\theta_t)\right)^\beta} \propto p(z, x_i|\theta_t)^\beta$$

We already proved that $F_\beta(p_1(z|x_i, \theta_t))$ is normalized, so we get

$$F_\beta(p_1(z|x_i, \theta_t) = \frac{p(z, x_i|\theta_t)}{\sum_{z\in\mathcal{Z}} p(z, x_i|\theta_t)^\beta} = p_\beta(z|x_i, \theta_t) \tag{4.3.2}$$

On the other hand, we have the following lemma

Lemma 4.3.3. *Let α and β be positive real numbers. Then $F_\alpha \circ F_\beta = F_{\alpha\beta}$.*

Proof. Let p be any distribution over a space X. We have

$$F_\alpha(F_\beta(p)) = F_\alpha\left(\frac{p(x)^\beta}{\sum_{x\in X} p(x)^\beta}\right) \propto \left(\frac{p(x)^\beta}{\sum_{x\in X} p(x)^\beta}\right)^\alpha \propto p(x)^{\alpha\beta}$$

As $F_\alpha(F_\beta(p))$ is normalized, we get

$$F_\alpha(F_\beta(p)) = \frac{p(x)^{\alpha\beta}}{\sum_{x\in X} p(x)^{\alpha\beta}} = F_{\alpha\beta}(p)$$

□

Combining this lemma with equation (4.3.2) yields

$$F_\alpha(p_\beta(z|x_i, \theta_t)) = F_\alpha(F_\beta(p_1(z|x_i, \theta_t))) = F_{\alpha\beta}(p_1(z|x_i, \theta_t)) = p_{\alpha\beta}(z|x_i, \theta_t)$$

Therefore, applying an α-mean filter to the aspect distribution of a tempered model is equivalent to executing an E step of the TEM algorithm with inverse computational temperature $\alpha\beta$ instead of β. As $\alpha > 1$, we get a higher value of β (possibly even greater than one). We know that this E step is equivalent to maximizing functional $F_{\alpha\beta}(q, \theta_t)$ with respect to q. Decomposition (4.2.10) of this

functional in EM functional and entropy implies that for higher $\beta \leq 1$ resulting distributions maximize a convex combination of likelihood and entropy, but with lower weight in the entropy term. Thus, less entropic solutions are expected to arise. In some sense, this almost corresponds to starting a new β epoch in the TEM algorithm proposed by Ueda and Nakano (1998), but only updating an E step (the difference being that θ_t and not θ_{t+1} parameters would be used to obtain $p(z|x_i, \theta_{t+1})$).

If $\alpha\beta > 1$, we get the algorithm to directly maximize the entropy. Functional $F_{\alpha\beta}$ is

$$F_{\alpha\beta}(q, \theta_t) = \beta F(q, \theta) - (\alpha\beta - 1)H(q)$$

Therefore, it minimizes the entropy of distributions q_i while maximizing the likelihood of the observed data. This will be the scenario in the experimental results, where we will use $\beta \in (0.6, 1]$ as a typical value and $\alpha = 2, 3$.

In conclusion, the previous arguments prove that the overall entropy of the aspect distributions is increased with respect to TEM solutions by applying α-mean filters to the distributions.

4.4 Folding-in

In Latent Semantic Analysis (Deerwester et al., 1990), folding-in is described as a way to obtain a latent factor approximate representation for documents and queries out of the training corpus. In the LSA approach, this is done applying a linear map to the term distributions of the new document/query using the term-factor matrix computed during the training. In its original paper, **?** uses the probability nature of the computed $p(w|z)$ to compute $p(z|q)$ for a query, incorporating it to the corpus as a document and approximating it through the execution of EM iterations for which only $p(z|q)$ distributions are updated at M steps. As the rest of the parameters are kept fixed, this is essentially equivalent to updating $p(z|q)$ via a Bayesian estimation through the complete data algorithm. Hofmann proposes to use this fold-in query representation in conjunction to the latent indexing to compute matching vector-space models .

In this section we will explore some further variants of classic LSA fold-in methods in order to be able to compare classic query folded-in pLSA to our query-wise RapLSA model. These estimations assume that classic pLSI is applied, so only $p(w|z)$, $p(d|z)$ and $p(z)$ distributions are available in the complete data model. Our objective is to explore suitability of estimates for $p(q|z)$ from Bayesian manipulation of these parameters and some known priors.

While the following list of Bayesian fold-in options is not intended to be completely exhaustive, it provides a taste of the kind of Bayesian inference methods which can be applied to this scenario, and subsequent experimentation will determine the overall effectiveness of these fold-in approximation in comparison with the new RapLSA models.

4.4.1 Query folding estimation

We will consider three principal approaches

1. Probabilistic fold-in via words: We assume that $p(q|w,z) \sim p(q|w)$.

$$p(q|z) \sim \sum_{w \in \mathcal{W}} p(q|w)p(w|z) = \sum_{w \in \mathcal{W}} \frac{p(w|q)p(q)}{p(w)} p(w|z)$$

2. Probabilistic fold-in via documents: We assume that $p(q|d,z) \sim p(q|d)$.

$$p(q|z) \sim \sum_{d \in \mathcal{D}} p(q|d)p(d|z) = \sum_{d \in \mathcal{D}} \frac{p(d|q)p(q)}{p(d)} p(d|z)$$

3. Binary model: We take the event $q|z$ to correspond to the joint extraction of the query terms from the distribution $p(w|z)$.

$$p(q|z) \sim \prod_{w \in q} p(w|z)$$

The first two computations require obtaining some additional distributions, namely $p(w)$, $p(w|q)$, $p(d)$, $p(d|q)$ and $p(q)$. As we lack of biases towards any query, we will consider $p(q) \sim \text{Unif}_{\mathcal{Q}}(q)$. On the other hand, the other distributions need to be approximated, either from the observed data distribution or from the estimated complete data distribution. Estimations which are nearer to the prior are closer to the observed data. Nevertheless, using a model-estimated distribution makes the computation become more compatible with the estimated complete data distribution to which we want to project the query.

4.4.2 Word estimation

If fold-in via words is used, an estimation of a priori distribution $p(w)$ is needed. $\mathcal{W}$ being an observed variable, this can be computed from both the observed and complete data models. The following options are considered

1. Uniform prior: As folded in terms are query terms and there is no priori information about query-specific relevance of these words, taking a uniform word distribution may simply correspond to assuming a neutral bias.

$$p(w) \sim \text{Unif}_{\mathcal{W}}(w)$$

2. Fold-in via query: As only query terms are being folded in, it makes sense to marginalize word distribution from query-word distribution

$$p(w) \sim \sum_{q \in \mathcal{Q}} p(w|q)p(q)$$

3. Fold-in via document: As documents are the main word sources to our model, it makes sense to use them in order to smooth the document distributions

$$p(w) \sim \sum_{w \in \mathcal{D}} p(w|d)p(d)$$

4. Fold-in via aspects: As term-aspect distribution is considered fixed, it is natural to obtain the distribution through them.

$$p(w) \sim \sum_{z \in \mathcal{Z}} p(w|z)p(z)$$

4.4.3 Document estimation

In our models, document prior is assumed to be proportional to the relevance of the document. We can compute this estimated relevance in several ways

1. Uniform prior: If no relevance or further information is used, we can simply take
$$p(d) \sim \text{Unif}_{\mathcal{D}}(d)$$
2. Fold-in through queries: We use document-query prior distribution as a measure of document relevance
$$p(d) \sim \sum_{q \in \mathcal{Q}} p(d|q)p(q)$$
3. Fold-in through aspects: Alike word, if we factor through the complete data model, it makes sense to compute the new document distribution from the document-aspect distribution
$$p(d) \sim \sum_{z \in \mathcal{Z}} p(d|z)p(z)$$

4.4.4 Query language model estimation

The distribution of terms within the query $p(w|q)$ is not a trivial parameter. We can take neutral, non-informative estimations of this distribution, but smoothing and extension through a language model are natural.

1. Uniform prior:
$$p(w|q) \sim \frac{1}{|\{w \in q\}|}$$
2. Fold-in via documents: We can extend the query language model by assuming that words appearing in documents which are relevant to the query likely belong to the query language model
$$p(w|q) \sim \sum_{d \in \mathcal{D}} p(w|d)p(d|q)$$
3. Smoothed uniform prior: A Jelinek-Mercer smoothing is applied to the query term distribution
$$p(w|q) \sim \lambda \frac{n(w,q)}{\sum_{w \in \mathcal{W}} n(w,q)} + (1-\lambda)p(w)$$

4.4.5 Document likelihood estimation

Estimation of distribution $p(d|q)$ is needed for certain models. We will use the smoothed normalized ranking discount prior previously described

$$p(d|q) \sim \lambda \frac{s(\tau(d,q))}{\sum_{d \in \mathcal{D}_q} s(\tau(d,q))} + (1-\lambda)\frac{1}{|\mathcal{D}_q|}$$

4.4.6 Possible different combinations

Here we list all possible different combinations of the previously described techniques. Combinations appearing in this list result in a full approximation of $p(q|z)$ from the basic known distributions, and two combinations are considered equivalent if one of them can be computed from the other with exact Bayesian manipulations (i.e., they lead to the same distribution $p(q|z)$ independently of the starting parameters).

- Fold trough w. Choice of $p(w)$ and $p(w|q)$ or $p(d)$ when suitable.
  - Word model 1: word-query model 1 or 2
  - Word model 2: word-query model 1 or 2
  - Word model 3: document model 1, 2 or 3
  - Word model 4
- Fold through d. Choice of $p(d)$
  - Document model 1
  - Document model 2
  - Document model 3
- Binary model

A pLSA execution over the whole corpus is held. Then fold-in is applied for each query and the aspect distributions $p(q|z)$ are estimated and passed to the diversifier together with the already estimated document mixtures $p(d|z)$ and $p(z|d)$.

4.5 Experimental results

4.5.1 RapLSA effectivenes in search diversity task

We will test the previous models using the diversity qrels from the TREC web trac diversity task datasets from 2009 to 2011. We will use two baseline search engines, Terrier with DLM stemmer and Indri with Porter stemmer. In both cases, a spam filter will be applied at a 70% cutoff level.

For each of the three years, reduced corpus are built from the top 100 documents retrieved by each baseline system for each of the 50 annual queries proposed in TREC diversity task. The effectiveness of each model will be evaluated independently for each annual corpus. IA-Select and xQuAD will be used as diversification algorithms. XQuAD λ parameter will be initially optimized independently for each choice model, year and parameter configuration. Later experimental results will prove that the optimal choice of λ is extremely stable for most of the proposed models and only depends on the baseline selection.

The optimal number of selected aspects for each model has been independently optimized. Explicit experimental methodology for this point will be described in the next chapter. This optimal number has shown to be more model-dependent than baseline-dependent. Results have been essentially equivalent for any of the used

baselines for a fixed given model, but vary a little among models depending on the self-organizing structures of the aspect space. For example, query-wise models like RapLSA or the execution of a single Hofmann pLSA for each query tend to need a much narrow aspect space than a single global execution of pLSA with fold in. The first ones tend to need among 15 and 20 aspects, while the latter attains maximum efficiency for around 100 aspects.

Similarly, the adjustment of the β parameter for tempered models have been initially done independently for each model and baseline. In this case, the choice of higher or lower β parameters do regulate the dynamics and has an analogous effect to the one produced by the learning parameter in neural networks training. Nevertheless, as with the other one, a reasonable choice of β is moreover mode-dependent and not baseline dependent or data dependent. Global choice of the parameter have been found to be specially stable for new models.

As the combinatorics of the experimental data regarding the parametric sweep are enormous, we will omit them from this final report. Instead, we will only present the obtained results for the optimal number of aspects, optimal tempering and optimal expected λ parameter for each model.

Regarding fold in strategies, all the combinatorics described in the previous section have been tested and only the results of the best model of each family is presented here.

In order to simplify experimental references, instead of large descriptions of the explicit system combinations, the following code system will be used

Indri Porter results and pLSA fold in analysis

We present the obtained results for each of the models and the Indri baseline

Figure 4.17: ERR-IA@20 results for Indri baseline and IA-Select unfiltered

	2009		2010		2011		Mean	
Code	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered
0	0,166893	0,166893	0,208419	0,208419	0,410715	0,410715	0,262009	0,262009
1	0,158133	0,158133	0,233351	0,233351	0,369415	0,369415	0,253633	0,253633
70	0,173972	0,171987	0,2433	0,218376	0,381426	0,41426	0,26623267	0,26820767
720	0,179271	0,169202	0,248561	0,215123	0,386763	0,41408	0,27153167	0,266135
7204	0,12374	0,16688	0,24414	0,208163	0,332255	0,410515	0,23337833	0,26185267
73	0,174871	0,166827	0,258747	0,208403	0,385456	0,410715	0,27302467	0,26198167
611	0,157691	0,151443	0,212631	0,18682	0,355812	0,317587	0,24204467	0,21861667
612	0,151426	0,165785	0,19705	0,233936	0,338882	0,333103	0,22911933	0,24427467
62	0,139447	0,161788	0,194814	0,249238	0,341534	0,301596	0,225265	0,23754067
622	0,152084	0,183202	0,203262	0,215474	0,363818	0,352631	0,23972133	0,25043567
6114	0,160869	0,160869	0,184511	0,184511	0,313244	0,313244	0,21954133	0,21954133
6124	0,149859	0,149859	0,150147	0,150147	0,343894	0,343894	0,21463333	0,21463333
6214	0,16432	0,16432	0,208276	0,208276	0,34473	0,34473	0,23910867	0,23910867
6224	0,160953	0,160953	0,207135	0,207135	0,346363	0,346363	0,23815033	0,23815033
25	0,184315	0,174119	0,206525	0,199771	0,387776	0,382672	0,25953867	0,25218733
32	0,166706	0,197668	0,250251	0,234868	0,413787	0,381616	0,27691467	0,271384
323	0,17645	0,17627	0,248861	0,255062	0,39072	0,404334	0,27201033	0,27855533
322	0,135063	0,172637	0,25738	0,20887	0,359603	0,40702	0,250682	0,26284233
325	0,158055	0,150656	0,241419	0,25342	0,345085	0,345661	0,24818633	0,24991233
33	0,160651	0,175793	0,243569	0,217967	0,403301	0,414011	0,26917367	0,269257

Table 4.3: Model code descriptions

Code	**Model**
0	Baseline
1	ODP
25	Model 2 passing $p(z\|d,q)$ to xQuAD
32	RapLSA
322	RapLSA + α-mean filtering
323	RapLSA passing $p(z\|d,q)$ to xQuAD
325	RapLSA + α-mean filtering passing $p(z\|d,q)$ to xQuAD
33	RapLSA with α-mean filtering in E step ($\beta > 1$)
611	Hofmann + word fold in
612	Hofmann + document fold in
62	Hofmann + query word fold in
622	Hofmann + query word fold in with Jelinek Mercer
6114	Hofmann + word fold in + α-mean filtering
6124	Hofmann + document fold in + α-mean filtering
624	Hofmann + query word fold in + α-mean filtering
6224	Hofmann + query word fold in with Jelinek Mercer + α-mean filtering
70	Hofmann executed query-wise
720	pLSI executed query-wise
7204	pLSI executed query-wise+ α-mean filtering
73	pLSI executed query-wise+ passing $p(z\|d,q)$ to xQuAD

Figure 4.18: ERR-IA@20 results for Indri baseline and xQuAD unfiltered

	2009		2010		2011		Mean	
Code	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered
0	0,166893	0,166893	0,208419	0,208419	0,410715	0,410715	0,262009	0,262009
1	0,166893	0,166893	0,207057	0,207057	0,410717	0,410717	0,26155567	0,26155567
70	0,166893	0,166893	0,219336	0,21891	0,410715	0,410715	0,265648	0,265506
720	0,173929	0,17368	0,25605	0,256109	0,425101	0,425932	0,28502667	0,28524033
7204	0,166893	0,173345	0,271869	0,254911	0,427823	0,425774	0,28886167	0,28467667
73	0,166893	0,166893	0,261795	0,243122	0,410715	0,410715	0,279801	0,27357667
611	0,168807	0,166893	0,211915	0,209541	0,410715	0,410715	0,26381233	0,262383
612	0,166893	0,167993	0,211606	0,218516	0,410731	0,410737	0,26307667	0,26574867
62	0,166893	0,16839	0,208845	0,224525	0,410732	0,410715	0,26215667	0,26787667
622	0,166893	0,169105	0,209845	0,209879	0,410731	0,410715	0,26248967	0,263233
6114	0,166893	0,166893	0,218168	0,218168	0,410715	0,410715	0,26525867	0,26525867
6124	0,166893	0,166893	0,217209	0,217209	0,410715	0,410715	0,264939	0,264939
6214	0,167117	0,167117	0,218268	0,218268	0,410715	0,410715	0,26536667	0,26536667
6224	0,166893	0,166893	0,205669	0,205669	0,410715	0,410715	0,26109233	0,26109233
25	0,173175	0,172897	0,21173	0,217286	0,413101	0,410715	0,266002	0,266966
32	0,166893	0,167732	0,240787	0,248301	0,410748	0,410757	0,27280933	0,27559667
323	0,17389	0,173778	0,255978	0,256472	0,425181	0,425401	0,28501633	0,285217
322	0,166894	0,166893	0,238174	0,246374	0,410874	0,410876	0,27198067	0,27471433
325	0,166893	0,172739	0,262887	0,248245	0,410784	0,410715	0,280188	0,277233
33	0,166893	0,180087	0,242087	0,257124	0,410874	0,418568	0,27328467	0,28525967

Figure 4.19: ERR-IA@20 results for Indri baseline and IA-Select filtered

	2009		2010		2011		Mean	
Code	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered
0	0,166893	0,166893	0,208419	0,208419	0,410715	0,410715	0,262009	0,262009
1	0,158133	0,158133	0,233351	0,233351	0,369415	0,369415	0,253633	0,253633
70	0,181589	0,181466	0,251576	0,222102	0,410546	0,41426	0,281237	0,27260933
720	0,179271	0,169202	0,254686	0,215123	0,410156	0,41408	0,281371	0,266135
7204	0,166723	0,168894	0,24414	0,208942	0,410126	0,410715	0,273663	0,26285033
73	0,174871	0,166917	0,263543	0,208539	0,410934	0,410715	0,283116	0,262057
611	0,166005	0,16708	0,213204	0,208118	0,4117	0,411124	0,26363633	0,26210733
612	0,166909	0,165785	0,208763	0,233936	0,41093	0,410778	0,26220067	0,27016633
62	0,166951	0,166074	0,20833	0,249238	0,410729	0,41127	0,26200333	0,27552733
622	0,16695	0,183202	0,209426	0,216685	0,411289	0,411426	0,262555	0,27043767
6114	0,166528	0,166528	0,210798	0,210798	0,410324	0,410324	0,26255	0,26255
6124	0,166656	0,166656	0,210078	0,210078	0,411851	0,411851	0,26286167	0,26286167
6214	0,166865	0,166865	0,209606	0,209606	0,410748	0,410748	0,26240633	0,26240633
6224	0,166263	0,166263	0,22141	0,22141	0,411173	0,411173	0,266282	0,266282
25	0,184315	0,174119	0,209929	0,208444	0,410923	0,410858	0,268389	0,26447367
32	0,166706	0,198415	0,256357	0,240281	0,413787	0,411222	0,27895	0,283306
323	0,17645	0,179793	0,26881	0,258992	0,409967	0,4109	0,28507567	0,28322833
322	0,167289	0,177412	0,25738	0,209984	0,41074	0,410515	0,27846967	0,26597033
325	0,165831	0,166172	0,241419	0,25342	0,4098	0,410858	0,27235	0,27681667
33	0,166193	0,175793	0,248349	0,218265	0,410642	0,414329	0,27506133	0,26946233

Figure 4.20: ERR-IA@20 results for Indri baseline and xQuAD filtered

	2009		2010		2011		Mean	
Code	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered
0	0,166893	0,166893	0,208419	0,208419	0,410715	0,410715	0,262009	0,262009
1	0,166893	0,166893	0,207057	0,207057	0,410717	0,410717	0,26155567	0,26155567
70	0,171318	0,167902	0,228382	0,224047	0,410715	0,419697	0,27013833	0,27054867
720	0,174076	0,17368	0,276988	0,274551	0,426101	0,428936	0,29238833	0,292389
7204	0,18169	0,173345	0,277716	0,254911	0,427823	0,425774	0,295743	0,28467667
73	0,166893	0,166893	0,261795	0,243122	0,410769	0,410745	0,279819	0,27358667
611	0,168815	0,166983	0,211924	0,209888	0,410716	0,410732	0,26381833	0,26253433
612	0,166893	0,167993	0,215609	0,218725	0,410739	0,410737	0,26441367	0,26581833
62	0,166893	0,168472	0,210638	0,224838	0,410732	0,410715	0,26275433	0,26800833
622	0,166919	0,169245	0,210416	0,210068	0,410732	0,410732	0,262689	0,26334833
6114	0,16715	0,16715	0,218168	0,218168	0,410715	0,410715	0,26534433	0,26534433
6124	0,16731	0,16731	0,220524	0,220524	0,410732	0,410732	0,26618867	0,26618867
6214	0,167117	0,167117	0,220866	0,220866	0,410732	0,410732	0,26623833	0,26623833
6224	0,166893	0,166893	0,209623	0,209623	0,410732	0,410732	0,262416	0,262416
25	0,173181	0,172897	0,22199	0,220228	0,413828	0,412916	0,26966633	0,26868033
32	0,166896	0,170196	0,240787	0,248301	0,41505	0,411246	0,27424433	0,276581
323	0,17389	0,173778	0,258948	0,269916	0,425181	0,425401	0,28600633	0,28969833
322	0,167606	0,167993	0,240614	0,247322	0,416865	0,412462	0,27502833	0,27592567
325	0,171486	0,172739	0,262887	0,248245	0,428282	0,4107909	0,28755167	0,2772583
33	0,167842	0,180122	0,242087	0,26681	0,41559	0,418568	0,275173	0,2885

As it was expected, we observe that fold in strategies develop worse diversification results than executing single independent pLSA over each individual query. The main difference is query-specificity of the constructed intent spaces. While global pLSI may get better semantic representations of documents, the extracted aspect spaces would lay in a different semantic level than the desired query intents. Therefore, query-wise executed pLSA unfolds query ambiguity more precisely than any of the tested fold-in strategy. This been said, if new documents or queries

were incorporated to the corpus, the only available tools for representing them in the aspect space would be precisely the described fold-in strategies. Among them, experimental data suggests than word fold in via aspects (Model 4) is the most effective one. Improvement of its performance by means of α-mean filtering has been observed, but is not conclusive.

In order to facilitate the comparison of the main models, result tables will be presented with the comparison between most relevant models:

Figure 4.21: Summarized ERR-IA@20 results for Indri baseline and IA-Select unfiltered

	2009		2010		2011		media	
Code	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered
0	0,166893	0,166893	0,208419	0,208419	0,410715	0,410715	0,262009	0,262009
1	0,158133	0,158133	0,233351	0,233351	0,369415	0,369415	0,253633	0,253633
70	0,173972	0,171987	0,2433	0,218376	0,381426	0,41426	0,26623267	0,26820767
720	0,179271	0,169202	0,248561	0,215123	0,386763	0,41408	0,27153167	0,266135
7204	0,12374	0,16688	0,24414	0,208163	0,332255	0,410515	0,23337833	0,26185267
73	0,174871	0,166827	0,258747	0,208403	0,385456	0,410715	0,27302467	0,26198167
611	0,157691	0,151443	0,212631	0,18682	0,355812	0,317587	0,24204467	0,21861667
32	0,166706	0,197668	0,250251	0,234868	0,413787	0,381616	0,27691467	0,271384
323	0,17645	0,17627	0,248861	0,255062	0,39072	0,404334	0,27201033	0,27855533
322	0,135063	0,172637	0,25738	0,20887	0,359603	0,40702	0,250682	0,26284233
325	0,158055	0,150656	0,241419	0,25342	0,345085	0,345661	0,24818633	0,24991233
33	0,160651	0,175793	0,243569	0,217967	0,403301	0,414011	0,26917367	0,269257

Figure 4.22: Summarized ERR-IA@20 results for Indri baseline and xQuAD unfiltered

	2009		2010		2011		media	
Code	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered
0	0,166893	0,166893	0,208419	0,208419	0,410715	0,410715	0,262009	0,262009
1	0,166893	0,166893	0,207057	0,207057	0,410717	0,410717	0,26155567	0,26155567
70	0,166893	0,166893	0,219336	0,21891	0,410715	0,410715	0,265648	0,265506
720	0,173929	0,17368	0,25605	0,256109	0,425101	0,425932	0,28502667	0,28524033
7204	0,166893	0,173345	0,271869	0,254911	0,427823	0,425774	0,28886167	0,28467667
73	0,166893	0,166893	0,261795	0,243122	0,410715	0,410715	0,279801	0,27357667
611	0,168807	0,166893	0,211915	0,209541	0,410715	0,410715	0,26381233	0,262383
32	0,166893	0,167732	0,240787	0,248301	0,410748	0,410757	0,27280933	0,27559667
323	0,17389	0,173778	0,255978	0,256472	0,425181	0,425401	0,28501633	0,285217
322	0,166894	0,166893	0,238174	0,246374	0,410874	0,410876	0,27198067	0,27471433
325	0,166893	0,172739	0,262887	0,248245	0,410784	0,410715	0,280188	0,277233
33	0,166893	0,180087	0,242087	0,257124	0,410874	0,418568	0,27328467	0,28525967

Figure 4.23: Summarized ERR-IA@20 results for Indri baseline and IA-Select filtered

	2009		2010		2011		media	
Code	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered
0	0,166893	0,166893	0,208419	0,208419	0,410715	0,410715	0,262009	0,262009
1	0,158133	0,158133	0,233351	0,233351	0,369415	0,369415	0,253633	0,253633
70	0,181589	0,181466	0,251576	0,222102	0,410546	0,41426	0,281237	0,27260933
720	0,179271	0,169202	0,254686	0,215123	0,410156	0,41408	0,281371	0,266135
7204	0,166723	0,168894	0,24414	0,208942	0,410126	0,410715	0,273663	0,26285033
73	0,174871	0,166917	0,263543	0,208539	0,410934	0,410715	0,283116	0,262057
611	0,166005	0,16708	0,213204	0,208118	0,4117	0,411124	0,26363633	0,26210733
32	0,166706	0,198415	0,256357	0,240281	0,413787	0,411222	0,27895	0,283306
323	0,17645	0,179793	0,26881	0,258992	0,409967	0,4109	0,28507567	0,28322833
322	0,167289	0,177412	0,25738	0,209984	0,41074	0,410515	0,27846967	0,26597033
325	0,165831	0,166172	0,241419	0,25342	0,4098	0,410858	0,27235	0,27681667
33	0,166193	0,175793	0,248349	0,218265	0,410642	0,414329	0,27506133	0,26946233

Figure 4.24: Summarized ERR-IA@20 results for Indri baseline and xQuAD filtered

	2009		2010		2011		media	
Code	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered
0	0,166893	0,166893	0,208419	0,208419	0,410715	0,410715	0,262009	0,262009
1	0,166893	0,166893	0,207057	0,207057	0,410717	0,410717	0,26155567	0,26155567
70	0,171318	0,167902	0,228382	0,224047	0,410715	0,419697	0,27013833	0,27054867
720	0,174076	0,17368	0,276988	0,274551	0,426101	0,428936	0,29238833	0,292389
7204	0,18169	0,173345	0,277716	0,254911	0,427823	0,425774	0,295743	0,28467667
73	0,166893	0,166893	0,261795	0,243122	0,410769	0,410745	0,279819	0,27358667
611	0,168815	0,166983	0,211924	0,209888	0,410716	0,410732	0,26381833	0,26253433
32	0,166896	0,170196	0,240787	0,248301	0,41505	0,411246	0,27424433	0,276581
323	0,17389	0,173778	0,258948	0,269916	0,425181	0,425401	0,28600633	0,28969833
322	0,167606	0,167993	0,240614	0,247322	0,416865	0,412462	0,27502833	0,27592567
325	0,171486	0,172739	0,262887	0,248245	0,428282	0,4107909	0,28755167	0,2772583
33	0,167842	0,180122	0,242087	0,26681	0,41559	0,418568	0,275173	0,2885

We observe that the new proposed models lead to overall better results for almost all the considered scenarios when. In particular, either query-wise RapLSI or basic RapLSA improve query-wise Hofmann in each considered scenario.

Taking the simplest of the models, i.e. untempered unfiltered RapLSI, it presents a consistent significant average improvement of 7,4% with respect to query-wise Hofmann results. If additional filtering is taken into account for both models, the average improvement increases to 8.8%. In particular, 2010 queries results get improved by more than 22%.

Terrier DLM results

Figure 4.25: ERR-IA@20 results for Terrier baseline and IA-Select unfiltered

	2009		2010		2011		media	
Code	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered
0	0,182594	0,182594	0,216493	0,216493	0,392195	0,392195	0,26376067	0,26376067
1	0,158133	0,158133	0,233351	0,233351	0,369415	0,369415	0,253633	0,253633
70	0,17508	0,182506	0,251331	0,222728	0,372662	0,394645	0,26635767	0,26662633
720	0,183331	0,183549	0,265483	0,221756	0,391405	0,394336	0,280073	0,266547
7204	0,153769	0,1825	0,256612	0,216491	0,32057	0,392093	0,24365033	0,26369467
73	0,18832	0,182343	0,273957	0,216462	0,38074	0,392091	0,28100567	0,263632
611	0,178316	0,171145	0,202132	0,186006	0,315203	0,30916	0,23188367	0,22210367
32	0,173224	0,181512	0,23343	0,236914	0,400433	0,383527	0,269029	0,26731767
323	0,181075	0,173439	0,262178	0,276795	0,37036	0,382466	0,27120433	0,27756667
322	0,16437	0,193086	0,236904	0,215767	0,257522	0,397199	0,21959867	0,268684
325	0,130572	0,130666	0,272227	0,217735	0,311873	0,320306	0,238224	0,22290233
33	0,203466	0,193948	0,270615	0,218612	0,361858	0,393634	0,27864633	0,26873133

Figure 4.26: ERR-IA@20 results for Terrier baseline and xQuAD unfiltered

	2009		2010		2011		media	
Code	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered
0	0,182594	0,182594	0,216493	0,216493	0,392195	0,392195	0,26376067	0,26376067
1	0,158133	0,158133	0,233351	0,233351	0,369415	0,369415	0,253633	0,253633
70	0,182744	0,182778	0,222549	0,22082	0,395736	0,39642	0,26700967	0,26667267
720	0,182594	0,182594	0,265363	0,262683	0,398497	0,394026	0,28215133	0,27976767
7204	0,184964	0,182594	0,280389	0,261948	0,394882	0,398477	0,286745	0,28100633
73	0,182594	0,185815	0,271244	0,245476	0,394478	0,394028	0,282772	0,27510633
611	0,186537	0,182594	0,2191	0,217437	0,392777	0,393086	0,266138	0,26437233
32	0,182689	0,184916	0,25436	0,257045	0,394555	0,394382	0,27720133	0,278781
323	0,182594	0,182594	0,264305	0,266027	0,398511	0,398172	0,28180333	0,28226433
322	0,182594	0,182594	0,237367	0,242122	0,392195	0,392195	0,27071867	0,27230367
325	0,182645	0,185472	0,266445	0,242811	0,400302	0,403122	0,28313067	0,277135
33	0,184856	0,185672	0,260139	0,253782	0,39424	0,394631	0,279745	0,27802833

Figure 4.27: ERR-IA@20 results for Terrier baseline and IA-Select filtered

	2009		2010		2011		media	
Code	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered
0	0,182594	0,182594	0,216493	0,216493	0,392195	0,392195	0,26376067	0,26376067
1	0,158133	0,158133	0,233351	0,233351	0,369415	0,369415	0,253633	0,253633
70	0,182975	0,183203	0,251331	0,227357	0,392371	0,394645	0,275559	0,26840167
720	0,183331	0,183549	0,265483	0,240619	0,392574	0,394336	0,28046267	0,27283467
7204	0,182918	0,182986	0,256612	0,216503	0,392332	0,392195	0,27728733	0,26389467
73	0,18832	0,183281	0,273957	0,216882	0,392422	0,392278	0,28489967	0,264147
611	0,184027	0,18174	0,218651	0,216572	0,392777	0,39201	0,26515167	0,26344067
32	0,183159	0,18264	0,242447	0,236914	0,400433	0,391882	0,27534633	0,27047867
323	0,184053	0,182366	0,266882	0,281978	0,392834	0,392478	0,28125633	0,28560733
322	0,182132	0,193792	0,244454	0,216691	0,394088	0,398534	0,273558	0,26967233
325	0,183308	0,181595	0,277243	0,219621	0,392621	0,393145	0,28439067	0,264787
33	0,203466	0,193948	0,270615	0,218612	0,392956	0,393634	0,28901233	0,26873133

Figure 4.28: ERR-IA@20 results for Terrier baseline and xQuAD filtered

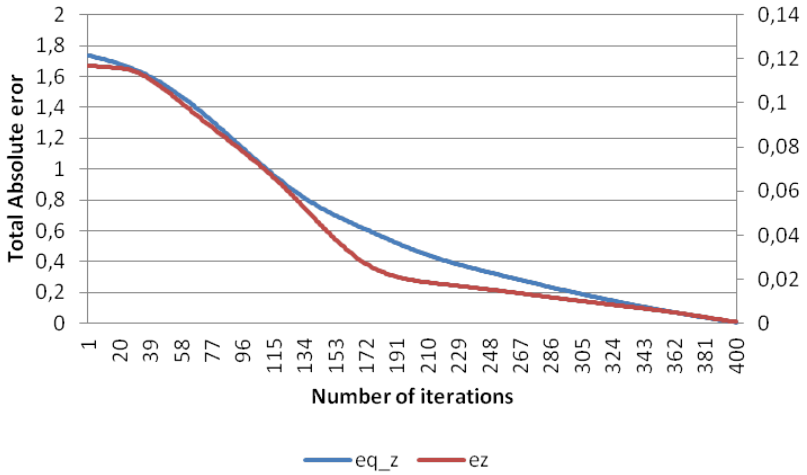
	2009		2010		2011		media	
Code	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered	Untempered	Tempered
0	0,166893	0,166893	0,208419	0,208419	0,410715	0,410715	0,262009	0,262009
1	0,166893	0,166893	0,207057	0,207057	0,410717	0,410717	0,26155567	0,26155567
70	0,171318	0,167902	0,228382	0,224047	0,410715	0,419697	0,27013833	0,27054867
720	0,174076	0,17368	0,276988	0,274551	0,426101	0,428936	0,29238833	0,292389
7204	0,18169	0,173345	0,277716	0,254911	0,427823	0,425774	0,295743	0,28467667
73	0,166893	0,166893	0,261795	0,243122	0,410769	0,410745	0,279819	0,27358667
611	0,168815	0,166983	0,211924	0,209888	0,410716	0,410732	0,26381833	0,26253433
32	0,166896	0,170196	0,240787	0,248301	0,41505	0,411246	0,27424433	0,276581
323	0,17389	0,173778	0,258948	0,269916	0,425181	0,425401	0,28600633	0,28969833
322	0,167606	0,167993	0,240614	0,247322	0,416865	0,412462	0,27502833	0,27592567
325	0,171486	0,172739	0,262887	0,248245	0,428282	0,4107909	0,28755167	0,2772583
33	0,167842	0,180122	0,242087	0,26681	0,41559	0,418568	0,275173	0,2885

The results for the new Terrier Baseline seem to reinforce the previous observations. In particular, filtered versions of query-wise RapLSI outperforms every filtered version of query-wise pLSA by an average 8.7%. Again, 2010 queries get extraordinary improvement. Tempered RapLSI outperforms tempered pLSA by almost a 30% in the ERR-IA@20 value.

Convergence results

In this section we show a typical convergence curve for the RapLSA model. Data has been taken for a complete convergent run over data coming from the Indri baseline.

Figure 4.29: Convergence results for RapLSA. Normalized total absolute difference of $p(q|z)$ and $p(z)$ between iterations



The convergence becomes clear. We observe that the absolute difference decreases almost linearly from approximately 50 to 200 iterations. Nevertheless, the

differences in the final diversity results have not been found to be significant among different choices of the maximum number of iterations. As it could be expected, the minimum number of steps for convergence has been found to depend slightly on the number of aspects. A greater number of latent factors lead to an increased number of parameters for the model, and therefore, to an increased number of local minima presence in the phase space of the algorithm. A combination of lower beta tempering and an increased number of iterations is recommended in order to attain better final diversity results.

4.5.2 Other models

Some basic tests have been carried out for instances of the general model apart from the main RapLSA application to search diversity. These tests aim to show the potential of some of some of the models

Two main tests have been taken, regarding the qualitative properties of the producer-consumer search model and the RapLSA application to recommendation diversity.

The first of the tests was carried under the experimental setup for search diversity described in the last section. A basic smooth word-counting approximation was taken for the query language model $\tilde{p}(w|q)$. Utility distributions $\tilde{p}(w,d)$ and $\tilde{p}(d,q)$ were estimated as in the RapLSI model, considering uniform q distribution. RapLSA producer–consumer model ((4.2.14) and (4.2.15)) is applied to extract an aspect space. While the initial experiments carried out didn't show significant improvement in diversity with respect to the results of the already optimized RapLSA variants described in the last section, some interesting properties of the aspect space were found.

In contrast to other models, the underlying complete data model of this version of the algorithm is completely symmetric. In contrast to RapLSA, where the aspect space was coupled with the query variable, the producer-consumer model builds a unique common aspect space for all queries. In the absence of a proper query language model grouping together query dependencies in model 4.13, it is natural that the developed aspect space looses specificity with respect to the query, therefore leading to diversity results being worse than those of other query-specific models. On the other hand, this opens up the possibility of exploiting inter-query information.

In order to explore the aspect distribution dynamics, we looked at the obtained filtered aspect distributions after a couple of EM iterations (before complete a convergence is obtained). We observed that the query aspect distribution grouped the queries into narrow clusters having between 2 and 4 queries each. As expected, many of the clusters seemed to be more or less fuzzy, with queries belonging to various clusters. Nevertheless, a pair of "liked queries" was found while analyzing 2011 results. Queries 127, "dutchess county tourism" and 137, "rock and gem shows" belong to a single common aspect from a pool of 20 possible ones. When analyzing the content of some document results for query 127, we found that some "dutchess county tourism" main pages contained announcements of geology activities, thus being content-related to query 137.

As a curious fact, searching in a commercial engine shows that Dutchess County hosts the Mid-Hudson Gem and Mineral Society annual gem and mineral show and

sale. Moreover, links to the events exist in main pages related to the Dutchess County tourism office. Of course, this initial results are not significant and an exhaustive experimental setup is to be made to understand the dynamics of the inter-query information within the system, but the described preliminary results point to the model being able to abstract such common information.

On the other hand, the proposed recommendation diversity model 4.5 have been shown to outperform pLSA as an aspect extraction method under certain conditions. The following exploratory results are obtained over the MovieLens 1M dataset. Hofmann pLSA version is used as a baseline recommender. ERR-IA@20 is computed for four possible recommendation setups:

- pLSA baseline is used as a the final recommender.
- A second execution of pLSA is used to build an intent space, and pLSA baseline is diversified using xQuAD with the subtopic data coming from that second execution.
- pLSA baseline score is fed to RapLSA, which is used as the final recommender.
- pLSA baseline secore is fed to RapLSA. The baseline is then diversified using xQuAD with the subtopic data coming from RapLSA.

50 aspects are used for both the pLSA baseline and RapLSA. For simplicity, both versions are taken untempered and a fixed $\lambda = 0.5$ is used by xQuAD.

Table 4.4: ERR-IA@20 comparison between pLSA and RapLSA as baseline recommenders or aspect space extractors

	Direct Recommendation	**Diversified results via xQuAD**
pLSA	0.149427	0.149708
RapLSA	0.149704	0.151589

We observe that in this scenario RapLSA improves pLSA diversity performance both as a baseline recommender and as an aspect extraction algorithm.

However, the tests correspond to preliminary exploratory experiments. No significant results were found. PLSA aspect space has been taken as optimal for the untempered scenario, but an analysis of the RapLSA dependency on parameters is to be performed. Moreover, scoring normalization variants, aspect filtering and tempering has not been tested, and are expected to improve the overall effectiveness of the algorithm. An exhaustive experimental setup, comparing the effectiveness of both systems as diversifiers over other fixed recommendation baselines such as KNN would be necessary.

It becomes clear that more experiments are needed to evaluate significantly the precise performance of these alternative models, but the shown preliminary results seem to indicate their diversity enhancing potential.

Chapter 5

Optimization of the aspect space size for diversity enhancement

Past chapters have shown the high potential of automatic query intent space extraction algorithms as a source of suitable aspect spaces for diversification tasks. The choice of appropriate aspects clearly impact the final diversity of the system, but, while being a core problem in diversity problems, there is little explicit research on the desirable properties of the space of extracted aspects, in terms of best enabling an effective diversification.

In this chapter we address a primary one among such properties, namely the size of the aspect space. We study the impact of the number of aspects on the diversification behavior in terms of two effects:

- The expected amount of change in the ranking resulting from diversification.
- The degree of enabled quality enhancement, in terms of diversity evaluation metrics.

We will study the problem both analytically and empirically. We will define predictive models describing the aspect dynamics of two families of diversifiers, including some of the most common diversification algorithms in search tasks, like IA-Select and xQuAD. Under certain simplifying hypothesis about the generating models of the aspect extraction algorithm and the true query subtopics (or real aspects), we will determine the expected distributions of both extracted and true aspects within the diversified ranking precisely. This will allow us to derive formal predictions on how the expected ranking distances and diversity quality evolve with the number of aspects. Particularly, we will derive (semi)closed exact combinatoric formulas for both the expected distance from the baseline to the diversified ranking and the expected value of certain diversity metrics, like subtopic recall and ERR-IA.

Analytic predictions within the simplified model will be tested against Monte Carlo simulations. Finally, measures of the studied distances ranging over the size of the aspect space will be taken in an experiment on the TREC diversity task. The empirical results show a fair correspondence to some of the theoretical models,

thus providing evidence of the predictive value of the analytical approach, and its potential usefulness to guide the configuration of the query aspect space size.

5.1 Diversity prediction models

In order to study the effect of the aspect space on the effectiveness of diversification algorithms, two different settings could be considered, according to whether the query aspects for which diversity is evaluated are known by the diversification system or not. Scenarios where aspects are known have been considered in such experiments as presented in Agrawal et al. (2009) or Vargas et al. (2012b), where diversity is both targeted and measured, for instance, in terms of ODP categories. The system does not necessarily know the right association of aspects to documents and queries, but it does know the space where aspects range. A different and more general scenario is set up in the TREC diversity task, where the subtopics for evaluation are hidden from the systems, and extracting an aspect space is part of the diversifier task (Santos et al., 2010). We shall focus on this second situation, where the choice of the aspect space for diversification clearly belongs to the system designer. Moreover, as we shall see, the first scenario can be treated as a particular case of the second one. Thus, the research question we aim to address now is how diversity evolves with the number of extracted aspects.

Predicting the optimal size for maximizing the effectiveness of a diversification algorithm is an important unsolved problem. Depending on the aspect extraction system, experimental adjustment on this parameter can be quite expensive. Suppose that we want to optimize the aspect size dimension for an extraction algorithm running on linear time on the number of aspects, such as pLSA or RapLSA. A blind check for the optimal parameter for a number of aspects ranging from 1 to N would have a quadratic cost on N. If other parameters such as the inverse computational temperature β for a tempered algorithm are to be adjusted simultaneously, and we add the uncertainty coming from pLSA random initial choice of parameters, a blind optimal choice for the aspect size can become intractable.

As a first approximation, we will omit some of the complex retrieved aspect-true aspect dynamics existing between the aspect extraction algorithm and the diversifier system. We will also simplify inherent biases coming from the corpus data, the extraction algorithm and the diversifier, resulting in an expected "neutral" state of the aspect distributions. Moreover, we will focus on pure diversity, omitting document relevance considerations.

Assuming an idealized generative model for aspect distributions, we will develop exact analytic expressions for the "potential diversity" of an aspect space, measured as the expected Kendall distance between the baseline and the diversified ranking (equations (5.1.2) and (5.1.3)), the expected subtopic recall of the baseline (5.1.4), or various families of diversifiers (5.1.7)(5.1.8) and the expected ERR-IA for a random diversifier (5.1.9).

Moreover, a correction term will be derived to adjust metric predictions when nonideal system recall conditions are present (5.1.11). Finally, a general procedure for computing analytically the probability of a diversification system being better than random in terms of a fixed metric is described, given that the state distribution

for the metric values is available (5.1.12).

5.1.1 Generative model for aspect distributions

Let us assume that we want to diversify a baseline ranking consisting on D documents. An intent approach will be used to measure diversity. Let us assume that a set of true aspects A_T with T true aspects is used for evaluation. An aspect extraction algorithm is used to approximate the true aspect space. A set A_S of system aspects is generated and document-aspect distributions are computed. As a common notation, we will denote the l-th document in the baseline ranking as d_l, starting with d_0, i.e., the baseline ranking is $(d_0, d_1, \ldots, d_{D-1})$.

Then a diversification algorithm is applied to re-rank the initial baseline result using the system aspect space A_S obtained from the aspect extraction algorithm. Finally, the diversified ranking is evaluated using the original true aspects A_T.

The following additional ideal assumptions will be made:

1. The baseline ranking function is constant over the D considered documents. In order to focus on pure diversity, we will consider a neutral baseline.

2. All retrieved documents are relevant, i.e., the baseline algorithm has perfect precision.

3. Each document has a unique true aspect, i.e., for each document $d \in D$, there exist a unique $a \in A_T$ such that $p(a|d) > 0$. Therefore, the set of documents are partitioned in A_T classes depending on their true aspect. We will denote the true aspect of a document d by $c_T(d)$.

4. The aspect extraction algorithm assigns each document $d \in D$ a single system aspect $a \in A_S$. We will denote it by the system aspect of document d, $c_S(d)$.

Assuming the previous We will consider the following idealized diversification algorithms

Definition 5.1.1. *A diversification algorithm of type 1 selects the first document with each of the system aspects $a \in A_S$ and reranks them in the first S positions of the ranking, preserving their relative order. Then it repositions the rest of the documents in the remaining $D - S$ positions with the same relative order.*

Definition 5.1.2. *A diversification algorithm of type 2 repeats type 1 procedure successively with the remaining documents after extracting the first S distinct aspects. More precisely, the algorithm chooses the document for the position j as the unselected one appearing in the highest position in the baseline ranking such that its aspect is the lest frequent among the first $j - 1$ documents.*

IA-Select is an example of diversification type 1, xQuAD, round-robin, and MMR with a distance function based on document aspects (such as Jaccard, cosine, or Pearson correlation) are examples of type 2.

As for the dynamics of the relation between A_S and A_T, we will consider two degrees of generality in our work.

The first simpler case corresponds to considering $A_S = A_T = A$. This situation will be assumed for the computation of the expected value of the Kendall distance between baseline and diversified rankings, and allows us to obtain some quantitative information about the "shape" of the metric-aspect size curves.

For a second more precise model describing true aspects, system aspects and the probabilistic dependencies between each of them and the documents, we consider the following abstraction. The purpose of aspect extraction is to approximate the unseen true aspects as closely as possible. The ideal aspect extraction result – the most effective one in terms of the re-ranking quality it enables – would be one which makes a perfect guess of true aspects, i.e. one that achieves a one to one correspondence between the extracted system aspects and the hidden true aspects. This is obviously not achieved in the general case (although quite good approximations have been found by TREC participants), but a certain dependency between system and true aspects needs to be obtained for diversification to achieve any kind of improvement – if the system and true aspects of all documents are mutually independent, the effect of diversification is naturally the same as random re-ranking.

We model this dependency as follows. Given a document d, its system aspect $c_S(d)$ does not determine with certainty what its true aspect $c_T(d)$ is. However, we assume that each system aspect s will tend to correspond to a certain specific true aspect t more often than to other ones. That is, some true aspect t (let us refer to it as preferential) will occur more frequently than others in the set of documents that have the s system aspect (thus lending the dependency between system and true aspects which makes diversification effective). We model this by a mapping $\varphi : A_S \to A_T$ between true and system aspects, where $\varphi(s)$ is the preferential true aspect of s, and we model the dependency between them as a certain constant probability p that $\varphi(s)$ occurs when s occurs. In case $\varphi(s)$ does not occur (a case with $1 - p_0$ probability), then any true aspect may occur with uniform probability.

This can be summarized formally as follows: given $d \in D$ and $t \in A_T$

$$p(c_T(d) = t | c_S(d) = s) = \begin{cases} p_0 & \text{if } t = \varphi(s) \\ \frac{1-p_0}{T-1} & \text{otherwise} \end{cases}$$

In order to simplify the ongoing analysis, we will reparameterize this intuitive idea in the following way. With probability p, the system "fails" to classify each document d. In that case, its true aspect is drawn from an uniform distribution over A_T independently of its system aspect. Then

$$p(c_T(d) = t | c_S(d) = s) = \begin{cases} p + (1-p)\frac{1}{T} & \text{if } t = \varphi(s) \\ \frac{1-p}{T} & \text{otherwise} \end{cases}$$

It is easy to check that this corresponds to a substitution $p_0 = p + (1-p)\frac{1}{T}$, i.e. $p = \frac{Tp_0-1}{T-1}$. We will refer to probability p as the aspect extraction accuracy.

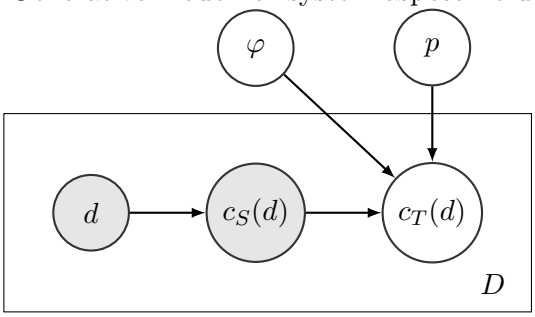
This abstraction thus models some stability in the correspondence between system and true aspects, determined by φ and p. We need not make any particular assumption on φ for our purposes, in fact we may just define it as a random assignment with uniform probability. The mapping is therefore neither injective not surjective (as in general $S \neq T$). Thus φ and p also capture the imprecision in the correspondence between true and extracted aspects: the correspondence would be

perfect if φ was bijective (assuming $S = T$), and $p = 1 (= p_0)$. The aforementioned scenarios where the true aspects are known to the system fit as a particular case of this model, where φ is the identity function, but still $p(\varphi(c_t(d) = a | c_s(d) = a) < 1$ in general. Adding up the previous considerations, we will assume the following generative model

1. A random map $\varphi : A_S \to A_T$ is sampled in the following way: for each system aspect s, the aspect $\varphi(s)$ is drawn from a prior distribution $\tilde{p}(t)$ over A_T. We will usually take $\tilde{p}(t) \sim \text{Unif}_{A_T}(t)$.
2. For each document $d \in D$, a system aspect $s \in A_S$ is sampled from a prior distribution $\tilde{p}(s)$. Again, we usually take $\tilde{p}(s) \sim \text{Unif}_{A_S}(s)$. We take $c_S(d) = s$. Then
   (a) With probability p, we fix the true aspect of d as $c_T(d) = \varphi(c_S(d)) = \varphi(s)$.
   (b) With probability $1 - p$, document d is misclassified by the system, and its true aspect is sampled from prior $\tilde{p}(t)$.

Parameter p models aspect extraction system accuracy, in the terms of one minus the percentage of "missclassified" or ambiguous documents. On the other hand, distribution φ regulates true aspect coverage. We can summarize the previous generative model with the following Bayesian network

Figure 5.1: Generative model for system aspect - true aspect generation



It is also interesting to note that the relation between A_S and A_T can be seen as an issue of clustering agreement degree. The assignment of aspects to documents define a partition of the document set. A common measure of how well two partitions approximate each other is the Rand index (Rand, 1971), which can be defined as the probability that two elements (documents) from the same cluster (having the same aspect) in one partition belong to the same cluster in the other. It can be seen (we omit the details) that the Rand index is directly determined from our model by:

$$RI(c_S, c_T) = p(c_T(d) = c_T(d') | c_S(d) = c_S(d')) = p_0^2 + \frac{1 - p_0^2}{T - 1}$$

5.1.2 Kendall distance prediction

Diversification algorithms take a baseline ranking and reorder some of the retrieved documents in order to maximize the expected user-perceived diversity of the list.

Depending on the information provided by the structure of the aspect space, the diversifier may reorder the entire list or just swap a couple of documents. In the described neutral conditions, the amount of changes in the ranking done by the algorithm is completely determined by the composition of the system aspects for the retrieved documents. In particular, the average amount of change is heavily conditioned by the size of the system aspect space. For example, if a system has $S = 2$, a type 1 diversification algorithm will just move one document, the first one whose aspect is not the same as the one of the first document of the list. The possible increase in the list diversity is then bounded by the amount of new information provided by that single document to the user.

We will analyze the diversity potential of an aspect space by the expected amount of change introduced in the baseline ranking by the diversifier when using the selected aspects. The expected difference between the original and the diversified lists will indicate the amount of information pieces that the diversification system can reorder to enhance the diversity.

This "potential for change' dependency on the aspect space size was first studied by Vargas et al. (2012b). They simulated different neutral baseline rankings (baselines with constant document scores) and a random assignment of aspects. Then, they applied IA-Select and xQuAD to the simulated rankings and used Spearman ρ rank correlation to measure the distance between the diversified ranking and the baseline.

In this section we will explore a similar approach. We will measure the expected "room for diversification" available for a random baseline in terms of the amount of changes introduced by the diversifier in the original ranking once the aspect space is fixed. In contrast to Vargas et al. (2012b), instead of using Spearman rank correlation to measure the difference between both rankings, we will use a more combinatorial index, the normalized Kendall distance.

Given two rankings of the same length N, i, e, two permutations $L_1 = (l_1(i))_{i=1}^N$ and $L_2 = (l_2(i))_{i=1}^N$ of the list $(1, \ldots, N)$. The Kendall distance measures the number of pairs of elements (i, j) that appear in different relative order in both lists, and normalizes it dividing it by the total amount of pairs, $\frac{N(N-1)}{2}$. Explicitly

$$Kendall(L_1, L_2) = \frac{2}{N(N-1)} |\{(i,j) | i < j, (l_1(i) < l_1(j) \wedge l_2(i) > l_2(j)) \vee (l_1(i) > l_1(j) \wedge l_2(i) < l_2(j))\}|$$

From the normalized Kendall distance, we can construct a rank correlation index, named Kendall τ correlation, simply taking

$$\tau(L_1, L_2) = \frac{|\{\text{concordant pairs}\}| - |\{\text{discordant pairs}\}|}{N(N-1)/2} = 1 - 2Kendall(L_1, L_2)$$

As Spearman ρ correlation, Kendall τ measures the correlation in ranking order between both lists. Indeed Kendall (1948) showed that both correlations are particular cases of a general correlation. Let $A = (A(i))_{i=1}^N$ and $B = (B(i))_{i=1}^N$ be two lists. For every pair (i, j), let us consider antisymmetric matrices (a_{ij}) and (b_{ij}) measuring a certain score of the relative order of the pairs $(A(i), A(j))$ and

$(B(i), B(j))$ respectively. Then the general correlation coefficient is defined as

$$\Gamma = \frac{\sum_{i,j=1}^{N} a_{ij} b_{ij}}{\sqrt{\sum_{i,j=1}^{N} a_{ij}^2} \sqrt{\sum_{i,j=1}^{N} b_{ij}^2}}$$

Then taking $a_{ij} = sign(A(j) - A(i))$ and $b_{ij} = sign(B(j) - B(i))$, $a_{ij} b_{ij}$ is 1 if (i, j) is a concordant pair and -1 otherwise. As $a_{ij}^2 = b_{ij}^2 = 1$ the denominator is $N(N-1)$. Taking into account that we are adding each pair (i, j) twice ((i, j) and (j, i)), we recover Kendall τ distance.

On the other hand, taking $a_{ij} = A(j) - A(i)$ and $b_{ij} = B(j) - B(i)$, direct computation shows that we recover Spearman's ρ correlation.

Therefore, in terms of qualitative measure of the amount of change between diversified and baseline rankings, both correlations are similar. In this case, we will choose to analyze the Kendall distance (and thus, the expected Kendall τ) due to its combinatorial nature. The inherent symmetries existing in the rankings will make exact computations more tractable that in the case of the Spearman correlation. We will exploit these symmetries to develop a closed formula for the Kendall distance both for type 1 (equation (5.1.2)) and type 2 (equation (5.1.3)) diversifiers in terms of the number of aspects A and the number of retrieved documents D.

Type 1 diversifier expected Kendall distance

Type 1 diversifier searches for an element of each available aspect until it gets every system aspect covered. Document d_l ends in position t if the following conditions hold

- If d_l is not the first document of class $c(d_l)$, t equals l plus the number of non-covered aspects among the l first documents, because the diversifier has to search for documents covering those aspects behind document d_l and then exactly those documents jump over d_l.
- If d_l is the first of class $c(d_l)$, then t is the number of covered aspects among the first l documents (it is the next new aspect found in the ranking and thus, it is positioned next to the last found one).

As type 1 diversifier preserves the relative order of the documents that are not the first of their classes, in order to compute the number of inversions we just have to consider the first elements of each class and count the number of positions that they jumped forward when selected, i.e., $l - t + 1$. Let us denote by $\pi(d_l)$ the number of documents "jumped" by d_l when applying the diversification algorithm. It becomes clear that

$$\pi(d_l) = \begin{cases} l - t & \text{if } d_l \text{ is the first of its class} \\ 0 & \text{otherwise} \end{cases}$$

Then

$$\mathbb{E}[Kendall] = \frac{2}{D(D-1)} \mathbb{E}\left[\sum_{l=0}^{D-1} \pi(d_l)\right] = \frac{2}{D(D-1)} \sum_{l=0}^{D-1} \mathbb{E}[\pi(d_l)] \qquad (5.1.1)$$

We can decompose this final expectation depending on the value of $\pi(d_l)$.

$$\mathbb{E}[\pi(d_l)] = 0 \cdot p(\text{not first}) + \sum_{t=0}^{l-1}(l-t)p(l-1 \text{ first documents cover } t \text{ aspects} \wedge c(d_l) \text{ covers one of the other } A-t)$$

As system aspects are assumed to be independent, we can decompose the probability as the product of the probability of documents $d_0, \ldots, d_{l-1}$ being assigned to exactly t different aspects and the probability of $c(d_l)$ to be one of the remaining $A-t$. As aspect distribution is uniform, the latter is

$$p(c(d_l)\text{covers one of the other } A-t) = \frac{A-t}{A}$$

The other probability corresponds to the proportion of functions $c : [0, l-1] \to [1, A]$ whose image has size t. There are $\binom{A}{t}$ possible ways of choosing the selected t aspects and for each choice, the probability of the image being exactly those aspects is (see (5.3.2) for details)

$$p(\{c : [0, l-1] \to [1, A] | Im(c) = [1, t]\}) = \sum_{j=0}^{t}(-1)^j \binom{t}{j} \left(\frac{t-j}{A}\right)^{l-1}$$

Therefore, substituting in equation (5.1.1) yields

$$\mathbb{E}[Kendall] = \frac{2}{D(D-1)} \sum_{l=0}^{D-1}\sum_{t=0}^{l}(l-t)\binom{A}{t}\frac{A-t}{A}\sum_{j=0}^{t}(-1)^j \binom{t}{j} \left(\frac{t-j}{A}\right)^{l-1} \quad (5.1.2)$$

Type 2 diversifier expected Kendall distance

If d_l is the n-th document with aspect c, when a type 2 algorithm selects it, it jumps over the elements d_i, $i < l$ which where the $n+1$-th document or more appearing in the ranking with their correspondent aspect. Let $NA(d_i)$ be the order of document d_i among documents d_j such that $c(d_i) = c(d_j)$,

$$NA(d_l) = |\{d_j | j < l, c(d_j) = c(d_l)\}| + 1$$

Then d_l jumps over d_i if and only if $i < l$ and $NA(d_l) < NA(d_i)$. Additionally, let us denote by $D_l(c)$ the number of documents with aspect c among the l first ones,

$$D_l(c) = |\{d_j | j < l, c(d_j) = c\}|$$

In particular, $NA(d_l) = D_l(c(d_l)) + 1$. Then d_l jumps over $\min\{0, D_l(c') - NA(d_l)\}$ documents with aspect c'. Let

$$\mathcal{C}_l = \{c' | D_l(c') \geq NA(d_l)\}$$

Then, by the previous characterization of the documents jumped by d_l, if $c := c(d_l)$ then

$$\pi(d_l) = \sum_{c' \in \mathcal{C}_l}(D_l(c') - NA(d_l)) = \sum_{c' \in \mathcal{C}_l} D_l(c') - |\mathcal{C}_l| NA(d_l) = \sum_{c' \in \mathcal{C}_l} D_l(c') - |\mathcal{C}_l|(D_l(c)+1)$$

Pondering over all possible aspect assignations for document d_l, taking into account that aspects are drawn independently from a uniform distribution, we get

$$\mathbb{E}_c[\pi(d_l)] = \frac{1}{A}\sum_{c\in A}\left(\sum_{c'\in\mathcal{C}_l} D_l(c') - |\mathcal{C}_l|(D_l(c)+1)\right) = \frac{1}{2A}\sum_{c,c'\in A}|D_l(c)-D_l(c')| - \frac{1}{A}\sum_{c\in A}|\mathcal{C}_l|$$

As for every pair $(c,c')\in A^2$, either $c\in\mathcal{C}_{c'}$ or $c'\in\mathcal{C}_c$ or $D_l(c)=D_l(c')$. Denoting $B = \frac{1}{A}\sum_{c\in A}|\mathcal{C}_c|$ and pondering over the aspect distributions of the remaining documents

$$\mathbb{E}[\mathbb{E}_c[\pi(d_l)]] = \frac{1}{2A}\sum_{c,c'\in A}\mathbb{E}[|D_l(c)-D_l(c')|] - \mathbb{E}[B]$$

By sum symmetry, it is enough to check which is the expected order for the difference between the number of documents in one aspect and the other

$$E[|D_l(c)-D_l(c')] = 2\sum_{i=0}^{l/2}\sum_{j=i}^{l-i} p(D_l(c)=i\wedge D_l(c')=j)(j-i)$$

In order to compute the last distribution, we have to count the number of distributions $c:[1,D]\to[1,A]$ for which $D_l(c)=i$ and $D_l(c')=j$. This corresponds to the number of ways of choosing i documents among the l first documents for aspect c, j documents among the $l-i$ remaining ones for aspect c', and choosing one of the other $A-2$ aspects for the remaining $l-i-j$ documents. Then

$$p(D_l(c)=i\wedge D_l(c')=j)(j-i) = \frac{(A-2)^{l-i-j}\binom{l}{i}\binom{l-i}{j}}{A^l}$$

On the other hand, as for every pair $(c,c')\in A^2$, either $c\in\mathcal{C}_{c'}$ or $c'\in\mathcal{C}_c$ or $D_l(c)=D_l(c')$, then

$$B = \frac{1}{A}|\{(c,c')\in A^2\}| - \frac{1}{A}|\{(c,c')\in A^2|D_l(c)=D_l(c')\}| := A - \frac{\lambda}{A}$$

Taking the expected value over all possible aspect distributions yield

$$\mathbb{E}[\lambda] = \frac{A(A-1)}{2}p(D_l(c)=D_l(c')) = \frac{A(A-1)}{2}\sum_{i=0}^{l/2}\frac{\binom{l}{i}\binom{l-i}{i}(A-2)^{l-2i}}{A^l}$$

Substituting all the previous computations in (5.1.1) results in

$$\mathbb{E}[Kendall] = \frac{2(A-1)}{D(D-1)}\sum_{l=0}^{D-1}\sum_{i=0}^{l/2}\left(\sum_{j=i}^{l-i}\frac{(A-2)^{l-i-j}\binom{l}{i}\binom{l-i}{j}}{A^l}(j-i) + \frac{\binom{l}{i}\binom{l-i}{i}(A-2)^{l-2i}}{2A^l}\right) - \frac{D}{2}$$

We can further simplify the last equation by some combinatorial manipulation. Let

$$\alpha = \sum_{i=0}^{l/2} \sum_{j=i}^{l-i} \frac{(A-2)^{l-i-j} \binom{l}{i} \binom{l-i}{j}}{A^l} (j-i)$$

We make the following change of variables

$$x \leftarrow i+1$$

$$y \leftarrow i$$

Then

$$\alpha = \sum_{x=0}^{l} \frac{(A-2)^{l-x}}{A^l} \sum_{y=0}^{\lfloor x/2 \rfloor} (x-2y) \binom{l}{y, x-y, l-x}$$

We will give an explicit form for the inner sum, so that the overall complexity of the global expression is reduced by one order. We have

$$\sum_{y=0}^{\lfloor x/2 \rfloor} (x-2y) \binom{l}{y, x-y, l-x} = \sum_{y=0}^{\lfloor x/2 \rfloor} (x-2y) \frac{l!}{y!(x-y)!(l-x)!} = \binom{l}{x} \sum_{y=0}^{\lfloor x/2 \rfloor} (x-2y) \binom{x}{y}$$

We decompose the last summation in two similar tasks. The first is computing

$$\sum_{y=0}^{\lfloor x/2 \rfloor} x \binom{x}{y} = x \sum_{y=0}^{\lfloor x/2 \rfloor} \binom{x}{y}$$

And the second is evaluating

$$\sum_{y=0}^{\lfloor x/2 \rfloor} y \binom{x}{y} = \sum_{y=1}^{\lfloor x/2 \rfloor} \frac{x(x-1)!}{(y-1)!(x-y)!} = x \sum_{y=1}^{\lfloor x/2 \rfloor} \frac{(x-1)!}{(y-1)!((x-1)-(y-1))!} =$$

$$x \sum_{y=0}^{\lfloor x/2 \rfloor} \binom{x-1}{y-1} = x \sum_{y=0}^{\lfloor x/2 \rfloor - 1} \binom{x-1}{y}$$

From $\sum_{y=0}^{x} \binom{x}{y} = (1+1)^x = 2^x$ and the symmetry of the binomial, we get that the desired sums essentially correspond to summing half of a line of the Pascal Triangle. Therefore

$$\sum_{y=0}^{\lfloor x/2 \rfloor} \binom{x}{y} = \begin{cases} 2^{x-1} & x \text{odd} \\ 2^{x-1} + \frac{1}{2} \binom{x}{x/2} & x \text{even} \end{cases}$$

Similarly

$$\sum_{y=0}^{\lfloor x/2 \rfloor - 1} \binom{x-1}{y} = \begin{cases} 2^{x-2} & x \text{ even} \\ 2^{x-2} - \frac{1}{2} \binom{x-1}{(x-1)/2} & x \text{ odd} \end{cases}$$

Adding up both equations, we get that

$$\sum_{y=0}^{\lfloor x/2 \rfloor}(x-2y)\binom{x}{y} = x\sum_{y=0}^{\lfloor x/2 \rfloor}\binom{x}{y} - 2x\sum_{y=0}^{\lfloor x/2 \rfloor -1}\binom{x-1}{y} =$$

$$= \left\{ \begin{array}{cc} x\left(2^{x-1}+\frac{1}{2}\binom{x}{x/2}\right)-x2^{x-1} & x \text{ even} \\ x2^{x-1}-x\left(2^{x-1}-\binom{x-1}{(x-1)/2}\right) & x \text{ odd} \end{array} \right\} = \left\{ \begin{array}{cc} \frac{x}{2}\binom{x}{x/2} & x \text{ even} \\ x\binom{x-1}{(x-1)/2} & x \text{ odd} \end{array} \right\}$$

Direct computation of the previous formula yield

$$\sum_{y=0}^{\lfloor x/2 \rfloor}(x-2y)\binom{x}{y} = x\binom{x-1}{\lfloor \frac{x-1}{2} \rfloor}$$

Therefore, substituting the value of α in the Kendall equation, we get

$$\mathbb{E}[Kendall] = \frac{2(A-1)}{D(D-1)}\sum_{l=0}^{D-1}\left(\sum_{x=1}^{l}\frac{l(A-2)^{l-x}}{A^l}\binom{l-1}{x-1}\binom{x-1}{\lfloor \frac{x-1}{2} \rfloor}+ \right.$$

$$\left. \sum_{i=0}^{l/2}\frac{\binom{l}{i}\binom{l-i}{i}(A-2)^{l-2i}}{2A^l}\right) - \frac{D}{2} \quad (5.1.3)$$

5.1.3 Subtopic Recall prediction model

Subtopic recall corresponds to a useful basic metric to evaluate the overall diversity of a system in terms of the subtopic coverage of each query. Using the cut version, $S-Recall@\alpha$ allows us to measure continuously the aspect coverage of the system when advancing through the ranking.

Under the hypothesis of documents belonging to a single class, in an ideal system, where the aspect extraction space is able to retrieved the exact intent space over which the system will be evaluated, the subtopic recall would be 1 at any cut. This is due to diversification algorithms placing documents with non-redundant aspects in the first positions of the ranking, thus covering the maximum number of true aspects possible.

Nevertheless, as reflected by our proposed generative model, a real aspect extraction system won't be able to retrieve the real aspect space perfectly. Then, the diversification algorithm placing different system aspects in the first positions won't imply that it will recover different real aspects. The real aspect coverage depends on the interaction between the space of system aspects A_S and the space of true aspects A_T. We will focus on the dependency of this interaction and the metric value on the system aspect space size, S.

In this section we will build a predictive model for the Subtopic Recall metric in terms of the metric cutoff α, the number of system aspects S, the number of true aspects T and aspect extraction accuracy p. Assuming the previously described generative model, we will develop exact analytic equations for the expected value of $S-recall@\alpha$ for the baseline ranking (5.1.4) and a type 1 diversifier (5.1.7), and we will approximate the expected subtopic recall for a type 2 diversifier under an additional condition on the accuracy of the aspect extraction system (5.1.8).

Baseline expected subtopic recall

Let α be our fixed cutoff for the metric. As in the experimental settings we will have a narrow number of topics, we will assume that $\alpha \geq T$, so that $S - Recall@\alpha$ corresponds to the number of retrieved aspects among the first α documents over the total number of true aspects T. Let us denote by $p(k)$ the probability of retrieving exactly k different aspects among the first α documents.

Let τ be the number of system failures, i.e., the number of documents whose system aspect has been incorrectly assigned. The probability of exactly τ failures happening comes from a binomial distribution

$$p(\tau \text{ failures}) = \binom{\alpha}{\tau}(1-p)^{\tau}p^{\alpha-\tau}$$

Given τ failures, let λ be the number of system aspects covered by the $\alpha - \tau$ remaining documents. This corresponds to $\binom{S}{\lambda}$ times the number of aspect functions $c : [1, \alpha - \tau] \to [1, S]$ whose image is exactly $[1, \lambda]$, so, by (5.3.2)

$$p(\lambda \text{ system aspects covered}|\tau) = \binom{S}{\lambda}\sum_{j=0}^{\lambda}(-1)^j\binom{\lambda}{j}\left(\frac{\lambda-j}{S}\right)^{\alpha-\tau}$$

Finally, both the λ retrieved system aspects and the τ failures are given random true aspects. The probability of those true aspects covering exactly k aspects corresponds to the ways of choosing those k aspects among the T possible true ones times the probability of assigning the $\lambda + \tau$ "free" aspects to those aspects covering all of them. This corresponds to the proportion of functions $\varphi : [1, \lambda + \tau] \to [1, T]$ whose image is $[1, k]$. Then, by (5.3.2)

$$p(k|\lambda, \tau) = \binom{T}{k}\sum_{j=0}^{k}(-1)^j\binom{k}{j}\left(\frac{k-j}{T}\right)^{\alpha+\tau}$$

Therefore, we obtain

$$p(k) = \sum_{\tau=0}^{\alpha}p(\tau)\sum_{\lambda=0}^{\min S,\alpha-\tau}p(\lambda|\tau)p(k|\lambda,\tau) = \sum_{\tau=0}^{\alpha}\binom{\alpha}{\tau}(1-p)^{\tau}p^{\alpha-\tau}\cdot$$
$$\cdot\sum_{\lambda=0}^{\min S,\alpha-\tau}\binom{S}{\lambda}\left(\sum_{j=0}^{\lambda}(-1)^j\binom{\lambda}{j}\left(\frac{\lambda-j}{S}\right)^{\alpha-\tau}\right)\binom{T}{k}\sum_{j=0}^{k}(-1)^j\binom{k}{j}\left(\frac{k-j}{T}\right)^{\alpha+\tau} \tag{5.1.4}$$

From the previous equation, the expected random subtopic recall at α can be found as

$$\mathbb{E}[S - Recall@\alpha] = \sum_{k=0}^{T}\frac{k}{T}p(k)$$

Type 1 diversifier expected subtopic recall

As we will execute our simulations taking the number of system aspects from 1 or 2 aspects to more than D, we can't suppose that $S \leq \alpha$, as we did with the true aspects. Instead, we must distinguish between two scenarios: the cutoff α being greater than S or not.

If $\alpha \leq S$, the diversifier positions a different system aspect in each of the first α positions. Therefore, either if the system has failed to classify each document or not, the choices of the true aspect for each of the α documents are independent. Then, the probability $p(k)$ corresponds to the proportion of functions $\varphi : [1, \alpha] \to [1, T]$ that cover exactly k aspects. Analogous computations to the previous scenario lead to

$$p(k) = \binom{T}{k} \sum_{j=0}^{k} (-1)^j \binom{k}{j} \left(\frac{k-j}{T}\right)^{\alpha}$$

If $\alpha > S$, we must distinguish between possible errors among the first S documents, whose system aspects are known, and errors among the other documents. Let τ_1 be the number of fails among the first S documents, and τ_2 the number of fails among the rest. The probabilities of finding exactly those errors are independent, and each of them is a binomial

$$p(\tau_1) = \binom{S}{\tau_1} (1-p)^{\tau_1} p^{S-\tau_1}$$

$$p(\tau_2) = \binom{\alpha - S}{\tau_2} (1-p)^{\tau_2} p^{\alpha-S-\tau_2}$$

The first $S - \tau_1$ hits correspond to $S - \tau_1$ different aspects. The other $\alpha - S - \tau_2$ hits can have any system aspect. Let λ be the number of system aspects covered among all hits. As the first documents cover at least $S - \tau_1$ different aspects, but there are only $\alpha - \tau_1 - \tau_2$ hits, we have $S - \tau_1 \leq \lambda \leq \min\{S, \alpha - \tau_1 - \tau_2\}$ and the total system coverage depends on the number of aspects covered by the last $\alpha - S - \tau_2$ hits. Exactly λ system aspects are covered if the last $\alpha - S - \tau_2$ hits cover at least the $\lambda - S + \tau_1$ remaining aspects, possibly covering also some of the already existing $S - \tau_1$. In particular, as we can choose arbitrary which $\lambda - S + \tau_1$ aspects are covered last, applying (5.3.1) we get

$$\begin{aligned} p(\lambda|\tau_1, \tau_2) &= \binom{\tau_1}{\lambda - S + \tau_1} \frac{|\{\varphi : [1, \alpha - S - \tau_2] \to [1, \lambda] | Im(\varphi) \supseteq [1, \lambda - S + \tau_1]\}|}{|\{\varphi : [1, \alpha - S - \tau_2] \to [1, S]\}|} \\ &= \binom{\tau_1}{\lambda - S + \tau_1} \sum_{j=0}^{\lambda - S + \tau_1} (-1)^j \binom{\lambda - S + \tau_1}{j} \left(\frac{\lambda - j}{S}\right)^{\alpha - S - \tau_2} \end{aligned} \quad (5.1.5)$$

Finally, true aspects are sampled independently from the λ covered system aspects and the $\tau_1 + \tau_2$ fails.

$$\begin{aligned} p(k|\lambda, \tau_1, \tau_2) = \binom{T}{k} \frac{|\{\varphi : [1, \lambda + \tau_1 + \tau_2] \to [1, k] | Im(\varphi) = [1, k]\}|}{|\{\varphi : [\lambda + \tau_1 + \tau_2] \to [1, T]\}|} = \\ \binom{T}{k} \sum_{j=0}^{k} (-1)^j \binom{k}{j} \left(\frac{k-j}{T}\right)^{\lambda + \tau_1 + \tau_2} \end{aligned} \quad (5.1.6)$$

As τ_1 and τ_2 are independent, we finally get

$$p(k) = \sum_{\tau_1=0}^{S} p(\tau_1) \sum_{\tau_2=0}^{\alpha-S} p(\tau_2) \sum_{\lambda=S-\tau_1}^{\min\{S,\alpha-\tau_1-\tau_2\}} p(\lambda|\tau_1,\tau_2)p(k|\lambda,\tau_1,\tau_2) \tag{5.1.7}$$

The explicit final formula for $p(k)$ is get substituting the binomial distributions for τ_1, τ_2 and equations (5.1.5) and (5.1.6) in (5.1.7). As before, the expected value of the metric is given by

$$\mathbb{E}[S - Recall@\alpha] = \sum_{k=0}^{T} \frac{k}{T} p(k)$$

Type 2 diversifier expected subtopic recall

The previous combinatorial strategies become intractable for the Type 2 diversifier. If we compare the combinatorics to the ones of Type 1, we see that the total system aspect coverage depends on the number of errors appearing at the documents for each aspect $s \in A_S$. Instead of counting errors by "diversified rows", we need to count for errors in "columns".

A parameterization over the number of errors τ_s over the documents of aspect s would be possible, but the number of independent parameters would grow as S, leading to an equation with exponential complexity on S. Instead, a model-driven approximation is possible. We will assume that the algorithm of type 2 places in the top S documents the ones for which the corresponding aspects are the most probable. Therefore, the documents in positions $[1, S]$ are almost surely correctly classified. In this approximation we will assume that they are always correctly classified, so that their true aspect is the image of their system aspects by a random $\varphi : A_S \to A_T$.

If $\alpha \leq S$, the behavior of a Type 2 diversifier is clearly the same of a Type 1 diversifier, as we don't reach to check documents after the first S retrieved aspects (the moment when Type 1 stops diversifying). We will focus on $\alpha > S$. Again, we will consider the number of errors. The infallibility hypothesis for the classification of the first S documents makes us suppose that there exist at most $\tau \leq \alpha - S$ classification error. As before, the probability of that exact number of errors happening is a binomial distribution. Then, the probability of the S system aspects and the τ error covering exactly k aspects is simply given by

$$(k|\tau) = \binom{T}{k} \sum_{j=0}^{k} (-1)^j \binom{k}{j} \left(\frac{k-j}{T}\right)^{S+\tau}$$

Putting all together we obtain

$$p(k) = \sum_{\tau=0}^{\alpha-S} \binom{\alpha-S}{\tau} (1-p)^\tau p^{\alpha-S-\tau} \binom{T}{k} \sum_{j=0}^{k} (-1)^j \binom{k}{j} \left(\frac{k-j}{T}\right)^{S+\tau} \tag{5.1.8}$$

5.1.4 Random diversifier ERR-IA prediction model

We recall, that the usual ERR-IA equation (2.1.1) corresponds to

$$ERR-IA = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{D}\frac{1}{i}\prod_{j=1}^{i-1}(1-p(c(d_{j-1})=t))p(c(d_{i-1})=t)$$

by binarizing relevance distribution ($p(c(d_i)=t)=1$ if t is the single d_i true aspect), the previous equation simplifies into

$$ERR-IA = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{\min\{i|c(d_{i-1})=t\}}$$

Even with the assumed simplifications, exact computation of this metric is almost intractable due to the extensive combinatorics of the problem. In contrast with the previous estimations, in which we could exploit the space of symmetries of the problem to factorize the computation into tractable cases, the space of possible ERR-IA values is much greater than that of Kendall distance or S-recall. As we will see, the expression of the ERR-IA value as a sum of inverses of positions induces a family of asymmetric states that depends on the factorial of the number of aspects, thus leading to non-polynomial computational costs.

Nevertheless, we present a semi-closed formula that can be used to estimate the metric value for small aspect spaces and a random diversifier. This will provide information about the inherent expected diversity perceived by the user if the subqueries were generated under the simplified model. Moreover, it can serve as a lower bound for the expected ERR-IA for diversification algorithms of type 1 and 2.

Let i_k, for $k=1,\ldots,T$ denote the position (starting by 1) of the first document belonging to the k-th aspect in the ranking. The value of ERR-IA is given by

$$ERR-IA = \frac{1}{T}\sum_{k=1}^{T}\frac{1}{i_k}$$

Let $p(i_1,\cdots,i_T)$, for $1=i_1<i_2<\cdots i_T\leq D$ be the probability of getting the first of the k-th aspect appearing in the i_k-th document for $k=1,\ldots,T$. By symmetry, we can reduce the computation to the case when aspects appear in order. then

$$\mathbb{E}[ERR-IA] = \frac{1}{T}\sum_{1=i_1<\cdots<i_T\leq D}\left(\frac{1}{i_1}+\ldots+\frac{1}{i_T}\right)p(i_1,\ldots,i_T)T!$$

We need to compute $p(i_1,\ldots,i_T)$. As we said, we can suppose that documents are ordered. As the diversifier is random, the aspect assignment is uniform among all possible aspect configurations. Indexes $i_1,\ldots,i_T$ occur if and only if for each j within $i_k\leq j<i_{k+1}$, the aspect of document d_{j-1} lies within aspects $[1,k]$, and $c(d_{i_k-1})=k$. Then we can split the suitable possible aspect functions $c:[1,D]\to[1,T]$ in those mapping each interval $[i_k+1,i_{k+1}-1]$ to $[1,k]$ and mapping i_k to k.

Then

$$p(i_1,\ldots,i_T) = \frac{\prod_{j=3}^{T}|\{c:[i_{j-1}+1,i_j-1]\to[1,1]\}||\{c:[i_{T+1},D]\to[1,T]\}|}{|\{c:[1,D]\to[1,T]\}|} =$$

$$\frac{1}{T^D}\prod_{j=3}^{T}(j-1)^{i_j-i_{j-1}-1}T^{D-i_T} = \frac{1}{(T-1)!}\prod_{j=2}^{T}\left(\frac{j-1}{j}\right)^{i_j} = \frac{1}{(T-1)!}\prod_{j=2}^{T}\left(1-\frac{1}{j}\right)^{i_j}$$

Therefore, we get that

$$\mathbb{E}[ERR-IA] = \sum_{1=i_1<i_2<\cdots<i_T}\left(\frac{1}{i_1}+\ldots+\frac{1}{i_T}\right)\prod_{j=2}^{T}\left(1-\frac{1}{j}\right)^{i_j} \tag{5.1.9}$$

5.1.5 System subtopic recall adjustment

Let us suppose that we have a quality function $f:\mathbb{N}\to\mathbb{R}$ such that for any number of system aspects S it determines the expected value of a certain metric (Kendall distance, S-Recall@k, ERR-IA, etc.) supposing ideal conditions on the system aspect coverage, i.e., $S-recall=1$ (every system aspect can be found among documents in the top N considered ranking).

Nevertheless, this ideal situation won't be possible neither in simulated experiments nor real data. In general, the pool of generated system aspects will be bigger than the set of observed aspects in the dataset. The proportion of the total aspect space that can be fund in the sample is determined by the global subtopic recall of the corpus with respect to the system aspect space.

Our objective is to obtain an estimating function $\tilde{f}:\mathbb{N}\to\mathbb{R}$ from quality function f that measures the expected value of the given metric without assuming the total coverage of the true aspect space. In particular, as expectations commute, from our simplified generative model we can assume that true aspects of documents are sampled from a uniform distribution over the complete set of true aspects. Let S be the number of total system aspects, and let us denote by $p(k)$ for $k=1,\ldots,S$, the probability of the top D documents covering exactly k of those aspects. Then the expected true value of the metric f is given by

$$\tilde{f}(S) = \mathbb{E}_{p(k)}[f(k)] = \sum_{k=1}^{S} f(k)p(k) \tag{5.1.10}$$

As every element are assumed to have random aspects, the distribution of possible aspect assignment functions $c:[1,D]\to[1,S]$ is uniform. On the other side, $p(k)$ can be decomposed by the sum of probabilities of retrieving each subset of k aspects from $[1,S]$. Reordering the retrieved aspects, $p(k)$ can be written as

$$p(k) = \binom{S}{k}p(\text{cover exactly the } k \text{ first aspects})$$

The later is computed in the last section of this chapter (5.3.2). Substituting into the real quality equation yields

$$\tilde{f}(S) = \sum_{k=1}^{S} f(k)\binom{S}{k}\sum_{j=0}^{k}(-1)^j\binom{k}{j}\left(\frac{k-j}{S}\right)^D \tag{5.1.11}$$

This equation holds for any quality measure under the given generative hypothesis. For example, Kendall prediction measures are obtained under the assumption of the system having exactly S aspects. However, in our generative model, document aspects from a certain ranking can have less than S aspects, as it is possible that a system aspect $s \in S$ is never assigned to a single document. Then the diversity algorithms effectively acts as if the system had just the number of aspects covered by at least one document of the baseline ranking, which may be strictly less than S. In other words, after applying the aspect extraction algorithm, the baseline ranking may present a subtopic recall over the system aspects below 1.

By applying equation (5.1.11) to Kendall distance equations (5.1.2) and (5.1.3) we account for this effect which, as we will see, appears in real data naturally. In the more general contest, an aspect extracted by the system may be only assigned as a secondary aspect (with low probability) to every document, or be assigned only to irrelevant documents, therefore becoming essentially invisible to the diversification algorithm.

5.1.6 Probability of a diversifier being better than random

Besides the expected value of a certain diversity metric, another interesting measure of the quality of a diversification algorithm is the probability of being better than the baseline or than a random diversifier. All previous arguments exploiting the symmetries of the aspect selection functions when we range over all possible transition functions $\varphi : A_S \to A_T$ allow us to identify both scenarios.

In our model, take the expected values both over the system aspect choices and the functions $\varphi : A_S \to A_T$. Both choices are independent, so the expected values with respect to each of the choices commute. Therefore, considering a random baseline and then finding the expected value of the metric over all possible random diversifiers is equivalent to assuming a random diversifier and getting the expected value of the metrics with respect to each possible baseline.

The probability of a system being better than random, measured as the probability of the system obtaining a better result in a certain metric, provides us quantitative information about how better the proposed algorithm is, but also gives us information about the statistical significance of this quantitative data.

We will assume that the probability is computed through a decoupled model / experiment, in which both systems are given random independent samples. Each system gets its own different data sample, and is executed independently. Then, metric measures for both systems are taken and compared to each other. While this is slightly different from the usual paired comparison, it corresponds to an estimation of the probability of the diversifier getting an average better result than a random diversifier if the number of samples is high enough.

The previously described models are developed combinatorially and provide, for the considered scenarios, a full state distribution for each of the metrics. In all the considered cases, there existed a finite set of possible states for which the metric was computed. Let $\mathcal{R}$ be the set of possible states of the metric. $\mathcal{R}$ is equipped with the total order $\leq$ inherited from the metric's order. We say that $r \leq s$, for $r, s \in \mathcal{R}$ if a system getting to state s is considered better than one reaching state r from the point of view of the metric. This does not mean necessarily that the numerical

value (if it exist) of the metric for state r is less than the value for state s. The "sign" choice is selected from the "goodness" of the system, not the objective of the function (for example, if we consider an "aspect redundancy metric", states with higher redundancy will be considered as smaller in the order of $\mathcal{R}$.

Then, the probability of the diversifier being better than random corresponds to the probability of the system reaching higher states than the random diversifier. Let $p_r(k)$ and $p_s(k)$ denote the probability of reaching state k in the random system an the diversified system respectively. Then, as p_r and p_s are independent, it yields

$$p(system \geq random) = \sum_{r \in R} p_r(r) \sum_{s \geq r} p_s(s) \tag{5.1.12}$$

In particular, taking $\mathcal{R} = \{1, \ldots, T\}$ and substituting equations (5.1.4) and (5.1.7), we can apply the previous equation to obtain the probability of a Type 1 diversifier of being better than a random diversifier in terms of the expected subtopic recall at a certain cut α.

5.2 Experimental results

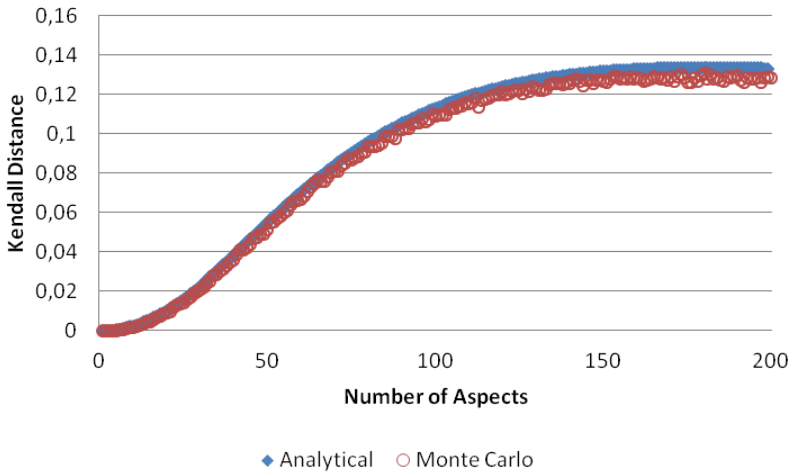
We run two kind of tests. First of all, a set of Monte-Carlo experiments simulating the proposed simplified generative model for both system and true aspects are executed. IA-select and xQuAD diversifiers under the described hypothesis of neutral rank and early stop over each simulated aspect distribution. As a soundness test, all the previously deduced analytical formulas for the expected quality metrics are tested against the Monte-Carlo experiments. We will prove that the obtained expressions are correct and predict with absolute precision the dependency of the metrics to the number of selected aspects.

On the other hand, experiments over the TREC WebTrack data are used to test the generative mode and the system prediction for a real scenario. RapLSA is used as an aspect extraction method in order to provide a homogeneous continuous variation of the aspect space size parameter. IA-select and xQuAD are applied and the studied diversity metrics are calculated. Analytically predicted expected value of the metrics for each number of aspects is compared with the real one, and the resulting data is explained in terms of the qualitative properties of the aspect extraction and the diversification algorithms.

5.2.1 Monte-Carlo experiments

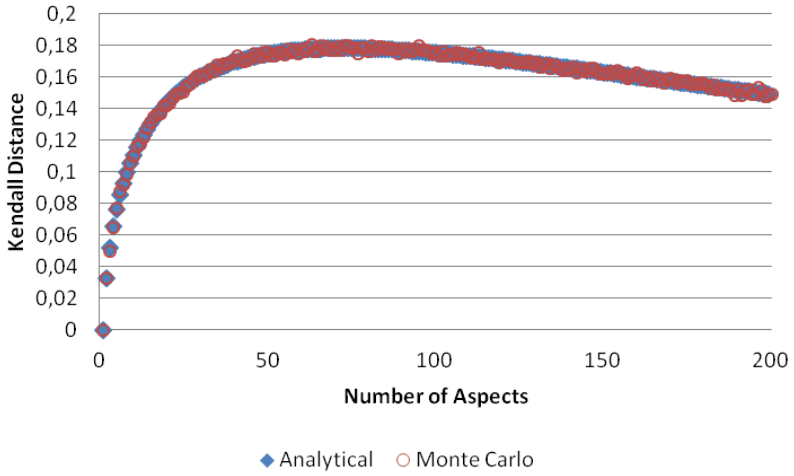
We will start by comparing the analytical expression for the evolution of the expected Kendall distance of type 1 algorithms to the empirical mean Kendall distance obtained by simulating 50 random baseline rankings per data point and applying IA-Select to each of them. We make the total number of system aspects vary from 1 to 200 aspects, therefore getting a total collection of 10.000 samples.

Figure 5.2: Comparison between type 1 analytical result and simulated IA-Select



we observe that the results correlate perfectly, with a Pearson coefficient of 0.9997. Similarly we compare the expected Kendall distance for a type 2 algorithm with the mean observed value of the distance for xQuAD executed over each sample.

Figure 5.3: Comparison between type 2 analytical result and simulated xQuAD

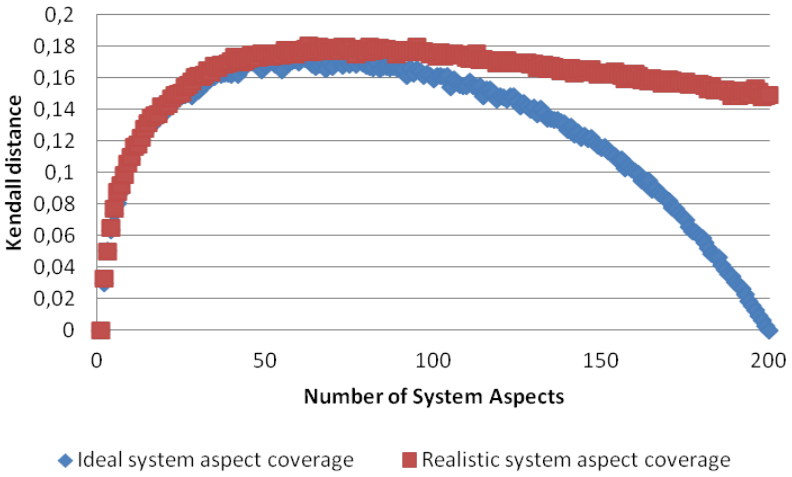


Once again, we observe a perfect correlation, with Pearson coefficient of 0.9987, proving that both analytic equations capture perfectly the diversifier dynamics within the proposed generative model.

In order to appreciate the effect of the system subtopic recall adjustment in the expected value of a metric, we show the difference between a simulation of xQuAD where the system aspects distribution for documents has been forced to

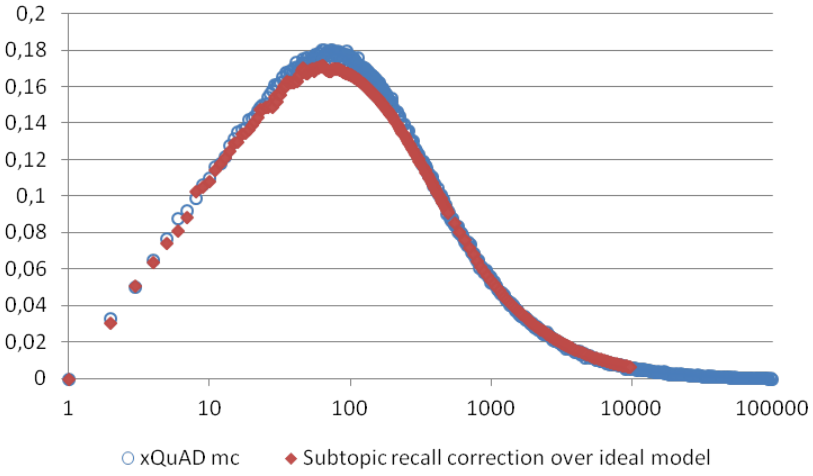
attain perfect subtopic recall (ideal situation) and the general free scenario, where not every system aspect has to appear in the corpus.

Figure 5.4: Comparison between ideal and non-ideal system aspect coverage in Monte Carlo simulation



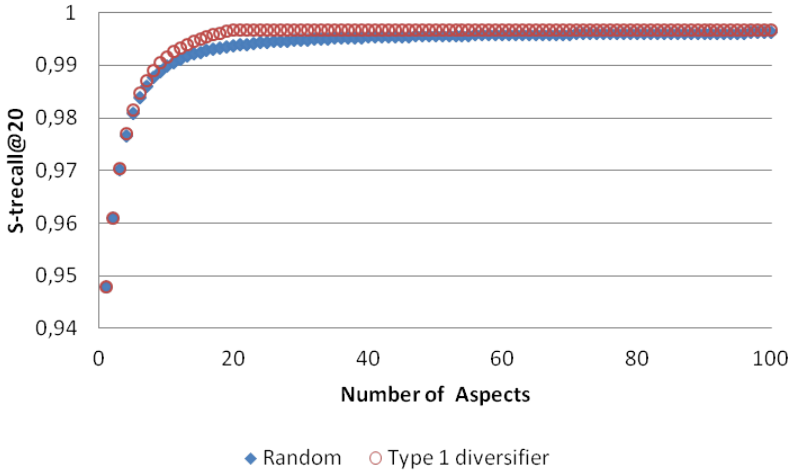
We observe that the curve for the non-ideal situation essentially corresponds to stretching the ideal one to the left. This effect becomes completely clear from equation (5.1.10). The corrected value of the non-ideal metric $\tilde{f}$ over a number of aspects S corresponds to a pondered average of the values of the ideal metric f over k for $1 \leq k \leq S$. Therefore, the new graphic can be understood as a kind of "moving average" of the previous values of the ideal metric, thus stretching it to the right. In order to prove the effectiveness of the correction formula, we take the mean values of the Kendall distance computed by Monte Carlo assuming an ideal system aspect recall and apply the correction formula (5.1.11) to estimate the true nonideal distance.

Figure 5.5: Extended comparison between ideal and non-ideal system aspect coverage in Monte Carlo simulation



We observe that the corrected function fits perfectly the simulated non-ideal data, up to a Pearson coefficient of 0.9966. Here we present a comparison between random and type 1 algorithm S-recall@20 evolution for a set of 4 true aspects.

Figure 5.6: Comparison between analytical expected S-Recall@10 for a random and a type 1 diversifiers

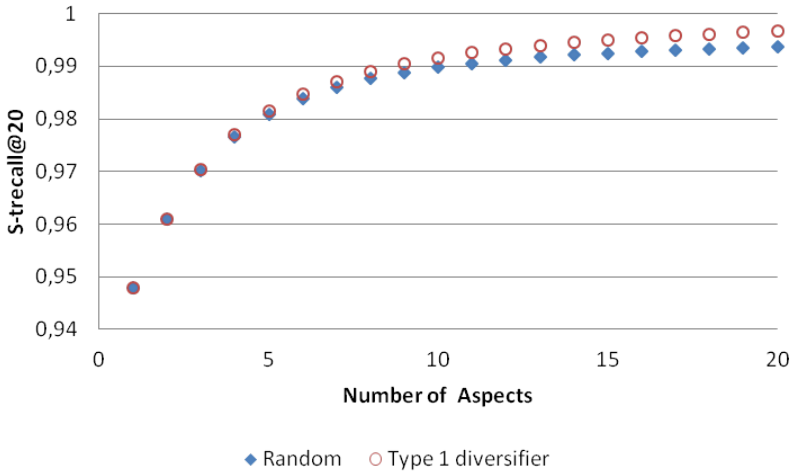


Under the ideal generative model, both systems are expected to converge to a high probability of covering all 4 aspects within the top 20 documents. This is to be expected due to the low number of system aspects in comparison with the high number of documents. The initial slope is essentially due to system aspects creating

a bottleneck for the generation of new aspects once the system aspects have been fixed and covered. After that point coverage of new different true aspects rely on the number of errors existing in the remaining documents.

If we amplify the initial slope we observe that during this process, type 1 diversifier are still better that random, as expected.

Figure 5.7: Comparison detail between analytical expected S-Recall@10 for a random and a type 1 diversifiers



This anomalous convergence phenomenon shows the theoretical prediction limits of the simplified generative model. In a real system, aspect extraction algorithms compress information which is supposed to exist within an inherent space of true aspects. Therefore, system aspect coverage and true aspect coverage depend on each other. In the simplified model, this dependency is just one-directional, as aspect effective distribution depends generatively on system distribution.

5.2.2 Real data resutls

In order to compare the theoretical results with the actual effects on real data, we run an experiment on the TREC 2009/10 diversity task. We take the Indri search engine as a baseline ranking, and we use latent factors extracted by pLSA as the system aspect space (similarly to He et al. (2011) and He et al. (2012)), in such a way that we have fine control over the number of system aspects. In order to avoid interactions between the selection of the aspect space and the tempering, pLSA results are taken untempered and unfiltered.

We run IA-Select, xQuAD and a random diversifier on the top 100 documents re-turned by Indri for each query, with the number of latent factors ranging from 1 to 100. 10 different independent executions of pLSA are run for each choice of the aspect space size, in order to randomize the starting point of the algorithm and improve statistical significance of the data. The analyzed metrics are evaluated

over each resultset. We compute Kendall distance to baseline, subtopic recall at cut 10 and 20 and ERR-IA@20 (as a fair fast approximation of ERR-IA). In order to be closer to the neutral-baseline consideration, xQuAD results are taken with $\lambda = 1$. While the overall diversity results of the diversification system with $\lambda = 1$ are worse than the results with other intermediate λ values, we choose to sow the "pure diversity" $\lambda = 1$ results to analyze the differences between the real imperfect xQuAD execution and our ideal model. The following figures show the obtained results.

Kendall distance

Figure 5.8: Kendall distance variation for IA-Select

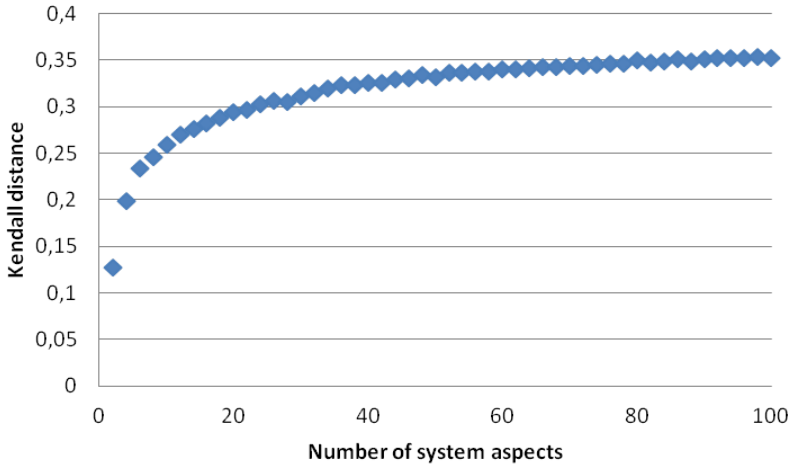
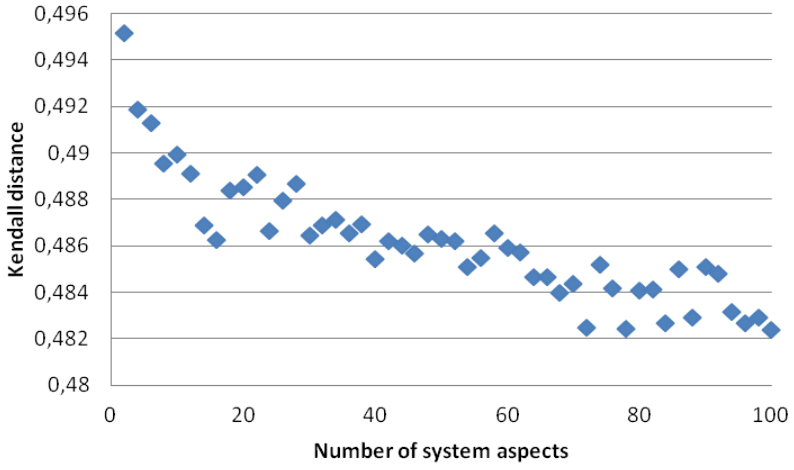


Figure 5.9: Kendall distance variation for IA-Select



We observe that IA-Select have a qualitative behavior comparable to the ones described in both type 1 and 2 algorithms, but xQUAD presents a completely different evolution shape. Moreover, xQuAD presents a much higher than expected number of lists saps. A Kendall distance of almost 0.5 is barely the same as the expected distance of a random re-rank. If we take into consideration the low diversity metric values, we conclude that, for this data, xQuAD diversification results for $\lambda = 1$ are essentially defective.

Regarding to IA-Select, we observe that it reflects exactly twice the expected amount of swaps of a type 2 algorithm. While IA select is classified as a type 1 algorithm, if none of the documents cover an aspect completely, it will continue to diversify expecting to attain a complete coverage, therefore transforming into a type 2. The 2 factor may correspond to swaps between relevant and irrelevant documents being intercalated inside the ranking.

Figure 5.10: Kendall distance variation for IA-Select

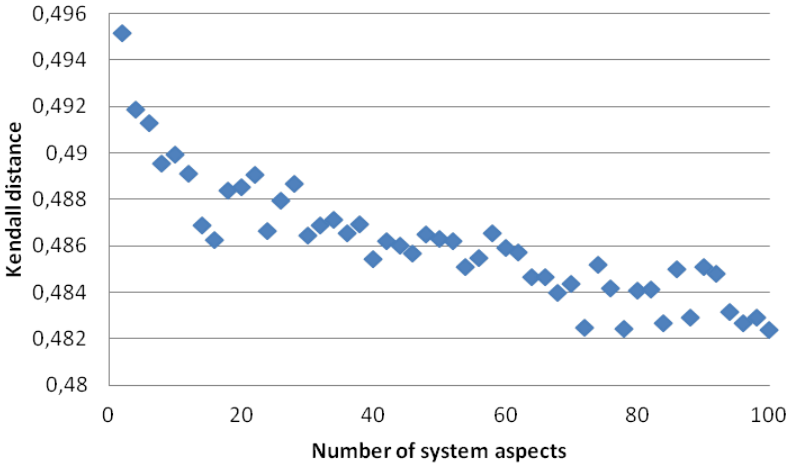


Figure 5.11: S-Recall@10 variation for IA-Select

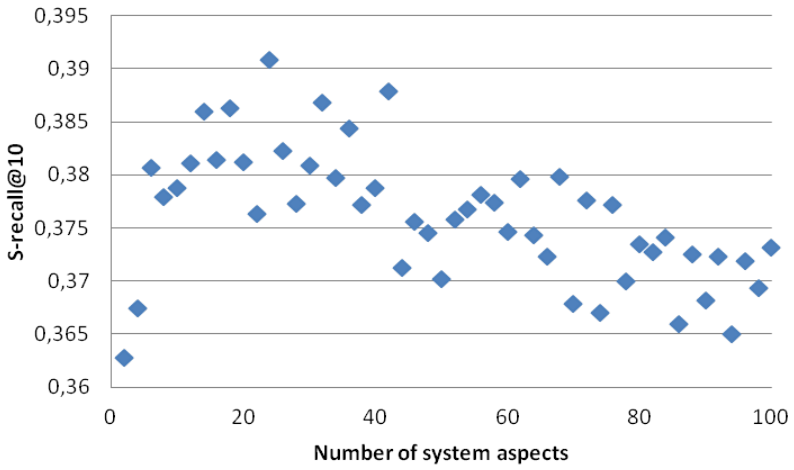
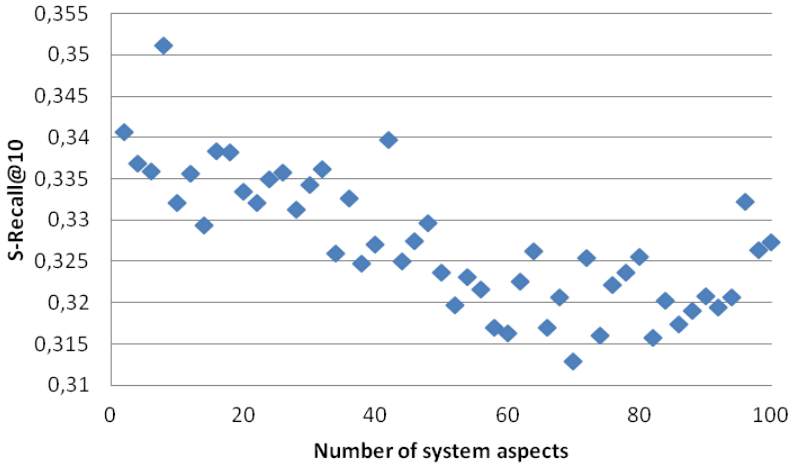


Figure 5.12: S-Recall@10 variation for xQuAD



We observe that the optimal number of aspects for aspects for maximum subtopic recall at 10 is obtained between 20 and 30 aspects. As we will see, this agrees with the results obtained for ERR-IA@20. Of course, in contrast to the theoretical model, perfect subtopic recall is not attained and arbitrary increase of the number of system aspects no longer perturbs true topic distributions. Low number of aspects seem to increase the expected subtopic recall of xQuAD.

Figure 5.13: S-Recall@20 variation for IA-Select

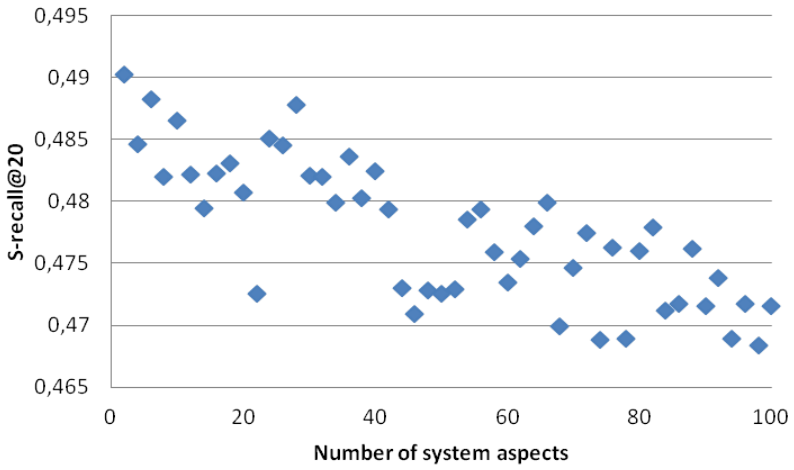
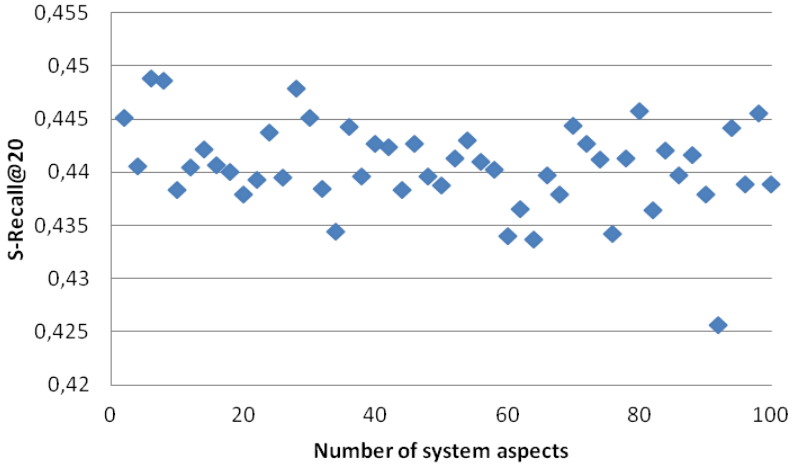


Figure 5.14: S-Recall@20 variation for xQuAD



Another experiment with doubled cut-off shows that the observed decreasing effect on xQuAD also appears in IA-Select. These kind of evolution curves have also been observed in Monte Carlo experiments when simulating document relevance and imposing total coverage conditions over system aspects. Complete analogous computations that the ones pondering the expected coverage by the expected number of errors (5.1.4) allow us to introduce relevance within the previous equations. Doing so has lead us to approximate qualitatively the shape and optimal value prediction for the previous curves, but complete fit has not been achieved.

Figure 5.15: ERR-IA@20 variation for IA-Select

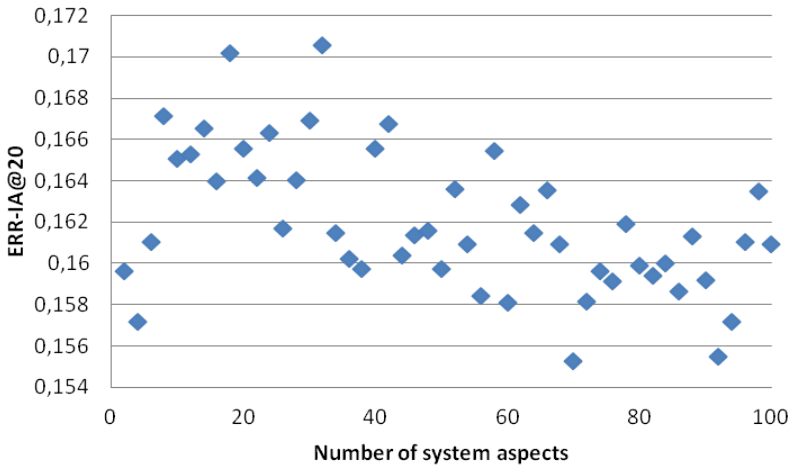
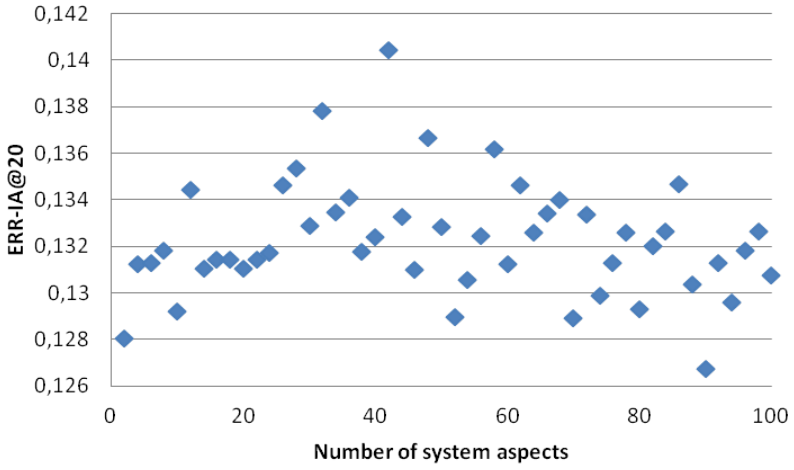


Figure 5.16: ERR-IA@20 variation for xQuAD



We appreciate that the optimum size of the aspect space with respect to the S-recall@20 and the ERR-IA@20 metrics correspond to a approximate 20∼25 aspects. XQuAD results for different values λ other than 1 show similar bounds, consistent with the simulated results obtained by Vargas et al. (2012b).

5.3 Computation of probability of aspect coverage

Let $p(k, r|D, S)$ denote the probability that an aspect assigning function $c : [1, D] \rightarrow [1, S]$ covers the first r aspects but not more than the first k ones, i.e. $[1, r] \subseteq$

$Im(c) \subseteq [1,k]$ for $r \leq k \leq S$. This corresponds to

$$p(k,r|D,S) = \frac{|\{c : [1,D] \to [1,k] | Im(c) \supseteq [1,r]\}|}{|\{c : [1,D] \to [1,S]\}|}$$

The numerator can be computed by inclusion-exclusion principle, counting functions whose image lies in $[1,k]$ but don't cover j of the first r aspects. Let A_i be the set of functions $c : [1,D] \to [1,k]$ not covering aspect i. Clearly

$$\{c : [1,D] \to [1,k]\} \backslash \{c : [1,D] \to [1,k] | Im(c) \supseteq [1,r]\} = A_1 \cup \cdots \cup A_r$$

By inclusion exclusion principle, we get

$$|\{c : [1,D] \to [1,k] | Im(c) \supseteq [1,r]\}| = k^D - (|A_1| + \cdots + |A_r|) + \sum_{i<j} |A_i \cap A_j| - \cdots + (-1)^r |A_1 \cap \cdots \cap A_r|$$

For each $j \in [1,r]$ and each $1 \leq i_1 < \cdots < i_j \leq r$, $A_{i_1} \cap \cdots \cap A_{i_j}$ corresponds to functions omitting $\{i_1, \ldots, i_j\}$, i.e., functions $c : [1,D] \to [1,k] \backslash \{i_1, \cdots, i_j\}$. There are exactly $(k-j)^D$ distinct such functions for each choice of the set $\{i_l\}$. As there are $\binom{r}{j}$ forms of choosing these j uncovered aspects among the first ones, we get

$$|\{c : [1,D] \to [1,k] | Im(c) \supseteq [1,r]]\}| = \sum_{j=0}^{r} (-1)^j \binom{r}{j} (k-j)^D$$

Substituting these computations yields

$$p(k,r|D,S) = \sum_{j=0}^{r} (-1)^j \binom{r}{j} \left(\frac{k-j}{S}\right)^D \tag{5.3.1}$$

In particular, taking $k = r$, we get the probability of a system aspect distribution covering exactly the first k aspects

$$p(\text{cover exactly the } k \text{ first aspects}) = \frac{|\{c : [1,D] \to [1,S] | Im(c) = [1,k]\}|}{|\{c : [1,D] \to [1,S]\}|} = \sum_{j=0}^{k} (-1)^j \binom{k}{j} \left(\frac{k-j}{S}\right)^D \tag{5.3.2}$$

Chapter 6

Conclusion and future work

Aspect space selection constitutes one of the main problems in intent-oriented diversity methodologies. Several approaches have been taken. We can classify them as those based on the use of external subtopic information sources and those building inherent aspect spaces from the observed data. The first ones use a variety of sources of aspect information such as ODP categories, Wikipedia disambiguation pages or an explicit sub-query structure. On the other hand, the second ones usually rely either on applying an algebraic transformation to the space of document representations (like LSA) or on extracting semantic data through a latent variable language model.

Among the implicit aspect space building proposals, probabilistic frameworks have been proved to be specially effective for the diversity problem. The use of topic models for extracting latent semantic information lead to statistically robust estimations for the desired intent space. Probabilistic Latent Semantic Analysis and Latent Dirichlet allocations have been proved to be specially effective tools for this purpose. In the literature, they have been consistently used either as direct source of query subtopics or as part of a more complex aspect extraction algorithm.

A relevance aware version of the Probabilistic Latent Semantic Analysis have been developed as an application of a proposed utility-biased likelihood statistical framework. The described algorithm incorporates relevance estimation coming from the baseline ranking information to the pLSA dynamics, leading in an overall blind relevance feedback effect and allowing us to build query-specific more informative intent spaces. The obtained spaces have been empirically proved to make common diversifiers like IA-Select or xQuAD attain better diversity bounds.

The proposed framework allows us to build a great variety of intent space construction algorithms that incorporate several variables related by arbitrary latent models. Almost every kind of available prior information about the observed data can be incorporated to the analysis using the combination of the utility-biased methodology and the aggregation of the corresponding variables to the generative model. In contrast to other pLSA variants, the given geometric interpretation proofs that the introduced factors intrinsically perturb the dynamics of the algorithm.

The proposed methodology has a great degree of plasticity. The final simple Bayesian form of the E and M step computations, together with the adaptability to arbitrary generative models provide an almost universal tool that information

retrieval system designers can personalize to obtain domain-specific optimized aspect space extraction algorithms. This system freedom provides a vast amount of possible fields of application.

Moreover, as convergence properties, geometric interpretations, tempering and filtering techniques were developed at the level of the utility-biased EM algorithm, they descend trivially to any particular instance, providing additional optimization tools. Overall, the mathematical formalism of utility-biased estimators give all derived applications a common statistically strong theoretical base, independently of the specific properties of each model.

Finally, under certain simplified assumptions over the generative model ruling the relations between system and true aspects, explicit analytic formulas have been described for estimating the evolution of the diversity quality in terms of the choice of the aspect space size. This kind of explicit algebraic formulas do not exist in the literature for this context and solve partially a long open problem in diversity tasks, namely the choice of the optimal parameter space size.

The most important equations have been proved to be sound through testing against simulated baselines. An experiment with real data has been developed and qualitative evolution of some of the principal diversity metrics with respect to the aspect space size has been studied.

While the simplified model itself has not been able to fit the data properly, the overall combinatoric development can be used to work out explicit formulas for a generalized model. Some preliminary results of this model were showed, proving the overall validity of the described methods for analytically treating the problem of diversity dependency on the aspect size.

The proposed results open some possible research questions:

- While a nice amount of effort has been addressed to build an abstract theoretical framework for the incorporation or relevance and additional features to the pLSA dynamics, applications for this framework to tasks other than search and recommendation diversity have not been fully analyzed yet. In particular, experimental analysis of the proposed personalization and recommendation algorithms is to be performed in depth .

- The proposed application of the framework to a content-based recommender model was stated in a general way, without explicitly considering a particular feature model. There exist some instances of that algorithm that can be specially interesting and are worth considering. As a basic example, experiments in movie recommendation incorporating the described features (genres, director, cast, plot summary, etc.) would be interesting validating tools for measuring the true generality limits of the proposed abstract model.

- A particular interesting feature to incorporate to RapLSA general models would be the time variable. Considered as a discretized variable, time can be incorporated to the model taking a similar role that the one assumed by rating variables or relevance. As with these other kinds of variables, incorporating time directly to the model wouldn't necessarily alter significantly the dynamics of the algorithm. The key comes in using the utility-biased framework to introduce time-dependent utility functionals. For example, user-item ratings can

be pondered with time dependent terms that make closest to present ratings more relevant than older ones. This would correspond to the idea that more recent observations are more significant to the current user profile. A complete model description would be interesting and experimental data comparing its effectiveness to other time-aware models like time dependent Collaborative Filtering methods could lead to some interesting interactions.

- LDA being essentially a regularized version of pLSA, a natural question arises, is it possible to build a RaLDA(Relevance aware LDA)?, i, e., is it possible to mixture utility-biased formalism with Latent Dirichlet Allocation model. While it seems to be possible, it would require perturbing slightly the variational LDA methods.

- Matrix transition framework for the study of the dependency of diversity metrics to the choice of the aspect size has been addressed just as a theoretical possibility, and the corresponding effects have been only proved in a simulation context. It is based on substituting transition function $\varphi : S \to T$ by a transition matrix $(\varphi_{st})_{s \in S, t \in T}$, modeling the full conditional probability matrix $p(t|s)$ for each $t \in T$ and $s \in S$. Matrix versions of the described formulas have already been obtained, but a suitable experimental setup has not been found. In particular, we would be interested in describing a learning algorithm for estimating transition matrices from intrinsic data, without using the evaluation subtopics.

- Using the information about the optimal number of aspects for diversification provided in this work and the EM-based techniques exposed during the past chapters, the next step would be to build an auto-adjusting pLSA algorithm which automatically selects the number of aspects needed for representing the data and diversifying. Initial results involving the use of divergence regularization terms have been obtained, but the question remains open.

Bibliography

Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S. (2009). Diversifying search results. In *Proceedings of the Second International Conference on Web Search and Web Data Mining, WSDM 2009, Barcelona, Spain, February 9-11, 2009*, pages 5–14.

Amari, S., Nagaoka, H., and Harada, D. (2007). *Methods of Information Geometry*. Translations of Mathematical Monographs. American Mathematical Society.

Baeza-Yates, R. and Ribeiro-Neto, B. (2011). *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison Wesley.

Bellegarda, J. R. (1998). Exploiting both local and global constraints for multi-span statistical language modeling. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98, Seattle, Washington, USA, May 12-15, 1998*, pages 677–680.

Blei, D. M. (2012). Probabilistic topic models. *Commun. ACM*, 55(4):77–84.

Blei, D. M., Ng, A., and Jordan, M. (2003). Latent dirichlet allocation. *JMLR*, 3:993–1022.

Böhning, D. and Lindsay, B. (1988). Monotonicity of quadratic-approximation algorithms. *Annals of the Institute of Statistical Mathematics*, 40(4):641–663.

Buettcher, S., Clarke, C., and Cormack, G. (2010). *Information Retrieval: Implementing and Evaluating Search Engines*. MIT Press.

Byrne, W. and Member, S. (1992). Alternating minimization and boltzmann machine learning. *IEEE Trans. Neural Networks*, 3(4):612–620.

Cai, D., Mei, Q., Han, J., and Zhai, C. (2008). Modeling hidden topics on document manifold. In *In Proceedings of the ACM conference on Information and knowledge management*, pages 911–920.

Capannini, G., Nardini, F. M., Perego, R., and Silvestri, F. (2011). Efficient diversification of search results using query logs. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 17–18, New York, NY, USA. ACM.

Carterette, B. and Chandar, P. (2009). Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of the 18th ACM Conference*

on Information and Knowledge Management, CIKM '09, pages 1287–1296, New York, NY, USA. ACM.

Chapelle, O., Ji, S., Liao, C., Velipasaoglu, E., Lai, L., and Wu, S.-L. (2011). Intent-based diversification of web search results: metrics and algorithms. *Information Retrieval*, 14(6):572–592.

Chapelle, O., Metlzer, D., Zhang, Y., and Grinspan, P. (2009). Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009*, pages 621–630.

Clarke, C. L., Craswell, N., and Soboroff, I. (2010). Overview of the trec 2009 web track. Technical report.

Clarke, C. L. A., Craswell, N., Soboroff, I., and Ashkan, A. (2011). A comparative analysis of cascade measures for novelty and diversity. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 75–84.

Clarke, C. L. A., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., and MacKinnon, I. (2008). Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, pages 659–666.

Cormack, G. V., Smucker, M. D., and Clarke, C. L. (2011). Efficient and effective spam filtering and re-ranking for large web datasets. *Inf. Retr.*, 14(5):441–465.

Crane, M. and Trotman, A. (2012). Effects of spam removal on search engine efficiency and effectiveness. In *Proceedings of the Seventeenth Australasian Document Computing Symposium*, ADCS '12, pages 1–8, New York, NY, USA. ACM.

Croft, W., Metzler, D., and Strohman, T. (2010). *Search Engines: Information Retrieval in Practice*. Alternative Etext Formats. Addison-Wesley.

Csiszár, I. (1967). Information-type measures of difference of probability distributions and indirect observations. *Studia Scientiarum Mathematicarum Hungarica*, 2:299–318.

Csiszar, I. (1975). I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158.

de Leeuw, J. (1988). Convergence of the majorization method for multidimensional scaling. *Journal of Classification*, 5(2):163–180.

De Leeuw, J. and Heiser, W. J. (1977). Convergence of correction matrix algorithms for multidimensional scaling. *Geometric representations of relational data*, pages 735–752.

De Pierro, A. R. (1995). A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. *IEEE Transactions on Medical Imaging*, 14(1):132–137.

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

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Dominich, S. (2001). *Mathematical Foundations of Information Retrieval*. Library of Public Policy and Public Administration. Springer Netherlands.

Dumais, S. T. (1995). Latent semantic indexing (lsi): Trec-3 report. In *Overview of the Third Text REtrieval Conference*, pages 219–230.

Foltz, P. W. and Dumais, S. T. (1992). Personalized information delivery: An analysis of information filtering methods. *Commun. ACM*, 35(12):51–60.

Girolami, M. and Kabán, A. (2003). On an equivalence between PLSI and LDA. In *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 28 - August 1, 2003, Toronto, Canada*, pages 433–434.

He, J., Hollink, V., and de Vries, A. (2012). Combining implicit and explicit topic representations for result diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 851–860, New York, NY, USA. ACM.

He, J., Meij, E., and de Rijke, M. (2011). Result diversification based on query-specific cluster ranking. *J. Am. Soc. Inf. Sci. Technol.*, 62(3):550–571.

Hofmann, T. (1999a). Probabilistic latent semantic analysis. In *UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden, July 30 - August 1, 1999*, pages 289–296.

Hofmann, T. (1999b). Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*, pages 50–57.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196.

Hofmann, T. (2003). Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 259–266, New York, NY, USA. ACM.

Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):89–115.

Huber, P. (2004). *Robust Statistics*. Wiley Series in Probability and Statistics - Applied Probability and Statistics Section Series. Wiley.

Huber, P., Wiley, J., and InterScience, W. (1981). *Robust statistics*. Wiley New York.

Hunter, D. and Lange, K. (2002). Computing estimates in the proportional odds model. *Annals of the Institute of Statistical Mathematics*, 54(1):155–168.

Hunter, D. R., Lange, K., Biomathematics, D. O., and Genetics, H. (2000). Quantile regression via an mm algorithm. *J. Comput. Graphical Stat*, pages 60–77.

Hunter, D. R., Lange, K., Biomathematics, D. O., and Genetics, H. (2004). A tutorial on mm algorithms. *Amer. Statist*, pages 30–37.

Hunter, D. R. and Li, R. (2005). Variable selection using mm algorithms. *Ann. Statist.*, 33(4):1617–1642.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical Review*, 106(4):620–630.

Jin, R., Chai, J. Y., and Si, L. (2004). An automatic weighting scheme for collaborative filtering. In *In SIGIR 04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 337–344. ACM Press.

Kendall, M. (1948). *Rank correlation methods*. Charles Griffin & Company Limited.

Krestel, R. and Fankhauser, P. (2012). Reranking web search results for diversity. *Information Retrieval*, 15(5):458–477.

Kullback, S. (2006). A lower bound for discrimination information in terms of variation (corresp.). *IEEE Transactions on Information Theory*, 13(1):126–127.

Landauer, T. (2007). *Handbook of Latent Semantic Analysis*. University of Colorado Institute of Cognitive Science Series. Lawrence Erlbaum Associates.

Landauer, T. K. and Dutnais, S. T. (1997). A solution to platos problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *PSYCHOLOGICAL REVIEW*, 104(2):211–240.

Lange, K. and Fessler, J. A. (1995). Globally convergent algorithms for maximum a posteriori transmission tomography. *IEEE Transactions on Image Processing*, 4(10):1430–1438.

Ma, S., Chuanyi, J., and Farmer, J. (1977). An efficient EM-based training algorithm for feedforward neural networks. *Neural Networks*, 10(2):243–256.

Neal, R. and Hinton, G. E. (1998). A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers.

Ng, S.-K. and McLachlan, G. J. (2004). Using the EM algorithm to train neural networks: misconceptions and a new algorithm for multiclass classification. *Neural Networks*, 15(3):738–749.

Ortega, J. and Rheinboldt, W. (1970). *Iterative Solution of Nonlinear Equations in Several Variables*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).

Pinsker, M. (1964). *Information and information stability of random variables and processes*. Holden-Day series in time series analysis. Holden-Day.

Radlinski, F. and Dumais, S. (2006). Improving personalized web search using result diversification. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 691–692, New York, NY, USA. ACM.

Rafiei, D., Bharat, K., and Shukla, A. (2010). Diversifying web search results. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 781–790, New York, NY, USA. ACM.

Rand, W. M. (1971). Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850.

Rose, K. (1991). *Deterministic annealing, clustering, and optimization*. PhD thesis, California Institute of Technology.

Rose, K., Gurewwitz, E., and Fox, G. (1990). A deterministic annealing approach to clustering. *Pattern Recogn. Lett.*, 11(9):589–594.

Sabati, C. and Lange, K. (2002). Genomewide motif identification using a dictionary model. In *Proceedings of the IEEE*, volume 90, pages 1803–1810.

Santos, R. L., Macdonald, C., and Ounis, I. (2012). On the role of novelty for search result diversification. *Inf. Retr.*, 15(5):478–502.

Santos, R. L. T., Macdonald, C., and Ounis, I. (2010). Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 881–890.

Tombros, A., Villa, R., and Van Rijsbergen, C. J. (2002). The effectiveness of query-specific hierarchic clustering in information retrieval. *Inf. Process. Manage.*, 38(4):559–582.

Ueda, N. and Nakano, R. (1998). Deterministic annealing {EM} algorithm. *Neural Networks*, 11(2):271 – 282.

Vallet, D. and Castells, P. (2012). Personalized diversification of search results. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pages 841–850.

Vargas, S., Castells, P., and Vallet, D. (2011). Intent-oriented diversity in recommender systems. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 1211–1212, New York, NY, USA. ACM.

Vargas, S., Castells, P., and Vallet, D. (2012a). Explicit relevance models in intent-oriented information retrieval diversification. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pages 75–84.

Vargas, S., Castells, P., and Vallet, D. (2012b). On the suitability of intent spaces for ir diversification. In *International Workshop on Diversity in Document Retrieval (DDR 2012) at the 5th ACM International Conference on Web Search and Data Mining (WSDM 2012)*, Seattle, Washington, USA.

Welch, M. J., Cho, J., and Olston, C. (2011). Search result diversity for informational queries. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 237–246, New York, NY, USA. ACM.

Wu, T. T. and Lange, K. (2010). Multicategory vertex discriminant analysis for high-dimensional data. *Ann. Appl. Stat.*, 4(4):1698–1721.

Yamano, T. (2014). A note on bound for Jensen-Shannon divergence by Jeffreys. In *ECEA-1: 1st International Electronic Conference on Entropy and its Applications, November 3-21*, page b002.

Zhai, C. (2009). *Statistical Language Models for Information Retrieval*. Synthesis lectures on human language technologies. Morgan & Claypool.

Zhai, C. X., Cohen, W. W., and Lafferty, J. (2003). Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 10–17, New York, NY, USA. ACM.

Zhou, H. and Lange, K. (2010). Mm algorithms for some discrete multivariate distributions. *Journal of Computational Statistics*, 19(3):645–665.