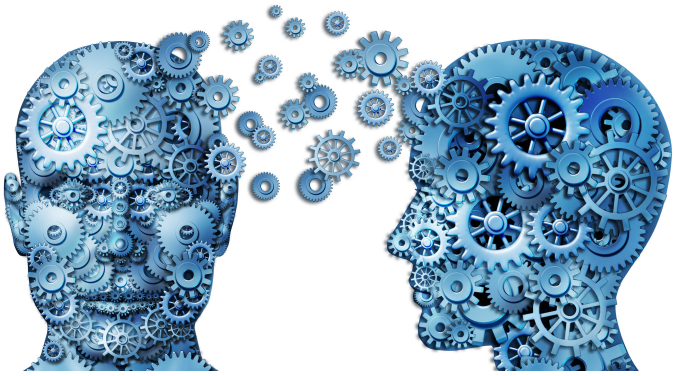
Probabilistic Models for Personalized Faceted Search

Matthijs van Dorth



Electrical Engineering, Mathematics & Computer Science
Pattern Recognition & Bioinformatics
Delft University of Technology

July 5, 2017

Probabilistic Models for Personalized Faceted Search

Matthijs van Dorth
Delft University of Technology

Electrical Engineering, Mathematics & Computer Science
Pattern Recognition & Bioinformatics

Matthijs van Dorth
July 5, 2017
Supervisors:
Prof.dr.ir. M.J.T. Reinders
Prof.dr. M. Loog
dr.ir. C.C.S. Liem
dr.J. Kooij

Delft University of Technology
Electrical Engineering, Mathematics & Computer Science
Pattern Recognition & Bioinformatics
Mekelweg 4
2628 CD, Delft

Abstract

Faceted search is a useful technique to search through large amounts of semi-structured data. It is often used in applications such as (digital) libraries, e-commerce sites and other retrieval systems with multidimensional properties.

Faceted Search allows users to refine queries in order to find documents faster. It gives suggestions for narrowing down the retrieved documents in order to find or suggest documents the user is looking for. It is therefore used for both finding specific documents in a corpus or exploratory search, where people do not know precisely which document he/she is looking for but does recognize one if it is presented to the user. Examples of this is people looking for a new job, apartment or holiday.

In personalized faceted search we take this approach even further by recommending specific facets that the documents must adhere to in order to find the most relevant documents to the user. Based on the characteristics of each user we can make specific recommendations for the type of documents the user is looking for.

In this thesis we will focus on implicit feedback from the user in contrast to explicit feedback in previous work in which the user has to mark each document it sees as relevant or not. We will use real data collected from a real website and show that probabilistic hierarchical models can improve the relevancy to the user substantially, especially when we do not have much data on a user, often referred to as the cold start problem.

Contents

Introduction

1

1.1 Faceted Search

When users are searching or exploring a large set of documents, it can often be hard to find the relevant documents. Whether these documents are products on an e-commerce site, movies, books or webpages, users are often presented a with search interface that allows them to define queries to search through all the documents.

Using a query, a search engine can search through all documents and return the documents most relevant to that query. In Figure 1.1, we see a user searching for "Faceted Search" on the TU Delft library website and shows the 10 most relevant documents on the first page.

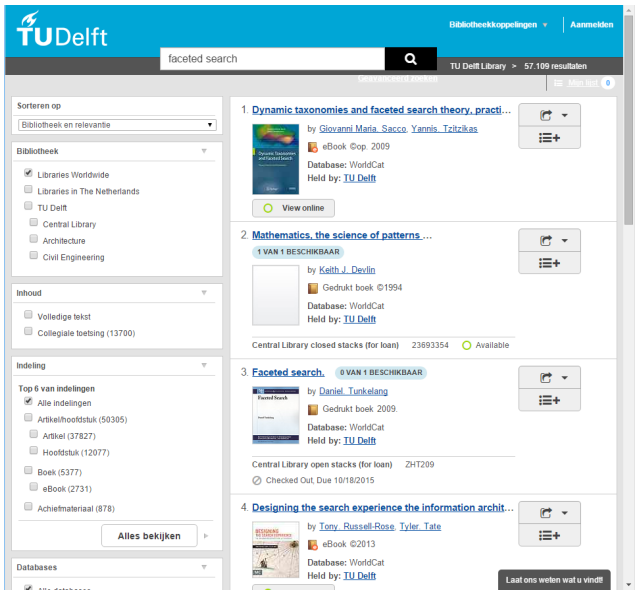


Figure 1.1.: Searching for "Faceted Search" on the TU Delft library website.

Search engines are a great way to search through large information spaces. Google searches through 20 billion webpages, Amazon allows you to search through thousands of articles and LinkedIn allows you to search through millions of people, companies and vacancies. One drawback of this is that there are many results, and people are very often only interested in a specific part of those results. Different people can even mean different things with the same query. People can try to formulate more complex search queries, but this can be difficult and most users do not know how to do this correctly [22].

Very often these search engines allow the user to click through these pages showing the next 10 results of the query. In Figure 1.1 we see a query that matched a total of 57.109 documents. Scrolling through all these results can be very time consuming so users can try to refine the query or make use of some advanced query commands.

Furthermore, a query can be too broad, too specific or ambiguous. This problem becomes even more evident when users are searching on mobile devices such as smart-phones and tablets and typing in long queries can be even more inconvenient.

One often used solution to this is **faceted search** (sometimes called faceted navigation) to narrow down the amount of results and guide the user to create more complex search queries. In a faceted search interface, such as in Figure 1.2, a user is presented with fields on which a user can narrow down the results. These fields are called **facets** and the terms on

which can be narrowed down the **facet-values**. When a user selects one of the facet-values, only documents that include those values are shown.

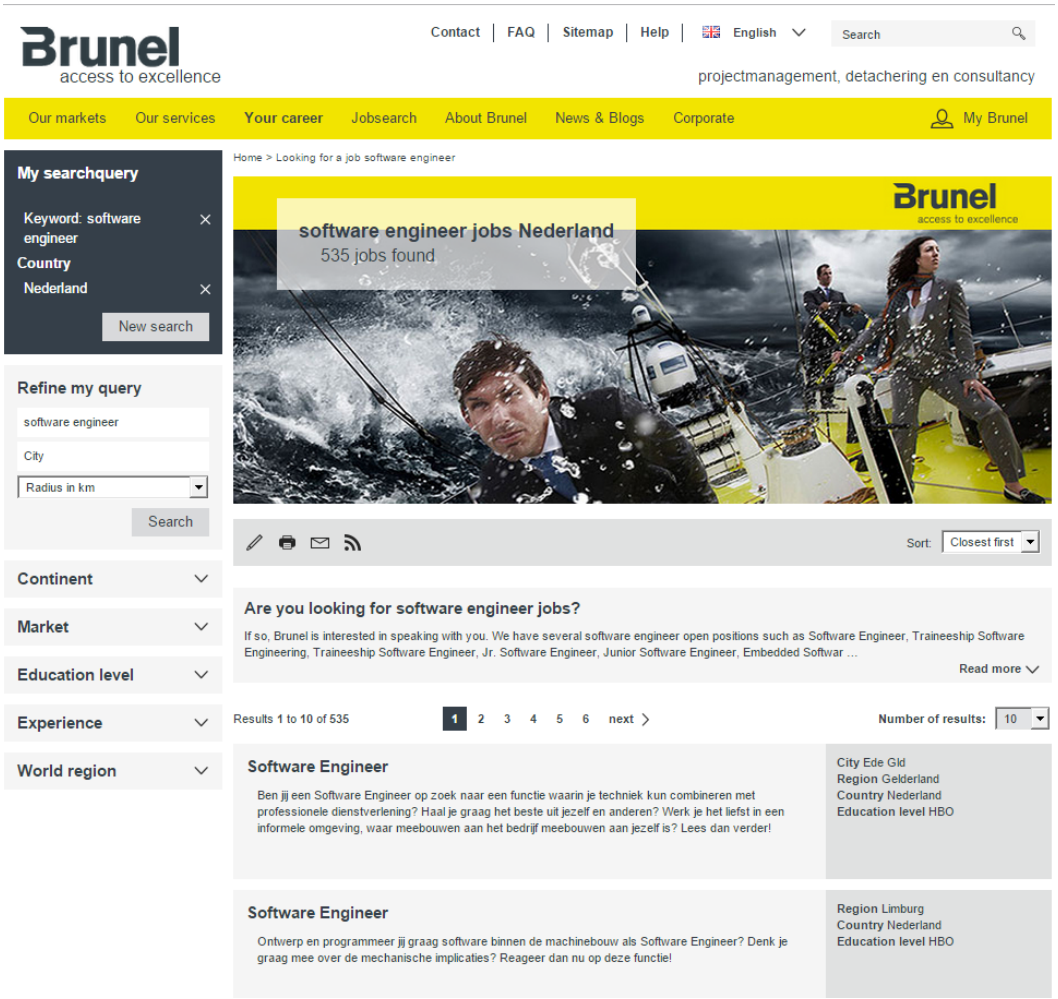


Figure 1.2.: Faceted Search on Brunel

An increasingly popular method is to present the user with several options to refine the retrieved documents. For example in Figure 1.1, we can restrict the results to only show articles or to restrict the result to show only documents that where published in the last 5 years. This allows the user to narrow down the amount of results and find a relevant document much easier.

1.2 Personalized Faceted Search

Although faceted search can help users formulate complex search queries, the number of facet-values is often very large. Take for example a facet `Country`, it is hard for users to shift through all the 196 countries in this list, even if they are sorted in alphabetic order. It is therefore not advisable to show all possible facet-values in the interface. A lot of faceted search interfaces show the most used facets with the amount of documents in each facet-value in descending alphabetic order.

One of the emerging trends of the last decade on the web is personalisation [3]. Personalisation is the adaption of web-pages based on the characteristics of the user. **Recommender systems**, probably the best known form of personalisation, play a huge role on website like Amazon, Netflix and Facebook. Even though a lot of the content in Amazon is personalized

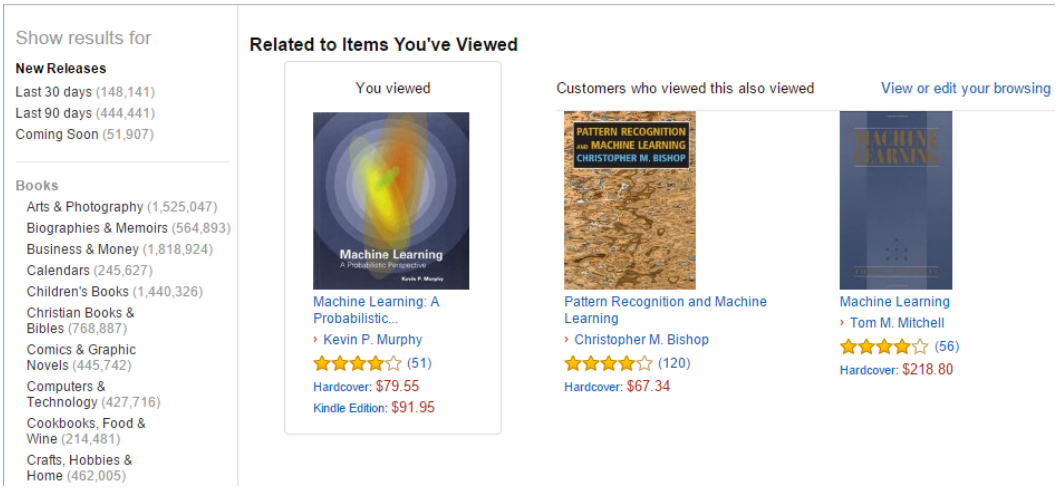


Figure 1.3.: Faceted Search on Amazon

based on previous viewed and bought items, the search itself is not, as can be seen in Figure 1.3.

The approach we will take is to personalize the faceted search interface in such a way that the user has to do the least amount of effort to find the item the user is looking for. This can be done by only showing the most relevant facet-values to the user. The assumption we make here is that although the user might not know all possible documents, the user is able the recognize a document that might be relevant from a list.

Different approaches for personalisation exist. Most systems for personalisation rely on implicit feedback [13]. Implicit feedback means that we are not asking a user directly about his or her preferences but try to infer those based on actions by the user, such as pages viewed, items purchased or the users' location. Although explicit or hybrid solutions also exits, they are used less often since they require the user to explicitly provide answers to what their preferences are [5]. In this thesis we will focus only on implicit data generated by the users.

1.3 Research Questions

In this thesis a probabilistic perspective of faceted search is taken. We estimate the probability of a user selecting a certain facet-value in order to maximize the efficiency of clicking to the right document. This means that we want minimize the amount of clicks necessary before the user arrives at the right target. For this project we implement a user modeling application. Using this application several methods for faceted search will be evaluated. This application will be developed for Brunel, a project management, recruitment and consultancy firm that allows users to search through a large set of vacancies. These vacancies include facets such as location, branch, type of job etc. Using faceted search several methods for faceted ranking will be compared. We will also look at how accurate these models are when we do not have a lot of data about a user and also look at how these models perform when we only show a fraction of the facet-values.

In this thesis we try to answer several questions. One main question (MQ) and several specific research questions (RQ):

- **MQ:** *How can we improve personalized faceted search when we do not have much data from a user?*
- **RQ1:** *How does adding a prior affect the performance of a personalized faceted search system?*

- **RQ2:** *How does the amount of data we have of a user affect the performance of personalized faceted search?*
- **RQ3:** *What is the error we make when we only show a fraction of the facet-values?*

1.4 Contributions

This thesis shows several methods of how we can create and estimate user models in order to allow users to perform faceted search more effectively. We extend the models created in [14] and perform evaluations on a real world dataset and discuss their performance. We will focus on how these methods perform for new users for which we do not have a lot of data yet. We will show with different experiments how much data we need to accurately learn these individual user models. Furthermore we suggest a method for evaluating these models that can evaluate the performance of a model when we want to show only a fraction of the facet-values.

1.5 Outline

Chapter 2 gives an overview of the background needed to understand the rest of this thesis and introduces some notation that we will use throughout this thesis. An explanation of the different methods is given in Chapter 3. Chapter 4 shows how we can evaluate each of the proposed methods from the previous chapter. The results of these evaluations are shown in Chapter 5. Finally, Chapter 6 concludes this thesis and discusses future options for research. In the appendix additional background material is provided.

Background & Previous Work

2

2.1 Document and User Models

2.1.1 Document Model

To allow documents to be filtered by facets, each documents in the corpus must have additional meta-data. This meta-data must have some properties in common in order to compare them and group documents together. An ontology is a description of the properties and relationship between items within a certain domain. We can view this meta-data as key-value pairs added to each of the documents. The keys are called the *facets* and can be set mandatory or optional, it may even be possible for a single facet to have multiple *facet-values*. Each facet-value belongs to only one facet. These key-value pairs are often called facet-value pairs (FVP). The amount of facet-values per document may differ if there are facets which may have more than one facet-value or in which there are facets that are set optional. The total amount of facet-value pairs is the sum of all facet-values for all facets.

A set of documents that share the same structure is what we call a corpus. Documents in a corpus often share a common topic, such as documents about movies or vacancies. These documents share the same structure in that they share the same facets and same possible facet-values. For example a corpus about movies may have a facet such as `Genre`, with facet-values such as `Action`, `Comedy` or `Drama`. A corpus about vacancies may have a facet such as `Market` with facet-values: `Industry`, `Services` and `Health Care`. An example document of such a vacancy taken from the Brunel corpus is shown in Table 2.1. A full description of this Brunel dataset is provided in Appendix B.

Facet	Facet-value
Title	Software Engineer
Market	Industry
Branch	High Tech
Country	Netherlands
Region	Noord Brabant
Area of expertise	Research & Development
Level	Specialists & experts

Table 2.1.: Example Vacancy Document

Different corpora thus can have different facets and each is often predefined into what values it can take. In each document we can have different FVPs. In some cases a facet may only have a single facet-value and the facet-values are mutually exclusive or a facet that can have more than one facet-value. Lastly the underlying facet-values can have different types or scales which are shown in Table 2.2.

Type	Values	Example
Nominal	Categorical	Gender(male/female)
Ordinal	Ordered Categorical	Rank(1st;2nd;3rd)
Interval	Numbers	Temperature(Celsius)
Ratio	Numbers	Duration(seconds)
Free-Text	All possible values	Text(name)

Table 2.2.: Different scales for the facets

The scales or level of measurement corresponds to how the data is represented. For example for ratio a user can select a number that satisfies the requirement of what the user is looking for. Or a category that conforms to the requirements of the user. In this thesis we will focus on (ordered) categorical values of which there is only one facet-value possible per document per facet. It is however fairly straightforward to generalize this assumption by simply adding multiple facet-values to a document and do a (weighted) sum of the facet-values seen.

2.1.2 User Model

We can define a user models in the following way. A user model is a function that can identify whether or not a document or item is relevant to the user. It can do this based on previous experience from the user in order to know what the user might like and not like. Therefore a user model also contains information about previous actions.

In general we assume a user is searching for a document that contains a combination of certain facets-values. If we look at how users search for vacancies they often search for a specific market and location to work. If a user is looking for a movie to watch he/she often has a preference for a certain genre or actor that is playing in the movie. In these types of exploratory search, faceted search is often helpful and can guide a user to find a specific item. In order to allow a user to find a document faster we can propose certain facet-values such that a user can select this facet in order to narrow down the results.

We will model each user based on the features of the ontology we are using. This ontology based user modelling [20] is often used in application like this. Each of the facet-values in the ontology corresponds to a value in the user model that indicates the preference of the user to that facet-value. Although we could add additional data about each user the focus will be on features available in the ontology.

2.2 Notation & Terminology

Now that we have described a high level description into what we refer to as the document model and user model, we will introduce some notation.

We define the following terms:

- A *corpus* is a collection of M documents denoted by $\mathcal{D} = \{x_1, \ldots, x_M\}$.
- A *document* is represented as a set of J facets, which are represented by a list of vectors $x_m = (f_1, \ldots, f_J)$, where f_j is the j^{th} facet.
- The *facet-values* are the components of this vector $f_j = (v_{j,1}, \ldots, v_{j,L})$ and are represented such that $v_{j,l} = 1$, if the l^{th} *facet-value* of facet j is present in the document; $v_{j,l} = 0$ otherwise.
- We define the length of this vector f_j to be L_j or just L if this is clear from the context.
- The total number of facet-value pairs is the number of all facet-values for all facets and is defined by: $K = \sum_1^J L_j$
- A *user* u is a visitor that has seen one or more documents and is indexed by $\{1, \ldots, U\}$.
- $\mathcal{X}_u = (x_1^u, \ldots, x_N^u)$ is a sequence of N_u documents seen by user u. Sometimes we will use just N if this is clear from the context.

Our main goal is to find the the user models captured by parameter θ_u for each user $u \in 1, \ldots U$, using different models **M**, which we will introduce in Chapter 3. Each of these different models assign different probabilities to the facet-values for each user, $p(v_{j,l})$. This user profile is a K-dimensional vector of all facet-values and indicates the preference for each facet-value to the user.

2.3 Previous Work

An overview of faceted search is provided in the book “Faceted Search” by Tunkelang [23]. In this book a short history of faceted search is provided, with some examples of commercial applications, such as on eBay and Amazon. Besides giving some insights into implementing a faceted search interface from both a front-end as a back-end perspective, this book provides some basic evaluation metrics such as precision and recall.

In the paper, “Beyond basic faceted search” by Ben-Yitzhak et al. [1] the authors describe *dynamic facets*, a way to add facets to documents based on the context, such as documents released in the last year or documents that have some spatial information in them such as: "give documents within a certain radius of my location".

Typically two types of recommender systems are used. The first one, content based recommender systems, recommend items based on the characteristics of the item and the preferences for the item by the user [19]. The second type is collaborative based recommender systems and recommends items based on items that similar users have seen [21].

Faceted search can be seen as a special type of a conversational recommender system [17]. One of the earlier work on interactive conversation systems used critiquing to ask the user to critique certain items. Based on the responses of the user the system was better able to suggest items that the user liked or marked as positive and remove the items that the user did not like or marked as negative or not relevant.

The earliest conversation like systems are critiquing based recommender systems. Critique-based recommender systems allow users to give feedback on individual recommendations such as "give me more like this, but located closer" [7]. For each item in the catalog, different features need to be available. Critique-based recommenders allow the user to navigate through the result set. This is especially useful if the user does not know precisely what the user is looking for.

The items can be fictional items or some popular items that need to be different enough to make an estimate of what characteristics in each item the user likes and which not. [17]

Another type of interactive dialogues is navigation-by-asking [7, 16]. This gives users the possibility to directly specify which features are required and therefore it allows user to very quickly navigate through the result set, in this case it is assumed that the user knows more precisely what he/she is looking for, which may not always be the case.

Navigation-by-proposing is a technique that gives the user several possibilities and asks the user to rate these items whether they are relevant or not. Based on these (fictional) items, recommendations can be made most related to the relevant items [24]. Instead of asking for features directly the system has to infer them based on the proposed items. This technique is different from Navigation-by-asking in that now the user has to rate proposed items and in which the system refines the possible proposed item until the right item is found.

In *The adaptive web: methods and strategies of web personalization* by Brusilovsky, Kobsa, and Nejdl [3], Adaptive Systems are defined as specially tailored pages that appeared differently or contained different content based on the individual user. The most well -known type of Adaptive Systems are Recommender Systems, such as those that can be found on Amazon [1], Netflix [2] and LinkedIn[3].

In a paper by Koren, Zhang, and Liu [14], Personalized Faceted Search was introduced which used a hierarchical model. They used this on the MovieLens [4] dataset. This dataset contains ratings by users, however only users are included that rated at least 20 movies. These movies where also explicitly rated as either positive or negative.

[1] `https://www.amazon.com`
[2] `https://www.netflix.com`
[3] `https://www.linkedin.com`
[4] `http://www.grouplens.org`

There has been a lot of research in ranking documents, such as news articles [9, 10], that use collaborative filtering or content based filtering, to prevent the cold-start problem. In the cold-start problem we have a new user or a new item that is added to a system and because we do nog have enough data from it we can not make accurate predictions for it.

Although in our system the facets and facet-values are already given, sometimes they are not present. In [6] the authors add additional semantics to text, in this case tweets, to add facet-values to the documents. This way unstructured documents can still be searched through efficiently.

Graphical models have been used in many different applications. In [2] Latent Dirichlet Allocation (LDA) is described. LDA is a probabilistic model to model collections of discrete data such as topics in a corpus of documents, where topics are defined as distributions words, and express which words typically co-occur. For each document the words are given and from these words the topics in these documents are inferred. The number of topics must be set as hyper-parameter.

Another application of graphical models is TrueSkill [12] in which the skill of a very large number of users is inferred based on the interaction between the users. This system is able to update the skill of the players each time they interact with another player. Compared to the the classical approach this method converges much faster.

Hierarchical user modeling such as in [27] have been used to model users to avoid overfitting to a specific users' data. In this case this has been applied to a recommendation system for movies and news.

Hierarchical Models are used to study a wide range of topics. The standard textbook on this is by [11]. Hierarchical models study the effects of subgroups within groups of data. These types of models work very well when there is a high imbalance between the size of the groups.

Methodology

3

3.1 Introduction

In this chapter we will focus on different methods for estimating the probability of a user being interested in a certain facet-value. This assumes that the most relevant facet-values have the highest probability of being selected. Therefore we can rank each of the facet-values within each facet in order of probability of being selected. This chapter describes how we can model our faceted search problem as a probabilistic model in order to find the most relevant facet-values for each user. This chapter focuses on parametric models; the data will be assumed to be sampled from distributions in the **exponential family**.

In table 2.2 we saw how facet-values could have different types. In parametric models we can assign different probability distributions to each of these types. For categorical facets we can use discrete distributions and for numerical facets we use continuous distributions. In this thesis we will make no distinction between interval and ratio types and between nominal and ordinal types. Furthermore in this thesis we will focus on these categorical values.

3.2 Document Count

The first method we discuss is ranking the facet-values in order of the amount of documents, showing facet-values with a lot of documents above and facet-values with a lower amount of documents lower. This is the default for Brunel and many other websites. The rationale behind this is that the probability that a facet-value is is chosen is higher when there are more documents that have this facet-value. This method may work well if each document has about the same probability of being selected.

This model is not using any interaction data from the users. It only uses data from the documents itself. We will refer to this model as the document count model or $\mathbf{M}^{Count}$. The ranking of each of the values is therefore the same for each user.

3.3 Popularity

A slightly different method is to order the documents by popularity. Thus, ordering the facet-values by how often the underlying documents have been seen. This means that facet-values with a lot of documents but not a lot of views can be ranked lower than a facet-value with few documents, but that are very popular.

This model does require data from all users in order to count how often each of the documents have been seen. For the remainder of this thesis, we will refer to this model as the document popularity model or $\mathbf{M}^{Pop}$. Just as the previous method this method is not personalized and thus has the same ranking for each user.

3.4 Maximum Likelihood Estimation

The next method is of Maximum Likelihood (ML) or Maximum Likelihood Estimation (MLE). In ML we take a very simple approach. Based on the previously seen documents we count the amount of facet-values in all the documents that were seen by the user and sort the facet-values on how often each of the facet-values were seen in the documents. Based on the counts of each of the facet-values we calculate the probability of each of these facet-values.

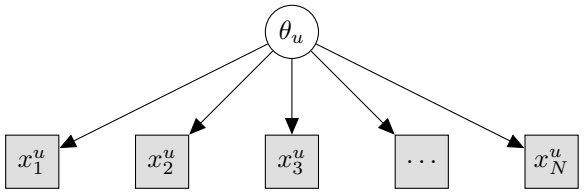
This method gives a personalized ranking for each user based on the previously seen documents. For each user we need to maintain a list of all the documents this user has seen

and count the facet-values in these documents and return for each facet the ordered list of facet-values.

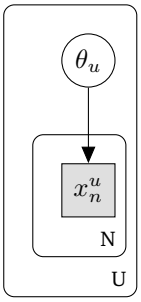
For each user we maintain a parameter θ_u, of each users' distribution, the user profile, such that it maximizes the probability that it generated this data X. We can see this as a **generative** model. Given data $\mathcal{X}_u = \{x_1^u, \ldots, x_i^N\}$, the documents seen by user u, we estimate θ_u directly using our data $\mathcal{X}_u$. Which is simply the normalized count data for each of the facets, this is often referred to as the empirical distribution. We will refer to this model as $\mathbf{M}^{ML}$. The viewed documents by a user are conditionally independent given the model parameters, which is why you can write as Equation .

$$\begin{aligned} \theta_u^{ML} &= \arg\max_{\theta_u} p(\mathcal{X}_u|\theta_u) \\ &= \arg\max_{\theta_u} \prod_i^n p(x_i^u|\theta_u) \end{aligned} \tag{3.1}$$

We can visualize this model as in Figure 3.1a. We have a distribution with parameter θ_u from which the documents x_u are "drawn", with the facet-values generated from the distribution with parameter θ_u.



(a) MLE as draws from a distribution



(b) MLE in plate notation

Figure 3.1.: Maximum Likelihood Estimation

Although this is still a simple graphical model, we prefer simpler notation. Thus we resort to plate notation as shown in Figure 3.1b [4, 26]. The plate around the x_u with the N in the lower right corner means that the x_u are repeated N times, for N documents for each user. The large plate with the U means that we have to estimate θ_u for each of the U users.

For facets that contain one and only one facet-value per document, each of these draws $x_{u,1} \ldots x_{u,N}$ can be modeled by a draw from a categorical distribution. For each of the k facet-values we have have a probability p_i of it being chosen, such that $\sum_{i=1}^k p_i = 1$. When we draw N times from this distribution, this generalizes to the multinomial distribution.

3.4.1 Implications

Maximum Likelihood Estimation is certainly one of the simplest and most natural approaches. It is very easy to calculate and easy to update with new data. The largest drawbacks however are that with ML it is very easy to overfit, especially when we have a small amount of data. Moreover, we can't say anything if we have no data at all. A second major drawback is that ML is a point estimate. We calculate a single number that represents the probability of a certain outcome. There is no way to express our certainty or uncertainty with ML.

Another observation we can make is that some facet-values may have a probability of 0. Which may seem strange, because just because we have not seen anything from happening, does not mean it cannot happen in the future. It can always happen at a point later in time, that a user will be interested in a facet-value estimated with a probability 0, which will

contradict our model. Also, how can we rank our facet-values when some of them have the same probability?

3.4.2 Example

For example if a user has seen 3 documents and of these documents two have the facet-value `Services` and one has the facet-value `Industry`. Then within the `Market` facet, we will rank `Services` first, because it has a probability of $2/3$ and `Industry` second with a probability of $1/3$. We can visualize this as in Figure 3.2. Here we see the probability for each of the facet-values, when a user has seen 0, 1 and 3 documents. The probability of 0 at the start is contradicted for the `Industry` and `Services` facet-values after seeing all 3 documents.

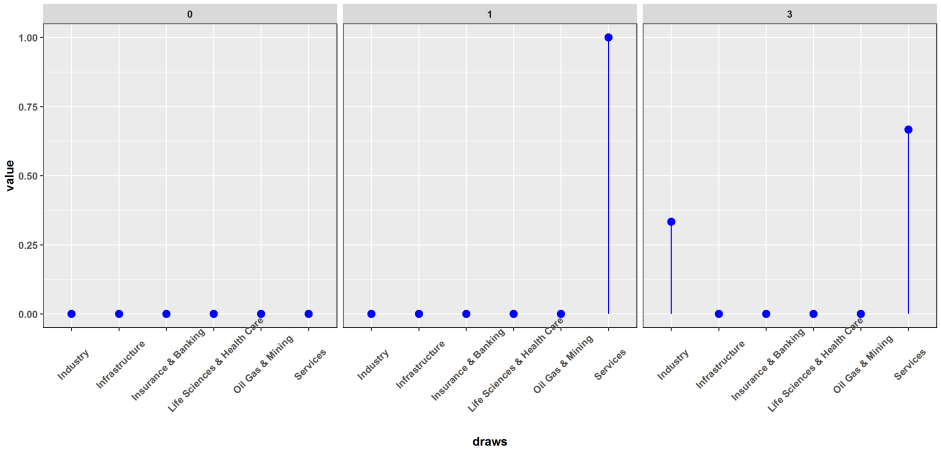


Figure 3.2.: A Maximum Likelihood user model for the `Market` facet with 0,1 and 3 viewed pages

If we take a look at all the users, we can get an idea of how each user prefers the different facet-values. In Figure 3.3 we can see a ternary diagram for three of the six facet-values of the `Market` facet. We observe that most of the points are on the corners (not very well visible since they overlap) and the borders. We also observe that most of the users only view a very low amount of documents, which may be the reason that most of the users are at the corners or border.

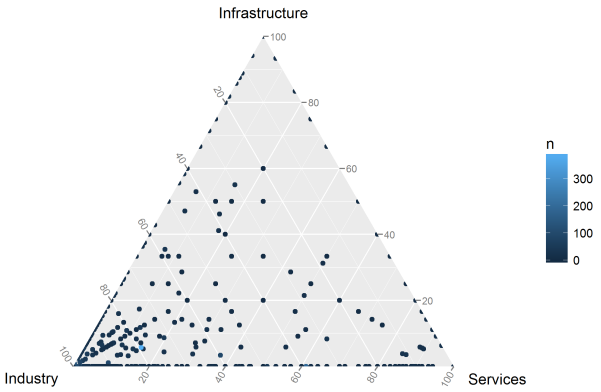


Figure 3.3.: MLE for 3 of the 6 different Markets

3.5 Maximum A Posteriori Estimation

A major drawback in Maximum Likelihood is that it is easy to overfit. We already saw in the examples of the previous section that for some users an estimation of 0 or 1 was given for the likelihood of a facet-value. Maximum A Posteriori (MAP) tries to avoid that by using a prior. A prior is a distribution that expresses our belief of what the probabilities are without having seen any data yet.

One method to do this is Laplace Smoothing or Additive Smoothing. This means that instead of starting a 0, we add a non-zero value. By adding this to the user data we get more smoothed probabilities. We can model the user models for every facet again as a multinomial distribution. Although we can choose any distribution as our prior, we prefer to choose a conjugate prior, since this allows for easy updating of our user model. For the multinomial distribution the conjugate prior is the Dirichlet distribution [**BishopXXXX**]. The Dirichlet distribution is a generalization of the Beta distribution and is parameterized by a vector $\boldsymbol{\theta_0}$ of length k. This k is the amount of categories in the multinomial distribution. For $k = 2$, this is a Beta distribution. If we take a prior of $Dir(\boldsymbol{\theta_0})$ with $\boldsymbol{\theta_0}_i = 1$ we add pseudo-counts of 1.

By using Bayes' rule we can update our prior distribution with new data to estimate our new beliefs about our posterior probability. This Bayesian updating makes it very easy to do on-line updating as new data comes in. Since the Dirichlet distribution is the conjugate prior of the multinomial distribution when we update a Dirichlet distribution with a multinomial distribution we get again a Dirichlet distribution.

$$\begin{aligned} \theta_u^{MAP} &= \arg\max_{\theta_u} p(\theta_u|\mathcal{X}_u) \\ &= \arg\max_{\theta_u} \frac{p(\mathcal{X}_u|\theta_u)p(\theta_u)}{p(\mathcal{X}_u)} \\ &= \arg\max_{\theta_u} p(\mathcal{X}_u|\theta_u)p(\theta_u) \end{aligned} \tag{3.2}$$

In Equation 3.2 we use the data as given, to find θ, we can rewrite this using Bayes' rule. Since θ doesn't depend on the denominator we can remove this from the equation.

The thing we do need now is find a prior that generalizes well over all data. This parameters of this prior are fixed for all users. We will call the parameters of this fixed prior θ_0. The complete model is shown in Figure 3.4.

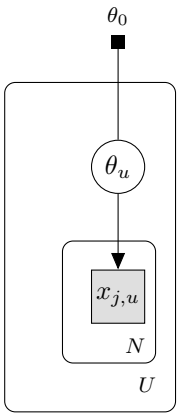


Figure 3.4.: Maximum A Posteriori with fixed prior θ_0

When we have a new user with no observations yet, our estimation is simply defined by the prior distribution. Every user that has no observations starts with the same estimation. When

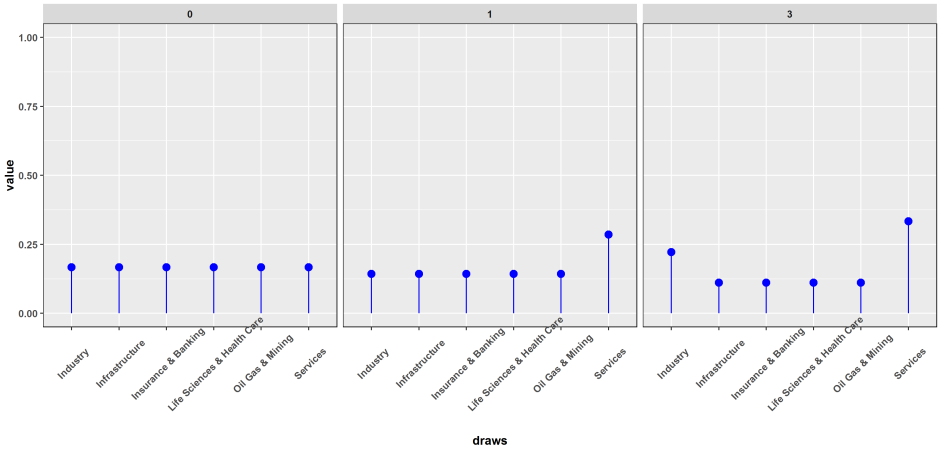


Figure 3.5.: a Maximum A Posteriori user model for the `Market` facet with 0,1 and 3 viewed pages ($\theta_0 = 1$)

this user is making an observation we can update our prior with the likelihood to obtain the posterior predictive distribution and make our new estimation based on this distribution.

3.5.1 Implications

The main difference between ML and MAP is that we treat θ as a random variable for the MAP model. By adding prior knowledge we can avoid overfitting on the data. Just as in the ML model, the MAP estimate is still a point estimate. The problem now is to choose a good prior that fits the data well. This prior distribution in Bayesian statistics represents our prior belief in what the resulting posterior distribution should look like. For this we can have two different types of priors that we can add. The first is a non-informative prior. The second one is a (weakly) informative prior.

3.5.2 Example

non-informative priors

Returning to the same example as in Maximum Likelihood model, with two facet-values `Services` and one `Industry`, we can start with a flat non-informative prior of $\theta_0 = 1$. This results in an equal probability if we do not have any data from a user yet, but this changes as we get more data from a user. This way the probability of selecting a facet-value can never get 0. Figure 3.5 shows the same example for this flat prior.

One thing we notice is that none of the facet-values will get a zero probability so that this model will be closer to what we actually think is reality. If we look at all the users in Figure 3.6, we see using the ternary diagram that the users are not cluttered against the corners and edges anymore, but more smoothed out. Another often used option is to add a Jeffrey's prior in which all the θ_0 values are $\frac{1}{2}$ [**BishopXXXX**].

(weakly) informative priors

Based on what we know of each of the facet-values it might not be appropriate to give each of the facet-values the same prior probability. Since we know that some facet-values are more common than other facet-values we can add that information in the prior. Informative priors can add prior knowledge that you have into the model such that it will resemble your believe into what the probabilities should be more closely.

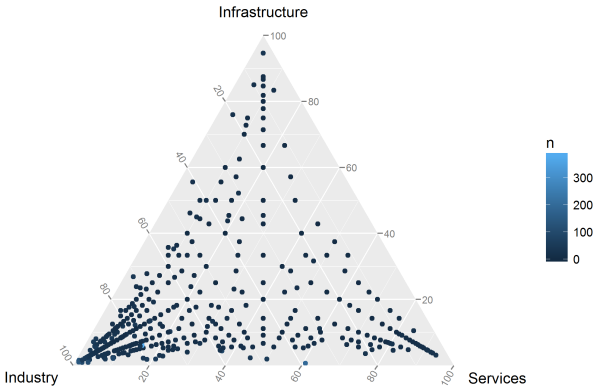


Figure 3.6.: MAP with Dir(.5,.5,.5) prior for 3 of the 6 different Markets

3.6 Expectation Maximization for MAP estimates

Although the probabilities of each of the facet-values for the MAP model now make more sense, it does not really help us if we rank the facet-values based on their probability. By using a flat prior the order of the ranking does not change. Using a more fitting prior distribution may be more useful such that the posterior probabilities predict the actual probabilities as best as possible. To do this we can make use of the Expectation-Maximization algorithm. Expectation-Maximisation is a technique that can learn the parameters of a distribution that includes hidden variables [8], including priors in a hierarchical model.

$$\begin{aligned} \theta_u^{HB} &= \arg\max_{\theta_u} p(\theta_u|\mathcal{X}_u) \\ &= \arg\max_{\theta_u} \frac{p(\mathcal{X}_u|\theta_u)p(\theta_u)}{p(\mathcal{X}_u)} \\ &= \arg\max_{\theta_u} p(\mathcal{X}_u|\theta_u)p(\theta_u) \end{aligned} \tag{3.3}$$

We need to find hyper-parameter θ_0 using $\theta_{1\ldots U}$, given only $\mathcal{X}_u$ as shown in Figure 3.7.

In our new setting we have a distribution with unknown parameters that is generated from another distribution with unknown parameters. In order to find these parameters we will iterate between an expectation step in which we fix θ_0 and calculate the expectation of $\theta_{1\ldots U}$ given θ_0. In the Maximization step we maximize θ_0 and fix the values for $\theta_{1\ldots U}$. After several iterations, the values converge and we can stop. Figure 3.7 shows the model that we want to learn. We refer to this as a hierarchical or multilevel model because we have parameters on (two) different levels.

3.6.1 Implications

Although we are now able to estimate our parameters, including θ_0, which is now part of the model we are estimating, convergence can be slow, especially if our initial guess is wrong. One method to guess our first estimate is to use the Method-of-Moments and and use this as initiallisation for the EM algorithm [**Minka2012**].

3.6.2 Example

If we do this for the `Market` facet, we get a prior α_0, for which each facet-value is different. Figure 3.8 shows how this prior is used when we have not seen any data from a user and how

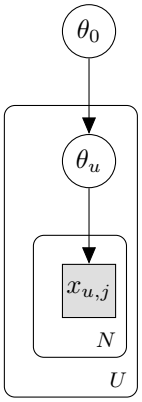


Figure 3.7.: Hierarchical model

this is updated if we have seen 1 and 3 documents from this user, using the same example as before.

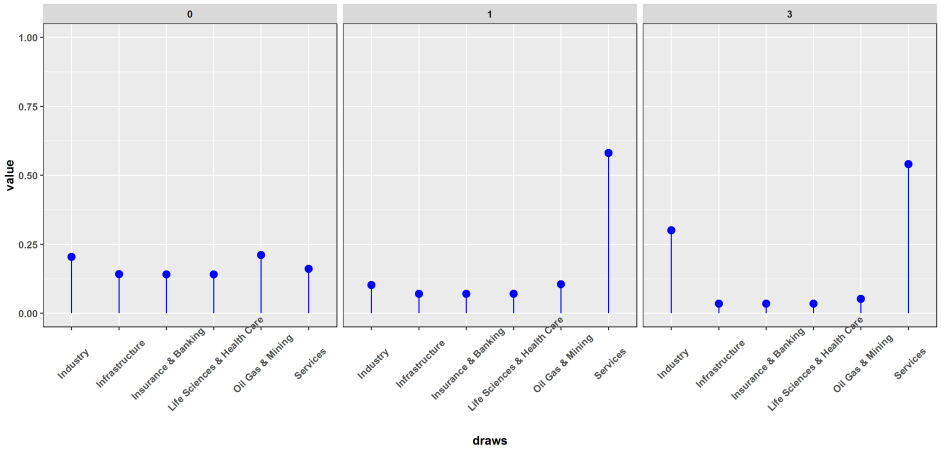


Figure 3.8.: a Maximum A Posteriori user model for the `Market` facet with 0,1 and 3 viewed pages ($\theta_0 = 1$)

In this case the first two facet-values we suggest are still `Services` and `Industry`, but we are now also able to better reason about the other facet-values.

If we observe all the users again we can see that although there is never a user with a 0 or 1 probability of selecting a facet-value, the users are again much more spread out.

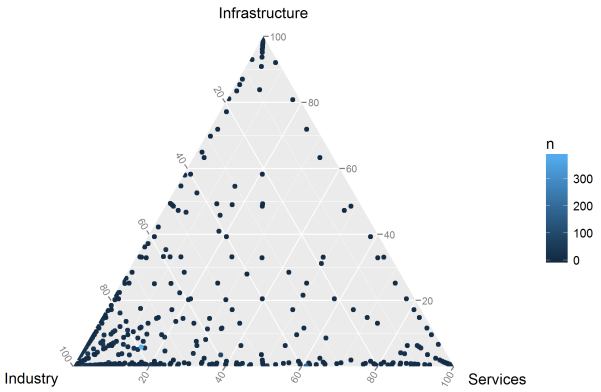


Figure 3.9.: Hierarchical Bayes with Dirichlet prior for 3 of the 6 different Markets

Experimental Setup

4

4.1 Data Gathering

In order to evaluate we will make use of actual visitor logs collected on the Brunel website. Using a JavaScript tracker, we accumulate for each user all the actions the user has taken and save all this data to a ElasticSearch [1] database. From this database we can re-enact every step the user has taken on the website. We gather all vacancies the user has viewed before viewing a vacancy that this user has applied to. The vacancy that a user eventually has applied to will be considered the target document. The facet-values of this document decide whether or not a prediction was correct or not. We do this because although viewing a page can be considered an implicit vote for that document, applying to a job means that a user has explicitly indicated that this document is relevant. An example is shown in Table 4.1. We will use the vacancies before applying, as training-data and the facet-values in the vacancy that the user applies to as target-values that we want to predict using the training-data.

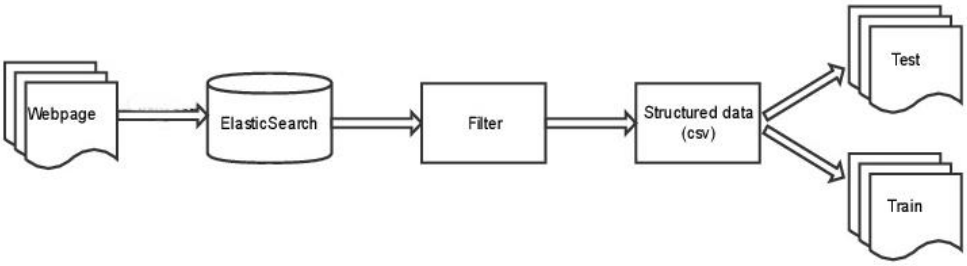


Figure 4.1.: Process of preparing data

In order to prepare the data we need to follow several steps as summarized in Figure 4.1. From the ElasticSearch database we use Apache Spark [2] to process the unstructured logs to structured csv tables. The first step is to remove all non-human visitors to the site. During the initial phase we found that a lot of traffic is coming from scrapers and bots, like Google, Yahoo, Bing and other sites that visited the Brunel website. Since we want to build the models around real users we need to filter out these bots such that the models will be learned on real user data only. Most of these scrapers identified themselves as scraper so could be filtered out easily, others had to be removed using some simple heuristics, such as very high number of pages visited or very low time on spend on each page.

The second step was to rearrange the entries such that the visits of each user are grouped together and sorted on time. This would make it easier to create user models and make sure that when we split the data in a training set and a test set, a user is in only one of those sets. An overview of the complete process of the collecting of the raw clack-data to a cleaned up train and test set is shown in Figure 4.1. This shows very clearly that a lot of visitors only view a low amount of documents showing the importance of making a good

The last step is to produce the actual data tables in a structured way, such that they where easier to analyze. These where split in a test set containing 10,000 users and a training set also containing 10,000 users. Each of these users contained one or more datapoints, the distribution of the amount of datapoints is shown in Figure 4.2

[1] `http://www.elasticsearch.org`
[2] `https://spark.apache.org`

Time	User	Event	Page
14:17:53	User A	Search	
14:17:58	User B	Search	
14:18:01	User C	View	Vacancy 1
14:18:02	User A	View	Vacancy 1
14:18:15	User B	View	Vacancy 2
14:18:19	User A	Apply	Vacancy 1

(a) Click Log Example

Time	User	Event	Page
14:17:53	User A	Search	
14:18:02	User A	View	Vacancy 1
14:18:19	User A	Apply	Vacancy 1
14:17:58	User B	Search	
14:18:15	User B	View	Vacancy 2

(b) Cleaned and Normalized Click Log Example

Table 4.1.: Click log Examples

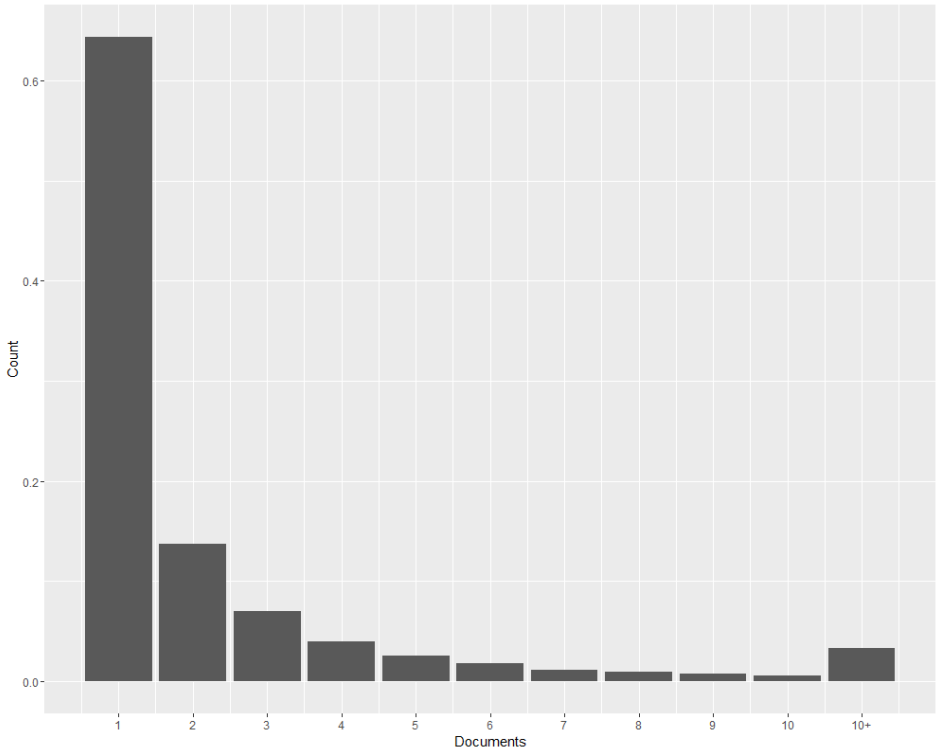


Figure 4.2.: Number of datapoints we have for each user in the total dataset

4.2 Implementation

Using the data produced, we used the R [3] programming language to implement each of the models described in Chapter 3. R was used because it is able to handle a large set of data and able to handle grouped data, such as the case with different users. Moreover, it has the ability to quickly create different models and evaluate them in the same way and make the results reproducible.

4.3 Evaluation Methods

In order to evaluate our different models, we need to define our cost function. A cost function is a function that reflect how good (or how bad) our model is. We will use cost functions suitable for ranking problems that arise in the faceted search problem. We will evaluate the ordering of the facet-values within each facet. We use two different methods to evaluate the different models.

- Mean Reciprocal Rank (MRR)
- Fold@k

For both approaches we have a different scoring function for a facet-value $v_{j,l}$ and the user u. This score is calculated for each facet-value within a facet and ordered in descending order such that we have a ordering $v_{j,l}$ for each $l \in f_j$.

The actual score we give to a facet-value is a 1 if the facet-value is relevant and 0 otherwise. In order to determine which facet-values are actually relevant we use the facet-values of documents that are considered relevant. The relevant documents are the vacancies that the user applied to. Thus for facet-value l of facet j we have:

$$b(v_{j,l}) = \begin{cases} 1 & \text{if } v_{j,l} \text{ is relevant} \\ 0 & \text{if } v_{j,l} \text{ is not relevant} \end{cases} \tag{4.1}$$

Since we have a facet with facet-values $f_j = (v_{j,1}, \ldots, v_{j,L})$. Furthermore we have a function that returns a certain permutation π of f_j. The rank of a facet-value is the number in this permutation this permutation is in.

4.3.1 Mean Reciprocal Rank

The Reciprocal Rank (RR) is a metric that assigns a score for a ranked predictions of a query and is defined as the inverse of the rank of the true value. It is not really a loss function as we would like to maximize this score. The Mean Reciprocal Rank is the mean of all queries by all users of the Reciprocal Rank. The Mean Reciprocal Rank is an often used method for evaluating ranked predictions in Machine Learning and Information Retrieval [25].

The Reciprocal rank is the inverse of the rank of the facet-value. The rank is a function that returns the rank of the facet-value $v_{j,l}$, based on the query $\mathcal{Q}_{u,t}$. The mean of all these queries is the MRR and is what we will use in our evaluations.

$$\begin{aligned} RR_j &= \frac{1}{rank(v_{j,l})} \\ MRR_j &= \frac{1}{U} \sum_{u=1}^{U} \frac{1}{rank(v_{j,l})} \end{aligned} \tag{4.2}$$

The rank is the integer $1 \ldots L_f$ of facet f. Since we have an ordering for each different facet, we can calculate a different MRR for each facet. The MRR is a measure on the interval $(0, 1)$.

[3] `https://www.r-project.org`

4.3.2 Fold@k

In order to find how many facet-values we have to show we came up with a metric called Fold@k, which returns 1 if the target facet-value is within the top k facet-values of the proposed ordering and 0 otherwise. As an example for the `Market` facet with a Fold@k=3, we get a score of 1 if the target document is in one of the top 3 facet-values and 0 otherwise, such as in Table 4.2. In this example we see that if the facet value of the target document appears above the dashed line, which cuts off the top 3 ranked items from the rest, the score is 1, and 0 otherwise.

Oil, Gas & Mining
Industry
Infrastructure
Services
Life Sciences & Health Care
Insurance & Banking

Table 4.2.: A fold at k=3 for the Market Facet

For each user we can take the proportion that falls within this top k facet-values. This can help us decide how many facet-values we need to show. In Equation 4.3 is shown that we need to take the average over all users and use the function b from Equation 4.1 to indicate, whether or not this facet-value is relevant. This shows the average proportion of relevant facet-values in the first k results of the permutation.

$$Fold@k = \frac{1}{U} \sum_{u=1}^{U} \sum_{1}^{k} b(v_{j,l}) \tag{4.3}$$

For different k and different facets we can calculate this $Fold@k$, which can help in deciding how many facet-values we need to show for each facet and what the estimated error is. Naturally, when k is larger than the amount of facet-values, the Fold@k will be 1 as we will be able to show all possible facet-values. Lower k will result in a lower score.

This metric is not to be confused with $precision@k$, which measures the total number of correct items within the first k items. In $precision@k$ we measure the amount of correct targets in the top k items compared to all correct target items. Since we will only target one document, there is only one relevant facet-value, which makes $precision@k$ less suitable for this problem as all other facet-values above the fold are irrelevant. For $k = 1$ this method behaves as the winner-takes-all metric [15].

Evaluation & Results

5

In this chapter we will show the results of the experiments based on the evaluation methods described in Chapter 5. These evaluations will lead to the answers to the research questions in Chapter 6.

5.1 Experimental Results

RQ1: How does adding a prior affect the performance of a personalized faceted search system?

In order to answer this question we will calculate the mean reciprocal rank (MRR) for each of the methods on all facets. The MRR is a function of the rank of the facet-values and is not influenced directly by the predicted probability of a user selecting that facet-value. This causes that even though the actual probabilities of the M^{MAP} model will be closer to the actual probabilities than the M^{Count} model, the ranking will stay in exactly the same order, therefore resulting in exactly the same MRR for bot M^{ML} and M^{MAP}.

In Table 5.1 we see for all facets that we can easily outperform the M^{Count} model. However in many cases the difference between M^{ML} and M^{HB} is rather small and is not statistically significant.

Facet	M^{Count}	M^{ML}	M^{HB}
areaofexpertise	0.08978578	**0.4768223**	**0.4736153**
branch	0.1083673	**0.590427**	**0.5877668**
continent	0.2270764	**0.901049**	**0.9027521**
country	0.09279324	**0.8024491**	**0.8046957**
educationlevel	0.3351209	0.6030912	**0.6568793**
experience	0.4890349	0.5254844	**0.6568793**
hoursperweek	0.4370968	**0.8410353**	**0.8411799**
market	0.591534	0.8559484	**0.8722634**
region	0.09376599	0.4878599	**0.6350919**
worklevel	0.4385104	0.7827348	**0.8079415**

Table 5.1.: Mean Reciprocal Rank for different models

In this table we can see the scores for several facets, with the best model in bold. When the

In other cases adding a prior does increase the MRR a lot more. In all cases, the M^{HB} never perform worse than the other models, thus confirming that adding a prior based on the Expectation-Maximization algorithm indeed proves to be a good choice.

RQ2: How does the amount of data we have of a user affect the performance of personalized faceted search?

In order to see how the amount of data affects the performance of the different models we plot for different models the score of our scoring function. In Figure 5.1 we see an example of the Fold@$k = 5$ score when we increase the amount of user data, shown on the x-axis for the `Expertise` facet. This means that after we have seen 3 vacancies from a user and we show the user a suggestion for 5 facet-values, we are right about 80 percent of the time when we use the M^{HB} model.

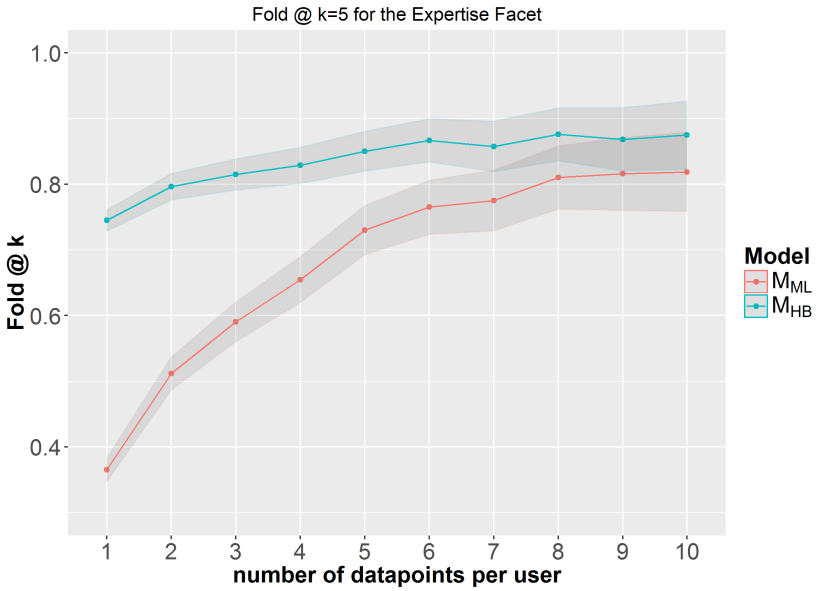


Figure 5.1.: Fold@k=5 for `Expertise` Facet

Another important observation is that the performance of the M^{HB} is considerable better than the M^{ML} model, when we don't have a lot of data from a user. However this difference between the models get smaller when we have more data available. This is what we would expect as when we get more and more data from a user the prior will have less effect and the two models converge to each other.

This is what we would expect as the prior of the HB model get relatively less weight as we increase the number of datapoints. For the other facets we see a similar observation. the first few datapoints improve the prediction greatly but it flattens out later on as we have get more data from each user.

Here we also see another advantage of these kind of models. The M^{Count} model, not included in this Figure will stay constant no matter how many previous documents we have seen before.

RQ3: What is the error we make when we only show a fraction of the facet-values?

We can show only a fraction of the facet-values for a certain facet by showing only the top-k facet-values and calculate how often the target facet-value is within these values and how often not. In Figure 5.2 we use the M^{HB} model to plot the Fold@k for $k = 1, 3, 5$ and 10 and see how they improve when we increase the amount of datapoints per user.

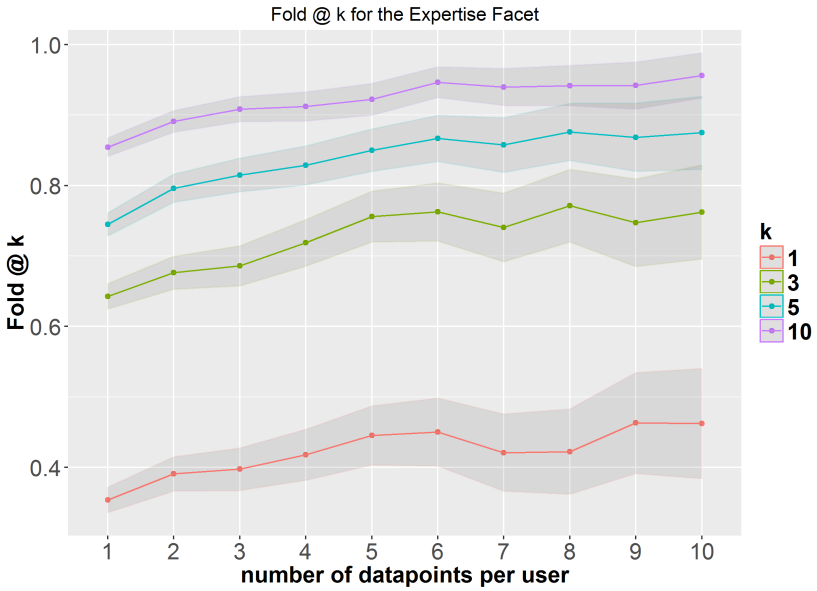


Figure 5.2.: Fold@k for `Expertise` Facet

When we increase the amount of facet-values we increase the Fold@k score. One can make the consideration of only showing the top 5 to be right between $80\% - 90\%$ of the time as soon as we have seen at least 2 views form that user or show 10 (out of 44 possible values) to get at least 90% accuracy. A small drop can be observed at 7 datapoints per user, but this can be explained by some noise and the limited data we have of users having at least this amount of datapoints.

Conclusions

6

In this chapter we will conclude our work and answer the research questions that we introduced in Chapter 1. Furthermore we will discuss future work that could follow from this thesis and discuss the results of this thesis.

6.1 Research Questions

RQ1: How does adding a prior affect the performance of a personalized faceted search system?

As we have seen in Section 5.1 adding different priors can affect the performance of the faceted search system. A flat prior, although it can make more sense to the probability of being chosen, does not add anything compared to the Maximum Likelihood model. However optimizing a prior based on the Expectation Maximization algorithm significantly improves the performance.

RQ2: How does the amount of data we have of a user affect the performance of personalized faceted search?

Using a hierarchical model improves the performance of a personalized faceted search interface, however the improvement diminishes with respect to the simpler Maximum Likelihood model after we know more about the user. Using a hierarchical model can therefore improve the performance of a faceted search interface for new users, which are most of the users in many cases.

RQ3: What is the error we make when we only show a fraction of the facet-values?

The error we make with only showing a fraction of the facet-values is highly dependent on the facet. Some facets have a lot of facet-values which make it harder to predict accurately, while other facets may not have that many facet-values, but are still harder to predict. One reason might be that users do not really know which facet-value they are interested in or the facet-value might not matter much to the user. A user might be interested in multiple facet-values within a certain facet.

MQ: How can we improve personalized faceted search when we do not have much data from a user?

Based on the Results in Chapter 5 we can see that for this dataset we can learn a lot from the first few views of a user to determine what this user is looking for. Although we do get better estimates when we get more data from the user, most improvement comes from the first few views.

When we don't have any data we can use prior data from other users. This two level hierarchical model can therefore combine both content-based models and collaborative filtering models in one framework. Based on the amount of data we have of a user we start with a user model that uses other users' data, but gradually develop each individual user model as we gain more information about that user.

6.2 Future Work

In this work we mainly looked at categorical values and used categorical distributions to represent these values. How this can be extended to numerical values, such that user can select ranges, should still be researched. The documents in this dataset also had only one facet-value per facet. An extension could be made to allow for multiple facet-values per facet.

In the dataset we used here all the entries in the Dirichlet prior distribution where lower than 1. This does not necessarily have to be the case for other datasets. The fact that the prior behaves this way is an indication that there is not much cohesion between the different values within each facets. Users are looking for specific documents and do not look for a broad range of documents. This can also be seen by the fact that once the first document has been seen the user model does not improve a lot and the first document seen is already a very good indication of what the user is looking for. For other datasets such as movies where people generally watch a broad range of documents, the prior will probably be very different.

One of the limitations in the collection of the dataset was that we could retrieve the documents viewed by a user from a search pages but can not accurately know which documents did show up in the search results, but on which the user did not click. This can be valuable information and may improve the suggestions done for the faceted search system.

6.3 Discussion

We saw that we could predict facet-values of some of the facets within 80%-90% accuracy, by only showing the top 3 or top 5 facet-values. Some other facets are more more difficult to predict. Not only due to the fact that there are many more facet-values, but users also seemingly change more between these facet-values.

There are currently not many public datasets to compare our results with. The movies dataset that was used in several other papers, used explicit feedback to assess whether a movie was relevant and thus used positive and negative feedback, the data we collected only used implicit feedback.

This work could help implementing a better faceted search engine where people are looking for specific documents. It shows how to evaluate these in some specific setting, namely how to evaluate when we only show a few of the facet-values and how to show the performance when we do not know much about the user yet.

Hierarchical Bayes

A

A.1 Beta-Binomial Model

In the case that the facet-values are Categorical, we can model these values as draws from a Binomial distribution. This is useful for facet-values that have more than one option at the same time within a facet type or for which a facet-value is optional. The probability that a facet-value is is relevant to a user is a can be modeled by a Bernoulli distribution. This is based on the previous seen documents of which k out of n documents had this facet-value, which can be modeled as a Binomial distribution. Both the Bernoulli and the Binomial distribution have a Beta conjugate prior.

$$\begin{aligned} \theta_{1...U} &\sim Beta(\alpha, \beta) \\ x_{u,1}, \ldots, x_{u,N} &\sim Bern(\theta_u) \end{aligned} \tag{A.1}$$

In order to estimate the parameters of this model we start with an initial initialisation of the α and β parameters of the Beta distribution for θ_0, for example, using the method-of-moments. Then we iterate between an E-step and an M-step. In the E-step we estimate the parameters θ_u for each user in U given θ_0, which is simply the MAP estimate from equation 3.2 in section 3.5. In the M-step we maximize θ_0 given all the values for θ_u. [18]

$$\theta_u = \frac{\theta_\alpha + |x_{k,u}|}{\theta_\alpha + \theta_\beta + |x_u|} \tag{A.2}$$

$$\begin{aligned} \psi(\theta_\alpha) - \psi(\theta_\alpha + \theta_\beta) &= \frac{1}{N} \sum_i \psi(\theta_\alpha + |x_{k,u}|) - \psi(\theta_\alpha + \theta_\beta + |x_u|) \\ \psi(\theta_\beta) - \psi(\theta_\alpha + \theta_\beta) &= \frac{1}{N} \sum_i \psi(\theta_\beta + |x_u| - |x_{k,u}|) - \psi(\theta_\alpha + \theta_\beta + |x_u|) \end{aligned} \tag{A.3}$$

We can use approximate this using the old estimate for θ_α and θ_β:

$$\begin{aligned} \psi(\theta_\alpha) &\approx \frac{1}{N} \sum_i \psi(\theta_\alpha + |x_{k,u}|) - \psi(\theta_\alpha + \theta_\beta + |x_u|) - \psi(\theta_\alpha^{old} + \theta_\beta^{old}) \\ \psi(\theta_\beta) &\approx \frac{1}{N} \sum_i \psi(\theta_\beta + |x_u| - |x_{k,u}|) - \psi(\theta_\alpha + \theta_\beta + |x_u|) - \psi(\theta_\alpha^{old} + \theta_\beta^{old}) \end{aligned} \tag{A.4}$$

To solve for θ_α and θ_β we need to invert the gamma function, which can be approximated using a Newton-Raphson method and using the first and second derivative of the gamma function.

$$\theta^{new} = \theta^{old} - \frac{\psi(\theta^{old}) - y}{\psi'(\theta^{old})} \tag{A.5}$$

A.2 Dirichlet-Multinomial Model

If we have Categorical facet-values of which a value in mandatory and the facet-values are mutually exclusive we can model all facet-values within the Facet as a Categorical distribution, which is a generalization of the Bernoulli distribution with more than two

option. The generalization of the Binomial distribution is the Multinomial distribution which has a Dirichlet conjugate prior.

$$\begin{aligned} \theta_1, \ldots, \theta_U &\sim Dir(\alpha_1, \ldots, \alpha_k) \\ X_1, \ldots, \mathcal{X}_N &\sim Cat(\theta_u) \end{aligned} \tag{A.6}$$

$$\begin{aligned} \theta_u &= \frac{\theta_\alpha + |x_{k,u}|}{\theta_0 + |x_u|} \\ \theta_0 &= \sum_1^K \theta_k \end{aligned} \tag{A.7}$$

θ_0 is often referred to as the concentration parameter. By dividing the θ_u with this concentration parameter we get the probabilities of each of the facet-values.

$$\begin{aligned} \psi(\alpha_k) - \psi(\theta_0) &= \frac{1}{N} \sum_i \psi(\theta_\alpha + |x_{k,u}|) - \psi(\theta_0 + |x_u|) \\ \psi(\theta_\alpha) &\approx \frac{1}{N} \sum_i \psi(\theta_\alpha + |x_{k,u}|) - \psi(\theta_0 + |x_u|) - \psi(\theta_0^{old}) \end{aligned} \tag{A.8}$$

A.3 Initialization

The Expectation Maximization method can take quite a few iteration before it converges. A good initial approximation can help reduce the amount of iterations needed. We have used the Method-of-Moment to initialize the EM algorithm as described in the previous section.

Method-of-Moments for the Beta-distribution is defined as:

$$\begin{aligned} \hat{\alpha} &= \bar{x} \left(\frac{\bar{x}(1-\bar{x})}{\bar{\sigma}^2} - 1 \right) \\ \hat{\beta} &= (1-\bar{x}) \left(\frac{\bar{x}(1-\bar{x})}{\bar{\sigma}^2} - 1 \right) \end{aligned} \tag{A.9}$$

The mean and variance of the Beta distribution can be calculated with:

$$\begin{aligned} \mu &= \frac{\alpha}{\alpha + \beta} \\ \sigma^2 &= \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \end{aligned} \tag{A.10}$$

So if we know the sample mean and sample variance of the data we can try to fit it and solve it for α and β.

$$\begin{aligned} \alpha &= -\frac{\mu(\sigma^2 + \mu^2 - \mu)}{\sigma^2} \\ \beta &= \frac{(\sigma^2 + \mu^2 - \mu)(\mu - 1)}{\sigma^2} \end{aligned} \tag{A.11}$$

The Dirichlet distribution has no direct closed form solution to the method-of-moments, but can be estimated [18]. In our solution we estimated each θ_i with choosing α for θ_i and β for $\alpha_0 - \alpha_i$ and repeat this for all $\alpha_i \in \alpha$.

Datasets

B

In this thesis I will focus on a dataset collected from Brunel [1]. This dataset contains vacancies and which users have viewed them.

B.1 Vacancies

The vacancy dataset is collected over a period of 6 weeks (2015-06-20 - 2015-07-31) by tracking visitors on the Brunel website. A total of 8.624 different vacancies where viewed by approximately 152.360 different visitors. Each of these vacancies could have one of the facets with corresponding facet-values as shown in Table B.1. In each of the following tables an overview is given of the facet-values within the Nominal and Ordinal facet-value types.

[1] http://www.brunel.nl

Facet	Type	#Facet-Values	Range
Area of Expertise	Nominal	44	
Branch	Nominal	40	
Continent	Nominal	6	
Country	Nominal	30	
Education Level	Ordinal	9	
Experience	Ordinal	5	
Hours per Week	Interval		0-88
Language	Nominal	8	
Market	Nominal	6	
Name	Free-Text	6708	
Region	Nominal	59	
Worklevel	Nominal	4	
Worklocation	Free-Text	591	

Table B.1.: Vacancy Facets

Market	Count
Industry	5551
Services	1532
Infrastructure	661
Oil Gas & Mining	381
Insurance & Banking	268
Life Sciences & Health Care	226

Table B.2.: Market

Branch	Count
Automotive	1849
IT & Telecom	1240
Machine & Equipment Construction	973
Electronics	821
Production & Manufacturing	454
Industrial Services	375
Building & Construction	364
High Tech	361
Oil & Gas	314
Energy Power Plants	267
Aerospace	170
Sales & Marketing	156
Banking	148
Building Facilities & HVAC	123
Finance	103
(Petro)Chemical	101
Shipbuilding	96
Pharmaceutical	73
Medical devices	67
Mining	67
Pension & Income	63
Rail	55
Logistics	46
Biotechnology	39
Defense Systems	38
Public & Civil	36
Traffic City & Urban Planning	36
Property & Casualty	34
Utilities & Distribution	33
Healthcare	30
Mortgage	22
Legal	21
Water & Waste Management	14
Research Facilities	8
Diagnostics	6
Care & Welfare	5
Non-Profit	4
Government	3
Institutional Healthcare	3
Health	1

Table B.3.: Branch

Area of Expertise	Count
Engineering & Design	2283
Project Management & Services	1436
Construction & Commissioning	762
Operations & Maintenance	624
Embedded Software	567
Software Development & Application Management	522
Research & Development	492
Health Safety Environmental & Quality	225
Network Systems Telephony & Hardware	193
Front Office & Sales	161
Project Management Consultancy & Auditing	144
Cost Control & Procurement	141
Testing	138
Sales	123
Support & Service Desk	116
Back Office & Administration	89
Functional & Business Analysis	85
Planning & Control	73
Architecture & Design	60
Database Management	60
Construction Commissioning & Operations	42
Drilling Completions & Geosciences	37
Medical	35
Regulatory	22
Financial Administration	20
Risk Management	19
Marketing	17
Trade	17
Clinical	13
Operations	13
Project Management & Coordination	12
Civil Law	10
HR Finance & Support	10
Safety	10
Public Law	8
Communications	7
Study & Science	7
Technical	5
Risk & Product Management	4
Exploration & Geosciences	3
HR & Support	3
Other Law	3
Mill & Metallurgy	2
Quality Audit & Quality Control	1

Table B.4.: Area of Expertise

Continent	Count
Europe	8295
North America	157
Asia	109
Australia	38
South America	19
Africa	5

Table B.5.: Continent

Country	Count
Germany	5733
Netherlands	1930
Belgium	386
Austria	131
Canada	115
Switzerland	42
United States	41
Qatar	39
Australia	38
United Kingdom	33
United Arab Emirates	30
Czech Republic	23
Brazil	19
Saudi Arabia	10
Denmark	9
India	9
Norway	7
Russian Federation	7
Angola	3
China	3
Iraq	2
Japan	2
Korea Republic of	2
Kuwait	2
Oman	2
Algeria	1
Honduras	1
Malaysia	1
Monaco	1
South Africa	1

Table B.6.: Country

Education Level	Count
Professional Bachelor	3790
Academic Bachelor	1943
Other	1265
Academic Master	730
Professional Master	266
Vocational School	230
Professional Courses	216
Secondary School	104
Academic Master +	77

Table B.7.: Education Level

Years of Experience	Count
2 - 4 years	1681
4 - 6 years	1060
0 - 2 years	842
6 - 9 years	255
more than 10 years	167

Table B.8.: Experience

Worklevel	Count
Specialists & experts	6619
Staff & support	1193
Middle (operational) management	733
Top (strategic) management	70

Table B.9.: Worklevel

Bibliography

[1] Ori Ben-Yitzhak, Sivan Yogev, Nadav Golbandi, et al. “Beyond basic faceted search”. In: *Proceedings of the international conference on Web search and web data mining - WSDM ’08* (2008), pp. 33–44 (cit. on p. 7).

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. “Latent Dirichlet Allocation”. In: *Journal of Machine Learning Research* 3.4-5 (2003), pp. 993–1022. arXiv: 1111.6189v1 (cit. on p. 8).

[3] Peter Brusilovsky, A Kobsa, and W Nejdl. *The adaptive web: methods and strategies of web personalization*. 2007 (cit. on pp. 2, 7).

[4] Wray L. Buntine. “Operations for learning with graphical models”. In: *JAIR* 2 (1994), pp. 159–225 (cit. on p. 10).

[5] Robin Burke. “Hybrid recommender systems: Survey and experiments”. In: *User modeling and user-adapted interaction* (2002) (cit. on p. 3).

[6] Ilknur Celik, Fabian Abel, and Patrick Siehndel. “Towards a framework for adaptive faceted search on twitter”. In: *CEUR Workshop Proceedings*. Vol. 823. 2011, pp. 11–22 (cit. on p. 8).

[7] Li Chen and Pearl Pu. “Critiquing-based recommenders: survey and emerging trends”. In: *User Modeling and User-Adapted Interaction* 22.1-2 (Oct. 2011), pp. 125–150 (cit. on p. 7).

[8] Arthur P Dempster, Nan M Laird, and Donald B Rubin. “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the royal statistical society. Series B (methodological)* (1977), pp. 1–38 (cit. on p. 14).

[9] Florent Garcin, Kai Zhou, Boi Faltings, and Vincent Schickel. “Personalized News Recommendation Based on Collaborative Filtering”. In: *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology* (Dec. 2012), pp. 437–441 (cit. on p. 8).

[10] Florent Garcin, Christos Dimitrakakis, and Boi Faltings. “Personalized news recommendation with context trees”. In: *RecSys ’13 Proceedings of the 7th ACM conference on Recommender systems* V.January (2013), pp. 105–112. arXiv: arXiv:1303.0665v1 (cit. on p. 8).

[11] a Gelman and J Hill. “Data analysis using regression and multilevel/hierarchical models”. In: *Policy Analysis* (2007), pp. 1–651 (cit. on p. 8).

[12] Ralf Herbrich, Tom Minka, and Thore Graepel. “Trueskill?: A Bayesian skill rating system”. In: *Advances in Neural Information Processing Systems*. 2006, pp. 569–576 (cit. on p. 8).

[13] Yifan Hu, Yehuda Koren, and Chris Volinsky. “Collaborative Filtering for Implicit Feedback Datasets”. In: *2008 Eighth IEEE International Conference on Data Mining* (Dec. 2008), pp. 263–272 (cit. on p. 3).

[14] Jonathan Koren, Yi Zhang, and Xue Liu. “Personalized interactive faceted search”. In: *Proceeding of the 17th international conference on World Wide Web - WWW ’08* (2008), pp. 477–485 (cit. on pp. 4, 7).

[15] Quoc Le and Alexander Smola. “Direct Optimization of Ranking Measures”. In: *arXiv preprint arXiv:0704.3359* 1.2999 (2007), pp. 1–29. arXiv: 0704.3359 (cit. on p. 20).

[16] Fabiana Lorenzi and Francesco Ricci. “Case-based recommender systems: a unifying view”. In: *Intelligent Techniques for Web Personalization* (2005), pp. 89–113 (cit. on p. 7).

[17] L. McGinty and Barry Smyth. “On the role of diversity in conversational recommender systems”. In: *Case-Based Reasoning Research and Development* (2003), pp. 276–290 (cit. on p. 7).

[18] Thomas Minka. *Estimating a Dirichlet distribution*. 2000 (cit. on pp. 27, 28).

[19] Michael J Pazzani and Daniel Billsus. “Content-based recommendation systems”. In: *The adaptive web*. Springer, 2007, pp. 325–341 (cit. on p. 7).

[20] Liana Razmerita. “Ontology-based user modeling”. In: *Ontologies*. Springer, 2007, pp. 635–664 (cit. on p. 6).

[21] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. “Collaborative filtering recommender systems”. In: *The adaptive web*. Springer, 2007, pp. 291–324 (cit. on p. 7).

[22] Vineet Sinha and David R Karger. “Magnet: Supporting navigation in semistructured data environments”. In: *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM. 2005, pp. 97–106 (cit. on p. 1).

[23] Daniel Tunkelang. “Faceted Search”. In: *Synthesis Lectures on Information Concepts, Retrieval, and Services* 1.1 (Jan. 2009), pp. 1–80 (cit. on p. 7).

[24] Paolo Viappiani, Pearl Pu, and Boi Faltings. “Conversational recommenders with adaptive suggestions”. In: *Proceedings of the 2007 ACM conference on Recommender systems - RecSys ’07* (2007), p. 8 (cit. on p. 7).

[25] Ellen M Voorhees et al. “The TREC-8 Question Answering Track Report.” In: *Trec*. Vol. 99. 1999, pp. 77–82 (cit. on p. 19).

[26] Martin J Wainwright and Michael I Jordan. “Graphical models, exponential families, and variational inference”. In: *Foundations and Trends® in Machine Learning* 1.1-2 (2008), pp. 1–305 (cit. on p. 10).

[27] Yi Zhang and Jonathan Koren. “Efficient bayesian hierarchical user modeling for recommendation system”. In: *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (2007), pp. 47–54 (cit. on p. 8).

List of Figures

List of Tables