University of Strathclyde

Department of Computer and Information Sciences

Bayesian Latent Variable Models for the Collaborative Web

Morgan A. Harvey

A thesis presented in fulfilment of the requirements for the degree of

Doctor of Philosophy at the University of Strathclyde

May 2011

Dedicated to the memory of William Boyd Steele

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed: Date:

Acknowledgements

It is often said that no one completes their PhD alone and this statement could not be more true than in this case. Over the course of the past three and a half years I have received help, guidance and support from a large number of people to whom I am immensely grateful. I have had the opportunity to work in a friendly and inspiring research environment and have also had the opportunity to travel and work with colleagues abroad. The people I have met and worked with have had a very positive influence on my research and I feel privileged to have had the chance to work with them.

Firstly, I would like to thank my supervisor, Ian Ruthven. He has been a constant source of encouragement and support throughout and has always made time in his increasingly busy schedule to speak with me about my research when I needed it. I'd also like to thank him for not only being immensely helpful from a technical standpoint but for also being there to have a joke with about things and for guiding me through this often difficult process. I also thank Crawford Revie and Mark Dunlop for their insightful comments and suggestions.

I would like to especially mention three people who have been a constant source of inspiration to me, not to mention being great friends and colleagues. They were always there to lend a friendly ear and I have enjoyed working with all three of them. Firstly, Mark Baillie who sparked my interest in statistics and helped me in the early days of my PhD. Secondly, Mark Carman with whom I have had the pleasure of working on a number of papers and who invited me to come to Lugano, an experience which I will always treasure. Lastly, but certainly not least, Dave Elsweiler. Thanks for all your help at the start of my PhD, I have learned so much from you.

I'd like to thank all of my colleagues and friends in the department who were always there to help and put up with my series of talks involving probability theory

and statistics! I'd like to especially thank the people I shared an office with: Fabio S., Michael, Kostas and Robert, without whom it would have been a very boring three and a half years. I'd also like to thank Fabio Crestani for all of his help and encouragement, I look forward to working with you again in the future. At the very least I'd also like to thank my fellow iLabers; Masnizah, Andreas and Laura and and also everyone else from the CIS department. Thanks also to my friends outside of the world of research; Ian, Gavin, Malcolm, Gary and Duncan. I'd especially like to thank Martin, our long talks in the flat and in the pub really helped me to get a better understanding of what I was doing and formed a lot of the ideas in this thesis. My sincerest apologies to anyone I've forgotten to mention.

Finally, I'd like to thank my parents, Pam and Gus, and my extended family. Their encouragement, support and love have made this possible. Thank you all.

Abstract

Since its creation in the early 1990s the Web has held the promise of allowing near-instantaneous communication, participation and sharing of resources and ideas between users across the globe. However up until fairly recently, the Web was predominantly a large collection of static documents providing no real scope for such interaction. The past decade has seen the arrival and rapid growth of the so called "Web 2.0" movement where sites have become increasingly more social with users able to share information with others. Due to the sheer volume of information available on the Internet and the massive number of products available on online shops, finding items which may be of interest can often be a very difficult task. Furthermore the continual expansion of the Web makes it impossible to manually evaluate each new item to determine if it might be of interest. In recent years the emergence of a more social web has resulted in the development of tools with the purpose of making this undertaking both easier and more enjoyable.

This thesis explores two avenues of this new social web: social tagging and ratings-based collaborative filtering. Both of these methods rely on the users of the system to provide some information about the resources contained therein and then use this information to improve the user experience. The main hypothesis of this thesis is that these new social tools can be significantly improved by the use of machine learning methods to model and make sense of the data available. The work introduces a family of novel latent variable Bayesian models designed specifically to deal with this sparse and extremely noisy data. A series of experiments carried out on real-world data sets show that these models can overcome the inherent difficulties and provide significant improvements in performance over state of the art systems. Furthermore it is shown that the output of these models is more readily interpretable than from competing models and can therefore be utilised to gain a more complete understanding of the complex social and topical dynamics of such systems.

Contents

List of Figures

List of Tables

Publications and Contributions

The following thesis and any work presented therein are my own and are the result of my own original research. Throughout the PhD process I have consulted and collaborated with my supervisor, Ian Ruthven, who was involved in discussion of how the work should unfold and who also assisted in the writing of the resulting publications. I have also collaborated with Mark Baillie, Mark Carman and David Elsweiler on various parts of this work, however in all cases the core work was my own and I was the main contributor, investigator and experimenter. Some of the work contained within this thesis has been published in the following:

- [HBRE09] **Harvey, M.**, Baillie, M., Ruthven, I., Elsweiler, D. Folksonomic Tag Clouds as an Aid to Content Indexing.
  2nd Annual Workshop on Search in Social Media (SSM 2009), SIGIR 2009, Boston, Massachussetts. (July 2009)

- [HBCR10] **Harvey, M.**, Baillie, M., Ruthven, I., Carman, M. Tripartite Hidden Topic Models for Personalised Tag Suggestion
  Advances in Information Retrieval, 32nd European Conference on IR Research, ECIR 2010. Milton Keynes, UK. (March 2010)

- [HRC10] **Harvey, M.**, Carman, M., Ruthven, I. Ranking Social Bookmarks Using Topic Models
  ACM 19th Conference on Information and Knowledge Management, CIKM 2010. Toronto, Canada. (October 2010)

- [HRC11] **Harvey, M.**, Carman, M., Ruthven, I.
  Improving Social Bookmark Search Using Personalised Latent Variable

Language Models Fourth ACM International Conference on Web Search and Data Mining, WSDM2011. Hong Kong, China. (February 2011)

- [HRCC11] **Harvey, M.**, Carman, M., Ruthven, I., Crestani, F. Bayesian Latent Variable Models for Collaborative Item Rating Prediction ACM 20th Conference on Information and Knowledge Management, CIKM 2011. Glasgow, Scotland. (October 2011)

Chapter 1

Introduction and Background

> "If you have an apple and I have an apple and we exchange these apples then you and I will still each have one apple. But if you have an idea and I have an idea and we exchange these ideas, then each of us will have two ideas."
>
> *George Bernard Shaw*

Humans are inherently social beings and much of the scientific and cultural progress made in history has been as a direct result of improvements to communication technology. From the early invention of basic language to the printing press and the telegraph, new technologies allowing people to share ideas and knowledge have been at the very heart of society. Since its invention in the early 1990s, the Web has held the promise of allowing near-instantaneous communication, participation and sharing of resources and ideas between users across the globe [BLCL^{+}94]. However up until fairly recently, the Web was predominantly a large collection of static documents providing no real scope for such interaction. The past decade has seen the arrival and rapid growth of the so-called "Web 2.0" movement where sites have become increasingly more social with users sharing their information with others.

Due to the sheer volume of information available on the Internet and the massive number of products available on online shops, finding items which may be of interest can often be a very difficult task. The continual expansion of the Web makes it impossible to manually evaluate each new item to determine if it might be of interest. Furthermore media items such as photographs, music and movies are extremely difficult to automatically classify and annotate and therefore it is often necessary to rely on human beings to provide their descriptions. In recent years the emergence of a more social web has allowed for the development of tools with the expressed purpose of making this both easier and more enjoyable. This thesis specifically explores two avenues of this new social web: social tagging and ratings-based collaborative filtering. Both of these methods rely on the users of the system to provide some information about the resources contained therein and then make use of this information to improve the user experience. The main contributions of the thesis are the development of a series of novel Bayesian latent variable models appropriate to these settings and the evaluation of these models via three different experiments on real-world data. The thesis is structured as follows:

- Chapter 1 presents a background overview of research into both social tagging and collaborative filtering, including some discussion of findings from this work. The aims of this chapter are to instruct the reader on basic theories and approaches to understanding the complex dynamics of such systems and to instruct and inform the approaches and methods investigated later in the thesis. The chapter discusses the posited benefits of these new paradigms, whether such systems are necessary and where they can be most effectively utilised. It then proceeds to analyse some common motivations identified for these social interactions and looks into the statistical structure of large folksonomies in order to better understand how they evolve and grow. In doing so we gain a more complete understanding of how social tagging systems are used in practice and how they are organised and structured.

- Chapter 2 reviews the existing literature in modelling socially generated data including details of previous attempts to solve the problem of personalising tag suggestion, search and collaborative filtering.

- Chapter 3 provides a short introduction to the Bayesian methods of statistical modelling - in particular latent variable models - and how they can be used as powerful tools for modelling socially generated data. Latent topic models are then introduced as a stepping-stone to explaining a series of novel Bayesian models (TTM1, TTM2, LITRM1 and LITRM2) designed for social data, which are used later in the experiments.

- Chapters 4, 5 and 6 show by experiment on real-world data that not only can the models developed in Chapter 3 be used to perform useful tasks on socially generated data but that they are also able to out-perform other state-of-the-art methods in the field.
- In Chapter 7 concludes the thesis by summarising its main ideas and themes and point to directions of possible future work.
- Appendices A and B discuss the two key distributions used in this thesis: the multinomial and Dirichlet, and mathematically derives the Tagging Topic Model and its related Gibbs sampling algorithm.

1.1 Social Tagging

Social tagging or social annotation refers to an increasingly popular method of data categorisation found, in some form or another, on many of these Web 2.0 sites. This new paradigm allows users of a system to define their own personal set of categorisations in order to organise and publicly annotate a diverse range of resources in a manner which is meaningful to them [GH05]. In such systems lightweight keywords (otherwise known as “tags”) are assigned to resources by users in a shared, usually on-line, environment. The resulting assemblage of tags from all users covering all resources form a “folksonomy,” a conflation of the words ’taxonomy’ and ’folk’; literally an organisation scheme of the people. This is in direct contrast with the much more conventional approach of providing users with a finite set of available categorisations, defined *a-priori* by the designers of the system or by information architects. This somewhat ad-hoc categorisation system is known as a bottom-up approach since the base elements of the system (the tags) are defined first in great detail and are linked together in such a way that higher level associations may be derived from them.

The rise of social tagging has caused a major resurgence of interest in manual indexing on the Web [Voß07], it is frequently posited as being one of the defining features of the Web 2.0 revolution and an integral part of the emerging social web. In this setting tags can be seen as being a form of metadata - literally "data about data". Social tagging systems have been used to organise, classify and share an extremely diverse range of topics and content types including (but certainly not limited to): URLs, photographs, academic papers, video clips, products for purchase and even music. Popular examples of online social tagging systems include Flickr (`http://flickr.com/`), delicious (`http://delicious.com/`), Last.fm (`http://last.fm/`) and BibSonomy (`http://bibsonomy.org/`), however examples of social classification can be found in a large number of other less obvious, and more professional, applications. For example the University of Pennsylvania Library allows users to tags records, supplementing the more traditional library subject indexes [BP06].

Despite its popularity in recent years, research into social tagging is so far relatively sparse and while there has been some early seminal work in the area, there is much still left to be done. The current literature draws from a number of related topics in the disciplines of Computer Science, Information Science and the Social Sciences. In Computer and Information science the fields of information retrieval, collaborative filtering, graph theory, data mining and social networking are of particular relevance. The majority of the existing literature attempts to superficially analyse the high-level structure of these systems in terms of descriptive statistics or looks at the motivations people have for annotating resources and being part of the community.

The data structure of a folksonomy consists of 3 different entities; resources, users and the tags themselves as well as the links between these entities resulting from the annotation of a resource by a user.

Users are responsible for assigning tags to resources in social tagging systems. Users will generally only annotate a resource he or she is interested in and therefore a user tagging a resource can be seen as a preference or vote for that resource by the user.

Resources are the individual objects which are tagged in the system and vary in their type depending on the system in question. They can also be referred to as objects or - in an information retrieval setting - documents. All of the tags used to annotate a resource, conflated over all users, is its description. In many cases the representation of the resource in the system is simply a unique ID number.

Tags are the free-form keywords chosen by users to annotate resources and are usually single-term words or short compound phrases. The complete set of unique tags within a given social tagging system is referred to as its vocabulary or lexicon.

If we view the structure formed by these entities as a large graph, then we can use the links obtained from tag co-occurence as an indicator of relationships between resources. More formally one can model a folksonomy as a tripartite graph with 3 disjoint sets of nodes: resources $\mathcal{D} = \{\mathtt{d}_1, \ldots, \mathtt{d}_D\}$, users $\mathcal{U} = \{\mathtt{u}_1, \ldots, \mathtt{u}_U\}$ and tags $\mathcal{W} = \{\mathtt{w}_1, \ldots, \mathtt{w}_V\}$ with the edges between these nodes representing the individual annotations. Note that the character $\mathtt{w}$ is used to refer to tags and $\mathtt{d}$ to refer to resources in order to be consistent with Information Retrieval conventions. Each assignment of a tag to a resource by a user is denoted as the relation $\mathcal{Y}$ and is typically called a tag assignment (*tas* for short). Therefore the complete folksonomy is a quadruple $\mathcal{F} := (\mathcal{U}, \mathcal{W}, \mathcal{D}, \mathcal{Y})$ and each tag assignment is a triple of the form (w_i, u_j, d_k) [HJSS06]. The complete set of tags used to annotate a resource over all users can be reffered to as that resource's description.

These entities are essentially meaningless in isolation; each tag is given meaning by the resources it is used to describe, each user's interests are described by the tags he or she uses and the resources he or she annotates, each resource is given meaning by the tags used to describe it and the interests of the users who have chosen to annotate it. Therefore each element in a tagging system is given semantic meaning by the other elements it is linked to. This dependance between entities for meaning is commonly referred to as "mutual contextualisation" [YGS07] and is where the real advantages of such systems are to be found.

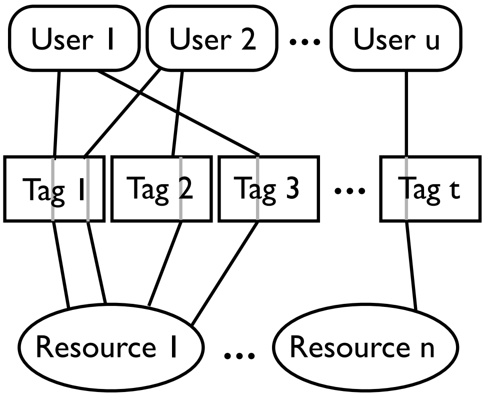


Figure 1.1: The structure of a folksonomy where User 1 has annotated Resource 1 with Tag 1 and Tag 3, User 2 has chosen Tag 1 and Tag2

The links between users can be implicit; users who are interested in similar topics are likely to be similar and we can discover these implicit links by analysing tag usage and resource annotation among users. Users are implicitly linked by the shared resources they tag and, reciprocally, resources are linked by the users who annotate them. In many tagging systems these user-user links can also be made explicit via the implementation and use of a friends system where people can maintain a contact list of friends on the system who they share a social relationship with.

Social network analysis is a key area of study and parallels can be drawn to work done in classical psychology dealing with how people communicate and form groups and with more formal network theory. By analysing people's individual networks of friends we can learn more about the resources being shared and discussed. In doing so we are able to expose more complex relationships leading to the discovery of latent groups or sub-communities. By utilising these relationships between users Marlow et al. [MNBD06] noted that tagging systems can be seen as complementary to collaborative filtering. Furthermore this complex network of associations has been used to assess the trustworthiness and relative expertise of users [NmAYG$^+$10].

Similarly, relationships between tags can be inferred based on their usage in the folksonomy. Tags that are frequently used together (to annotate the same resources) are said to co-occur and can be assumed to be in some way similar.

We can use these implied relationships between tags to better understand their semantic meanings or to construct term-hierarchies [PLG11]. Tags which are used to annotate the same resources are said to have a first-order co-occurence relationship, we can extend this notion further to a second order relationship by considering the potential similarity between terms that co-occur with terms that also co-occur with other terms. For example we may find that the tag 'fruit' frequently co-occurs with the tag 'apple' and we may also find that the tag 'granny smith' also co-occurs frequently with 'apple' . However it may be that the tags 'granny smith' and 'fruit' never share a first-order relationship but by considering the second-order relationships we discover that these tags are in some way related.

The freedom of choice permitted by an unrestricted vocabulary is seen as an important advantage for such systems where tags become more personally meaningful and the initial categorisation process is made easier. Social tagging systems facilitate traditional forms of classification and indexing by keywords but also allow for a new more personalised dimension of organisation and collaboration. Users are able to annotate resources with not only descriptive nouns, but also with expressions of opinion and contextual information pertinent to them [KC06]. The structure of links in the system allows users to *browse* through the resources, users and tags as well as search for specific tags. This provides a mechanism for discovering new resources and to find other users with common interests, which may not be possible or indeed practical with a more traditional text-based search.

Browsing and filtering content via tags is commonly made simple and intuitive in tagging systems by allowing users to click on any tags they see. For example a user may be interested in finding new resources about the British car marque Jaguar and may start by either clicking on or searching for the tag `jaguar`. However, as we shall discuss in more detail later, this is a good example of a highly ambiguous tag which has many different meanings and using it to query a large collection is likely to return a conflated set of resources covering many or all of these distinct interpretations [LZT09].

Users are much more likely to be willing to dedicate some time to manually annotating resources that either they have created or indeed that they find interesting or useful. A widely cited, early paper on social tagging by Mathes showed [Mat04] that not requiring users to pigeonhole their annotations into a rigid, pre-defined vocabulary lowers cognitive load significantly. Tagging dramatically lowers this perceived cost of annotation as there is "no complicated, hierarchically organized nomenclature to learn, users simply create and apply tags on the fly". Sinha [Sin05] argues that users generally find tagging much easier to use than a taxonomy-based system, particularly when classifying new items and revising existing classifications. Users don't report the same kind of difficulties discovered in studies of hierarchical systems either, such as the reluctancy to categorise an item, either because they are not sure which category the resource belongs to or because they are not confident in their ability to recollect at a later date how an item was categorised [AKD07a]. A further cognitive benefit of folksonomies is that the process of tagging utilises existing processes without adding to the cognitive load experienced by the user.

1.1.1 The Importance of the Community in Social Tagging

The idea of assigning keywords to media objects is hardly a new one, desktop software has allowed organisation and categorisation of photos, music and movies via free-form keywords for some time. However the option to tag resources in such systems is frequently ignored or underused. Rodden et al. [RW03] noted that while users of off-line systems rarely annotate their media, they generally feel that doing so will be useful and wish they did it more often. Clearly in this case the perceived benefits of annotation does not outweigh the investment of time required. However this may not be the case with online social tagging systems. Research has shown [Ame07] that people using similar media organisation systems online annotate their content much more frequently and that public photographs on the popular site Flickr are more thoroughly tagged than those that are kept private. In these online tagging systems the motivations for annotating have changed; it is no longer purely about self-recall and organisation but about sharing ones content and exposing work to the wider community. If authors annotate their content well it is much more likely to be discovered by other users than if it is sparsely annotated, or indeed not annotated at all.

Hence, arguably the most important aspect of social tagging is the community aspect, after all the word “social” is in the name. The dramatic increase in use of the Web in recent years had precipitated a sea change in how people communicate; lowering the - previously significant - costs of social sharing, group formation and collaboration. In his book “Here Comes Everybody” [Shi08], Shirky explains that having a shared pool of resources distributes the metadata creation workload amongst many contributors and increases the likelihood that a resource will be densely annotated.

Some seminal work on the usage of folksonomies [GH06] indicated that while most people tend to tag for their own benefit, the categorisations they choose can be of use to the community as a whole. It was found that after a relatively small number of users had tagged a resource, a nascent consensus forms that remains unaffected by the addition of further tags. Over time, tag use stabilises and the community forms an unspoken group consensus of how things should be categorised, creating a shared and agreed upon vocabulary [CLP07]. Users’ motivations to tag are still influenced by the desire to appeal to the community at large and it is this aspect that helps to reinforce the tagging process, encouraging further annotation of resources and sharing of knowledge [Fit06]. In these communities of practice even though the set of tags used at a global level is freely determined, patterns in usage rapidly emerge leading towards a shared terminology.

This so called process of semiotic dynamics [CLP07] - how populations of agents can establish and share semiotic systems, driven by their use in communication - is in many ways key to the usefulness and unrealised potential of the social tagging idea. If a consensus can be formed regarding the use of vocabulary and the semantics of said vocabulary then the system becomes stable and consistent. It does not preclude the possibility of new tags being used and included in the community and indeed it is these emergent semantics that allows such a system to grow and adapt to new topics and resources. This is again an area where the unrestricted vocabulary of social classification systems become a significant asset, allowing new terminology to develop naturally when it is required.

Over time a community-driven feedback loop allows an implicit agreement to be formed regarding the classification of resources, thus improving the likelihood that a given resource is accurately tagged. The end result of this process is a classification scheme that represents the community's perceptions of the resources within the collection. This complete set of categorisations includes the most commonly used tags - which most users agree on - as well as more idiosyncratic tags that may be useful descriptors for a small number of people.

Once they begin to reach some form of convergence these loose categorisation systems derived from the emergent, implicit information structures are referred to as "folksonomies", a term originally coined by Thomas Vander Wal in 2003. It is clear that there are a number of significant benefits to social tagging systems and that the resulting folksonomies can, over time, approach stable semantic meanings of terms agreed on by the tagging community.

Tagging systems can generally be partitioned into 2 distinct categories or types based on their chosen tagging model. These models are chosen based on decisions made about who should have the right to tag resources and as a direct consequence, how the overall folksonomic system will form.

The earliest and most common type is known as a **narrow folksonomy** or self-tagging, "where users only tag the resources they created" [MNBD06]. In such systems users are only able to tag the resources they contribute and not those provided by other users of the system, however they are still able to view and share content contributed by other users. A popular example of a narrow folksonomy is that of the online photo sharing web site Flickr where only the user who uploaded each photograph is allowed to assign tags to it. As a result the system only maintains a single set of annotations for each resource and the assignment of tags drawn from the vocabulary to a single resource is binary (either the tag is used to describe the resource or it is not).

In contrast to this are the so called **broad folksonomies** or "collaborative" tagging systems where users have the ability to annotate not only their own resources, but also the resources of others. In these systems a separate set of annotations is maintained for each user who is interested in a given resource. This model of tagging is almost always used for bookmarking sites such as delicious where users can either contribute new links to the global directory or choose to bookmark an existing link that she is interested in. When bookmarking an existing link the user is also encouraged to provide their own set of tags to describe the resource meaning that a single resource can be tagged by multiple users. In this model it is possible for the same tag to be used by multiple users to describe the same resource, meaning that the assignment of vocabulary words to resources is ordinal and this can be used to provide a weighting of tags to resources.

This distinction between tagging models is important as while both types of folksonomies share some common traits, the community tagging process in broad folksonomies does have a significant effect.

1.1.2 The Need for Social Annotation

The emergence of the web brings new problems: an ever-expanding and constantly evolving corpus generated by many millions of users. As the Internet continues to grow in size and scope, it is becoming increasingly apparent that the more traditional methods used for organising and locating data on it are insufficiently powerful [Lyn97, QCI04]. The recent changes to the way people use the Internet with the arrival of social networking and social media sites (such as Facebook, Flickr and delicious) have highlighted the need for a more collaborative and robust approach to categorising this mountain of data. Not only are these collections of data growing at an increasingly rapid rate but the data itself is also changing in many key ways. It is becoming more socially motivated and is evolving at a rapid pace. New services such as twitter and Facebook have precipitated the new trend of microblogging, an erratically changing stream of consciousness approach to communication and information sharing. Rather than simply being a collection of static web pages, this new form of online data is like a blog that is almost constantly being updated with new information, new terms, new phrases and new ideas.

If we look closely at the methods currently dominant for information classification on the Internet the issues with them and their lack of suitability for use with these new forms of media quickly become apparent. Initially, portal services such as Yahoo! provided a useful way of scouring the early web's relatively limited content by providing a classification system based on a common shared vocabulary [VG02]. This categorisation paradigm was derived from traditional library classification schemes which have changed little in design from the famous Dewey Decimal Classification System [Dew76], now the most widely used classification scheme in the world. They work by defining a set hierarchy of classification labels for items which can not be easily changed in structure once the system is in use. Each item, in the case of libraries a book and in the case of Yahoo! a link to a web page, is placed in the class to which the librarian or information architect believes it is most suited.

These categorisations schemes make a lot of sense for physical objects such as books and are a natural response to storage constraints. A physical book can only exist in a single place at a given time and so each item has to be given a single label or category in a hierarchical system. However on the web (particularly in social tagging systems) the "books" (resources) are no longer tangible real-world objects. They are therefore not subject to the same constraints; as Shirky [Shi05] succinctly puts it "in the digital world ... there is no shelf".

The top-down nature of these systems can also cause problems as documents will not always clearly fit into a single category and it is almost impossible to design a taxonomy that everyone will agree with. Class and cultural issues frequently crop up, particularly when the documents to be categorised are as diverse as they are in most libraries and particularly on the Internet. A classic example of this, perhaps a result of Melvil Dewey's Westernised values, is the way religious books are classified in the Dewey Decimal System. The system sets aside a full super-class (200-299) for religious books, subclasses 200-219 cover general religious issues and theology, classes 220-290 are devoted exclusively to Christianity leaving all other religions having to share the remaining section 290 between them.

It is no surprise, therefore, that as the content and scope of the web grew in size these methods no longer remained as feasible options. As a result the de-facto standard on the web is now the fulltext search engine. Search engines index documents automatically by exploiting statistical methods, such as term frequency, to establish keywords that describe resources and link structure as an indicator of an individual page's value [BP98]. By categorising documents based on their own vocabulary in an *ad-hoc* nature, rather than using a predefined hierarchical model the Google model allows for much greater flexibility. It could be argued that a large part of Google's tremendous success stems from the fact that they did not attempt to provide categorisations for the web *a priori*, rather they allowed the system to evolve organically. The PageRank algorithm allows relationships between categories to evolve in a similarly organic manner, via implicit connections made by the links defined by content authors rather than a single, fixed set of relationships defined by a professional ontologist.

While they are clearly far more suited to the task at hand, these methods still struggle to uncover the semantics of a resources content, only working at a superficial level. There is little definition to be found in terms of context and a lack of understanding of the main concepts behind a resource. This results in diminished precision in the resulting keywords [AKD07b], particularly when attempting to classify multimedia such as images and video where a large volume of easily machine interpretable data is not readily available [CW04]. Surely an obvious way to resolve this issue is to use humans rather than machines to label and classify such data.

Indexing is the process by which keywords are chosen that accurately describe the content and meaning of a resource which is to be indexed. Or, choosing terms whose semantics help in remembering the documents main themes [BYRN99]. According to Lancaster[Lan98] this process involves two steps: conceptual analysis and translation. Conceptual analysis involves deciding on what a given resource is about by breaking down or analysing concepts into their constituent parts in order to ascertain what is relevant in particular. The results of the conceptual analysis stage heavily depends on the needs and interests of users that a resource is tagged for; different people can be interested in different aspects. The second stage, translation, is where appropriate index terms are generated based on the results of the previous stage i.e. terms that best represent the substance of the conceptual analysis.

Many studies have shown that obtaining high consistency among different indexers is very difficult to achieve and can be affected by many factors including vocabulary, personal understanding of the resource and use of language [Hoo65, ZD69]. It has been shown that indexers are more likely to agree on the concepts that should be indexed rather than on the terms that best represent the concepts themselves [Iiv95], suggesting that the disparity issue occurs during the translation stage and not during the conceptual analysis.

This phenomenon has troubled IR researchers since its inception and was noted in the literature as early as 1958 when Vic Yngve presented a paper on the feasibility of text searching at the International Conference on Scientific Information (ICSI) [Yng59]. His insights on the ubiquity of ambiguity and on the need to find "formal connections between widely divergent ways of saying essentially the same thing" are generally considered to be prophetic and are still useful for informing contemporary research.

This lack of consistency among users is commonly a result of the so called "vocabulary problem," the natural variation in word use between people. In an extensive study Furnas et al. [FLGD87] showed that the probability that two people describe a given object with a common word is less than 0.2 (1 in 5). In many cases this problem was discovered to be so difficult to overcome that in 1985 Blair wrote "Stated succinctly, it is impossibly difficult for users to predict the exact words, word combinations, and phrases that are used by all (or most) relevant documents and only (or primarily) by those documents" [BM85].

It is clear that the sheer number of resources to be indexed on the Internet makes manual indexing from domain experts or professional indexers impractical. It can also be said that, particularly for resources such as images, automated indexing does not provide accurate enough results. [AKD07a] showed that folksonomy tags agree more closely with the human generated keywords than those that are automatically generated and it is therefore possible that folksonomies might offer a solution to this problem, providing a cheap source of semantically meaningful index terms.

Having said that, the issue of (lack of) agreement with regard to index terms is a significant problem and can inhibit the usefulness of tags provided by users to categorise resources. Furthermore as peoples' motivations for tagging resources vary it is important to understand these motivations before we can begin to make use of folksonomies for the purposes of information retrieval and organisation.

1.1.3 Motivations for Tagging

A fundamental question raised when discussing social tagging systems is why are they so popular and why do people so happily give up their own time to tag resources? After all, people are generally very reluctant to give up their own free time to do work if there is not some form of reward or incentive for doing so. A related topic of discussion is what motivates people to tag resources online, how does this affect the terms and specific language used and does this have implications for deriving hidden semantics from tagging data?

[Ame07] studied data from the social photo sharing site Flickr in order to uncover what incentives and motivations existed for people to annotate their images. They cited a study [RW03] showing that in offline image organisation systems (such as Apple's iPhoto) users very rarely bother to annotate their photos, although they do see the benefit of doing so and wish they did it more often. Clearly in this case the perceived benefits of annotation do not overcome the required investment of time.

In contrast, Ames et al. showed that users of Flickr very rarely fail to annotate resources at all and posit that this is because the motivations for tagging are fundamentally different in the online setting. In social systems, tags not only facilitate search and recall by the owner of resources but also enable discovery of potentially interesting resources by the community at large. Therefore the traditional use for annotation is now augmented by the ability to expose ones work to the online community. The study found that in only a small number of cases users annotated their photos solely for the purposes of personal retrieval and organisation.

Four distinct categories of motivation over two dimensions were discovered, which are shown in Table 1.1. The first dimension has two classes; "social" and "self," both referring to who the tag is intended for. The second dimension describes the function of the tag; whether it is intended for organisation and search or for communication and personal expression. Personal expression tags

	Organsation	**Communication**
Social	Retrieval/directory	Context for self
	Search	Memory
Personal	Contribution/attention	Content descriptors
	Ad-hoc photo pooling	Social signalling

Table 1.1: Taxonomy of motivations for tagging photos on Flickr according to Ames and Naaman

such as ‘awesome’, ‘funny’, ‘inspirational’ and ‘helpful’ may be of considerable benefit as they indicate the usefulness or quality of a resource. They provide index terms which are very unlikely to be present in the resource itself and would therefore not be available to a more traditional search engine.

A similar study by Zollers [Zol07] examined the tags on Amazon.com and Last.fm and found comparable motivations. They identified several more specific emerging motivations including self-presentation, expression and activism. For example people used tags on Amazon to indicate how good they thought a product was and on Last.fm to express their views on music and bands. Self-presentation tags are when users “write their own identity into the system,” for example on Last.fm users tagged bands with ‘seen live’, ‘songs from my youth’ and ‘my favourite’. Activism tags are a direct result of the inherently social and collaborative nature of online tagging systems and allow people to form action groups by using the same tag to annotate resources. For example the tag ‘defectivebydesign’ is used by people to denote products or services which they believe should be avoided due to that product’s use of DRM (Digital Rights Management) technology.

Many of these motivations for tagging are an attempt to add personal meaning to a resource, which may help the user to later re-find that item. Morville [Mor05] suggests that in an environment such as the Internet where there is an excess of content, findability and re-discovery of content is critical. Proponents of social tagging systems therefore cite this as one of the key advantages of an unrestricted vocabulary and posit that while the initial motivation in this case is for personal benefit, the additional information can still benefit the community in ways traditional keywords would not. Other motivations are more explicitly social where the original intention is to communicate views, opinions or information to other users of the system.

The combination of the various motivations identified for tagging suggest a number of potential issues with regard to direct application of tagging data for information retrieval and searching purposes. Clearly in some cases tags used are extremely specific to the user who tagged a resource and therefore may not be of any benefit to the wider community. However, they do highlight that in many cases in social tagging systems not only do users feel they should annotate resources for their own future recall, but that they also appreciate a need for the resource to be discovered by other users of the system. Therefore it is assumed to be more likely that the user will choose a more diverse, descriptive and less ego-centric set of tags than if they were purely tagging for their own benefit.

1.1.4 Statistical Structure of Folksonomies

A significant area of research into social tagging has been in analysing the statistical properties of large-scale, real world folksonomies. In this section I will discuss a number of seminal early studies and will comment on their findings with regard to the statistical structure of tagging systems and the patterns that frequently emerge. We will leave discussion of actual term usage and issues borne out of this for later chapters.

An early - and heavily cited - journal article by Golder and Huberman [GH06] analysed 2 sets of data from the social bookmarking site delicious, a very good example of both a broad folksonomy and a social bookmarking system. In delicious users can either add new URLs (resources) to the system or 'bookmark' existing resources by annotating it with their own set of tags. The first set comprised of a sample of 212 most popular URLs from delicious. The second set was the complete collections of 229 users sampled at random.

Unsurprisingly they found that tagging behaviour varied greatly over the subset of users in the sample with some users making frequent use of the system and others only using it very infrequently. However they also found, perhaps somewhat surprisingly, that there was not a strong relationship between the number of distinct resources a user had tagged and the number of tags used. They did however show that the list of distinct tags used increases over time as the user tags more resources, particularly as they discover new interests. These growth rates varied immensely over different users and tags, perhaps reflecting how users' interests and each tag's popularity changes over time.

By looking at individual URLs the authors remarked that the vast majority of resources are annotated by users very quickly, with the rate of new annotations decreasing over time. Some resources on the other hand do not show this trend and may not be densely annotated to begin with but experience several modal peaks of popularity over time. These sharp peaks may be a result of the resource being 'rediscovered' by the community, perhaps because the topic covered by the resource has experienced a resurgence of popularity or because it has been referenced on a popular web site or blog. They may alternatively be the result of a popular (or 'hub') user choosing to bookmark the resource thus temporarily increasing its exposure to other users of the system.

In this study it was found that 67% of resources reached peak popularity within the first 10 days of being on delicious, 17% of which reached their peak on the first day. This distinct peaking of popularity is frequently referred to as 'burstiness' and is also typically displayed by not only resources but also by tags themselves. A resource or tag's surge in popularity is self re-enforcing as popular resources are displayed prominently on the 'popular URLs' page of the delicious web site.

Surprising regularities were found in tag frequency, user activity and resource popularity and discovered that after only a small number of users had tagged a resource its tag set tended to converge to a stable subset of popular tags. They argue that this subset of "stable" tags represents the community's consensus on the best way to annotate the resource. As users add more resources to the system their sets of distinct tag terms grows, however the rate at which this set of tags grows differs greatly between different users. This result is notably similar to the stochastic Pólya urn model [Mah08] where a stable pattern eventually emerges from what appears to be an entirely random process. Generally speaking this stable pattern emerges quickly, usually requiring fewer than 100 bookmark events and thus, it is argued, resources do not need

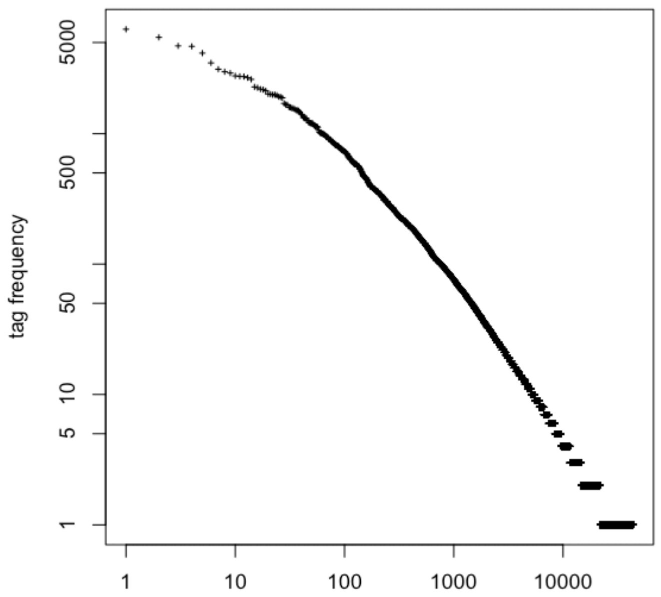


Figure 1.2: Log-log plot of tag frequency distribution in Bibsonomy showing characteristic power law

to become particularly popular before the combined tag data is useful.

Similar analyses are performed in [KC06], however the focus in this paper is more on how similar tagging is to conventional indexing. As subsequently found by future studies (such as [SvZ08]), a consistent characteristic of folksonomies is that the frequency of use of tags follows a power law. That is that a small number of tags are observed very frequently with the frequency of use tailing off sharply. This pattern follows Zipf's law "given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table". This distribution can be seen clearly in the log-log plot of tag frequency in Figure 1.2. It has been noted [Shi05] that for some folksonomies the drop-off does not occur immediately and is initially less steep, perhaps indicating that the resources in that folksonomy cover a broad range of topics.

In a broad folksonomy we can analyse tag distributions and patterns for individual resources as more than one user can annotate a given resource and it is therefore possible for tag frequencies to be ordinal numbers rather than simply binary. Kipp et al. [KC06] note that in cases where a resource has been tagged by a sufficiently large number of users, a similar frequency distribution begins to emerge with a small number of very stable high frequency tags. This suggests that the community has to some extent agreed on how a resource should be categorised, however the so called 'long tail' - the list of tags that do not have a high frequency - may have more personal meaning for individual users and are therefore also useful. It is noted that this may indicate that "taggers apply tags according to a mix of communal and individual notions of aboutness and usefulness".

The analysis of tagging data performed in these early works provide useful information describing the general structure of folkosonomic data and offer illuminating insights into the patterns that emerge. It is clear that tag distributions tend to form in a similar manner to terms in free-form text and in natural language in that a small number of agreed upon tags tend to dominate. As use of the system increases the community forms a general consensus on what topics are prevalent in both individual resources and also in the web site corpus (the entire collection of resources). The rate at which tag frequencies drop off for a resource indicate the perceived breadth of topic coverage of that resource as defined by the community.

These communal tag sets grow and develop in a very consistent and predictable manner and as such are fairly consistent with conventional indexing terms. However, the 'long tail' of tags beyond the initial high frequency peak are more personal, idiosyncratic and therefore provide useful classification information regarding the aboutness of a resource that is beyond the scope of traditional classification schemes. These two properties give credence to the notion that tagging systems may be extremely useful as a cheap source of metadata, however they also highlight the fact that in their raw and unprocessed form they can be difficult to interpret.

1.1.5 Problems with Tagging Data

As with any naturally evolved system, the set of classifications chosen and used in tagging systems will be imperfect and will suffer from a number of problems related to language use. We can pigeonhole these problems into 3 categories: polysemy, synonymy and basic level variation. Polysemy refers to words which have more than one meaning, for example the word ‘bat’ may refer to a small flying mammal or it may also refer to a piece of sports equipment. In many cases the meanings are very different and this can have an effect on the precision of search results: if you were looking for information about baseball bats then information about the habitats of fruit bats would not be very helpful.

Synonymy refers to different words with the same (or at least similar) meaning, for example the words ‘computer’ and ‘pc’ both refer to the same thing but are quite different. This relates back to the age-old vocabulary mis-match problem, however the issue is particularly prevalent in social tagging systems where vocabulary is unrestricted. This is compounded when you consider how few annotations are generally available for each resource, meaning that it is highly unlikely that different synonyms will be used together. Expanding on the classifications noted by Golder et al. [GH06] we can also consider the use of different languages as being a strongly related problem. Going back to the tag ‘television’ we can expect for example that an Italian user would be more likely to use the tag ‘televisiore’ instead, further decreasing the likelihood of users choosing the same descriptive terms. Traditional methods of dealing with this problem such as word stemming will not resolve the problem for the vast majority of cases where there is a language mis-match or where completely different terms are being used.

The third category of vocabulary problem is that of basic-level variation. This occurs because people have different levels of familiarity of knowledge of items and as such may choose more or less complex or specific terms for a resource. Even in cases where people have a similar level of understanding it is common for them to use different levels of sophistication of granularity in their tags. For example the difference in granularity and sophistication between the tags ‘animal’, ‘cat’, ‘tiger’ and ‘panthera tigris tigris’.

These issues are the primary motivation for the use of machine learning techniques, and particularly dimensionality reduction, to uncover and make use of the implicit hidden relationships inherent in social tagging data and informs the work in this thesis.

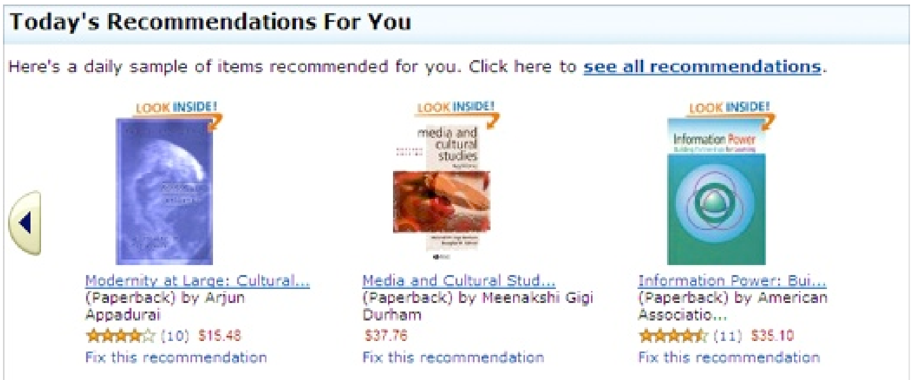
1.2 Collaborative Filtering with Ratings

The previous section discussed the process of social tagging in which users submit short descriptions of resources either already in the system or when contributing a new resource of their own. We can consider that these tags indicate an implicit association of the user with the resource, suggesting that the user likes (or is at least interested in) the item. Another form of socially contributed data available on many modern web sites are where users assign ratings to items, a more explicit indication of interest or utility. Some sites also exploit less explicit ratings and may obtain binary associations by considering a user's purchase or bookmark history or by utilising tags as indicators of interest, as described above.

Explicit ratings systems are commonly found on movie and music recommendation sites such as MovieLens (`http://movielens.org/`) or imdb (`http://imdb.com/`) where users can give each item a rating (usually from 0 to 5 stars). Implicit systems are used in online retail stores such as Amazon (`http://amazon.com/`) or can be used within a desktop application, in the case of iTunes (`http://apple.com/itunes/`). In these systems users purchase items or add them to a wish list, indicating that they are interested in that kind of item. Ratings are generally chosen from a discrete set of values, for example any number of stars between 1 and 5, however for the purposes of modelling this data it can often be appropriate to treat them as being from a continuous, but bounded, range. Note that in some cases both forms of information can be used, for example Figure 1.3 shows an example of both explicit ratings and implicit information in use on Amazon's web site.

1.2.1 Data Structures and Goals

We can formalise the ratings mathematically in a very similar way to tags: we have a set of items (like resources) $\mathcal{M} = \{\mathtt{m}_1, \ldots, \mathtt{m}_M\}$, users $\mathcal{U} = \{\mathtt{u}_1, \ldots, \mathtt{u}_U\}$ and a set of discrete rating values $\{\mathtt{r}_1, \ldots, \mathtt{r}_R\}$. Each individual rating i for an item by a user is also a tuple: (u_i, m_i, r_i) and there are a total of N tuples in the system representing all ratings the users have supplied. For example the



Today's Recommendations For You

Here's a daily sample of items recommended for you. Click here to see all recommendations.

Modernity at Large: Cultural...
(Paperback) by Arjun Appadurai
★★★★☆ (10) $15.48
Fix this recommendation

Media and Cultural Stud...
(Paperback) by Meenakshi Gigi Durham
$37.76
Fix this recommendation

Information Power: Bui...
(Paperback) by American Associatio...
★★★★½ (11) $35.10
Fix this recommendation

Figure 1.3: Example of collaborative filtering being used commercially on Amazon

tuple (u_i =1, $m_i = 1$, $r_i = 4$) would indicate that user 1 had given item 1 a rating of 4. Again, in a similar vein to tags, it is usually convenient to visualise the complete collection of ratings as a large, (typically) very sparse matrix R of size $U \times M$ where r_{um} indicates the rating given by user u to item m.

Given the collection of ratings provided by users, the goal is to attempt to use the data to suggest more items or resources each user may like or be interested in. This process of machine recommendation is frequently called "ratings-based collaborative filtering" or simply "collaborative filtering" and the systems themselves are frequently referred to as "recommender systems". The process is in fact very similar to information filtering and has significant links to more orthodox information retrieval. Collaborative filtering systems can be placed in the context of information retrieval by considering that in a retrieval system items are pulled to users by the issuing of explicit search queries. Filtering systems on the other hand are described as push systems since they quite literally push those items at a user that they predict the user will like.

For some kinds of items is also possible to make recommendations based on their content or some attached metadata. This is known as content-based filtering and functions in a very similar way to classical information retrieval. Such systems attempt to find items that are similar to items the user has already indicated as being of utility to them and rank them in ascending order of distance. This is done by using heuristic methods such as the cosine distance between TF-IDF content vectors [LZ04, Lan95] or the Winnow algorithm [Paz99]. Other approaches make use of more principled machine learning methods such as Bayesian classifiers [MBR98] and decision trees [PB97]. This work, however, will focus on the task of item recommendation based solely on ratings data.

There are two main potential outcomes of such systems: firstly the suggestion of items to users they may like and secondly the related task of predicting the rating a user will give to an as-yet unrated item. If a system is able to predict unknown ratings then the suggestion task can be achieved by simply ordering unrated items in descending order of predicted rating. The intuition behind collaborative filtering is that similar people tend to like similar things, therefore if you can reliably identify people similar to the “target” user you can use their ratings of items to infer what rating the target user might give. More modern systems also make the assumption that there is some consistency in how an individual user will rate items and also how an individual item will be rated by different users.

By utilising ratings we obviate the need to automatically interpret content, which is a very difficult and error-prone process. This allows the same algorithms to be applied to any kind of items be they textual, photographic, musical or even more abstract such as items for sale in an online store. We can use such systems to filter and recommend diverse items based on subjective and hard to represent concepts such as taste and quality. For example it is often very difficult to describe exactly what it is about films, songs or artworks that you like however it is quite easy to provide specific examples. Similarly when looking for new content it is difficult to describe in a textual search exactly what it is you are looking for and again, providing examples as a reference point is much easier.

A small example extract (or fragment) of a ratings matrix from a music recommendation site is shown in Table 1.2. The matrix indicates that the user *Ian* has given the album *The Joshua Tree* a rating of 5 and *London Calling* a rating of 4, however he has not yet rated *Pet Sounds*. A recommender system should be able to predict all of the unknown ratings (indicated by $\emptyset$) within the matrix based on the ratings which are known.

	The Joshua Tree	Pet Sounds	London Calling
Steve	4	2	5
David	$\emptyset$	4	2
Ian	5	$\emptyset$	4
Emma	5	$\emptyset$	$\emptyset$

Table 1.2: Small fragment of a ratings matrix for a music recommender system

1.2.2 Background

Formal work in the field was started in the early 1990s [GNOT92, SM95] however it can be argued that the concept is much older than that. Perhaps the earliest example of a collaborative filtering system is the so called "Grundy system" [Ric79] which suggested the use of stereotypes as a means of building models of users. Each user was assigned to the closest stereotype and predictions were made as to what books the user would like based on the stereotype profile. The Tapestry system [GNOT92], on the other hand, relied on individual users to identify neighbours manually who were then used to suggest items. Later work such as GroupLens [KMM+97] were seminal in the development of fully-automated recommender systems.

Collaborative filtering algorithms can be generally classified into 2 distinct types: memory-based and model-based. Early systems were memory-based and make use of the original ratings matrix in its entirety to formulate predictions. The vast majority of such systems operate via a relatively simple 2 step process. First they identify a neighbourhood of users similar to the target user and then use an aggregate weighted summation of the neighbours' ratings for an item in order to predict the rating for the target user. These algorithms form the basis of most filtering currently performed on the Web including sites such as Amazon and CDNow and were the cornerstone of much early research in the field [GNOT92, BHK98]. We refer to [AT05] for a much more comprehensive description of how these methods operate. It has been speculated that their popularity is due to their relative simplicity and their inherently intuitive nature [Hof04].

Unfortunately these simple, memory-based algorithms suffer from a number of major shortcomings. The number of items rated by most users is oftentimes small and therefore it can be difficult to choose a good neighbourhood of similar users. Once a neighbourhood is chosen only a very small number of similar users may have rated the item for which we wish to predict a rating leading ultimately to suboptimal accuracy. Beyond the recommendations made, memory-based systems do not provide much scope for data mining and learning from the user profile information collected. Furthermore it is difficult to design neighbourhood-based systems that do not bias strongly towards certain users, particularly those who submit a large number of ratings.

More recently a new approach to solving the problem has become more popular: the model-based approach. These newer systems use the observed ratings to construct a model of the data, usually based on some form of dimensionality reduction to uncover latent factors. These latent factors are constructed in a manner that best explains the training ratings and if we make the assumption that any further ratings will be drawn IID[1] from the same distribution then the model should be able to predict new ratings well.

The recent resurgence of interest lately is primarily due to the Netflix prize [Pat07]. Netflix are an online DVD rental company who had been using collaborative filtering as a means of recommending films to users but were not satisfied with the quality of existing recommendation systems. They offered a significant prize to any team who could improve upon their algorithm by more than 10%. Many attempts to solving this problem use gradient descent algorithms to estimate a Singular Value Decomposition (SVD) of the original sparse ratings matrix [Pat07]. These methods will be explained and discussed in more detail in the next section.

1.2.3 Difficulties and Challenges

Collaborative filtering has proven to be a very difficult problem due to a number of factors. Firstly the sheer number of items and users in a typical online store or media recommendation site necessitates that the algorithm be both fast and highly scalable. Due to their design, early recommender systems suffered from severe scalability issues, particularly for users with large profiles.

[1] IID stands for *independently and identically distributed.*

This is because the process of determining good neighbours has a computational complexity of at least order n^3 and as such any large increase in the number of possible neighbours will have a significant impact on the time required to complete the task. As we shall see in the next section, modern algorithms seek to get around this problem by making use of dimensionality reduction techniques and I follow a similar approach in this work.

The second, and perhaps more obvious, challenge for collaborative filtering algorithms is to improve the quality of recommendations made to users. This is of particular importance in applications where purchase decisions may be made on the basis of recommendations. After all, if a system is recommending items to users with the intention of encouraging them to purchase said items then it is important to be certain of the recommendations otherwise distrust of the system could result.

Recommendation errors can be said to fall into 2 distinct categories which relate to type I and type II errors in statistical hypothesis testing. Type I errors are false positive and are potentially the most serious of the two. They occur when an item is recommended to a user that the user does not like. This type of error is generally the most frustrating for users, especially if there is no clear rationale behind why the system is recommending some items over others.

Type II errors are less likely to irritate users but can ultimately cost a company sales so should still ideally be kept to an absolute minimum. These errors occur when the system fails to suggest an item to a user that the user does actually like. Both of these error types can be minimised by increasing the overall accuracy of predictions, or more objectively to minimise the prediction error. However, much like in statistical hypothesis testing, reduction of one error is likely to increase the incidence of the other type so choosing an optimal model that mediates between the 2 error types is of critical importance.

A third problem, which is somewhat related to the first, is the sheer sparsity of the ratings matrices in real-world systems. Amazon sell millions of different products to hundreds of millions of customers. Individual users will only have rated (either implicitly or explicitly) a very small number of items in the complete product catalogue and as a result a massive proportion of the matrix will not have a rating. For example, imagine a system consisting of "only" 1000 items and 10,000 users. This system will have a ratings matrix with 10 million individual user-item combinations and in practise much less than 99% of these cells will be filled-in. Therefore systems must have the ability to make a very large number of (hopefully) accurate predictions based on only a very small amount of data.

1.3 Summary

In this chapter I first described in detail what social tagging is, how it can be used in practise and what its perceived benefits are. I have also given a brief introduction to the concept of rating-based collaborative filtering and outlined some early background work in the field as well as challenges to be overcome. The next section will look specifically at 3 problems in social systems that I attempt to tackle in this thesis and discusses why it is important to research better ways to deal with these issues. Furthermore prior work related to these problems is detailed and their relative merits and drawbacks discussed, providing motivation for the later techniques and models that form the main contributions of this thesis.

Chapter 2

Related Work

"What we work on today, others will first think of tomorrow."

Alan Perlis

Online social collaborative systems represent a very large area of study and there are certainly many avenues for possible research. Such work is necessary so that we may better understand and use the information contained therein. The focus of this thesis is on three main problems; two relate to tagging systems and the third applies to collaborative filtering. While these works apply to different problems they are all strongly related and are all based on Bayesian statistical models which are explained in detail in the next chapter. Rather than simply using existing techniques to work with these new sources of data, this work instead involves the design of novel models specifically adapted to the form and statistical properties of the data. This section briefly describes the three problems tackled and discusses related work pertaining to each of them. The three problems tackled are as follows:

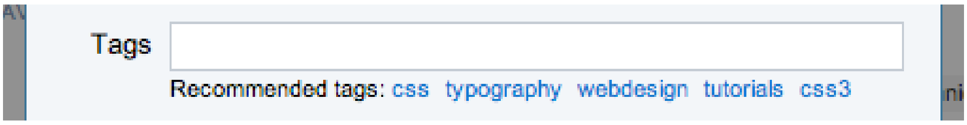
Tag suggestion One of the problems facing the use of social tags for resource description and item metadata is the small number of tags assigned per resource on average. Tag suggestion systems attempt to partially alleviate this problem by suggesting additional (relevant) tags to users when they are annotating a resource in the hopes that they will also add some of the suggested tags and thus increase the total tag count. The methods presently used on social tagging web site tend to be very simplistic in nature and are often a global list which is not specific to the item being tagged.

Personalised search When searching on the web, search engines generally make the assumption that responses to queries are user-independent, that is if two different users submit the same textual query then they are looking for the same thing. However it has been shown that most queries are short and many are highly ambiguous with several possible interpretations of their meaning [HO06, TT10]. Personalised search attempts to gain some understanding of a user's interests from some user profile any uses this information to improve the accuracy of search results by leveraging this extra information. Search in social bookmarking systems tends to be particularly frustrating due to a number of factors. Therefore it is an area where improvements in the general search algorithm and attempts at personalisation have the potential to make material improvements to search results.

Collaborative filtering Many web sites on the Internet allow users to provide ratings for items to indicate how much they like that item or how interested they are in the item. This method of determining people's interests is commonly used by online stores such as Amazon or on recommendations sites such as MovieLens. Collaborative filtering algorithms use these ratings in order to form profiles of user's tastes and interests and then use these profiles to suggest new items to users that they may like.

2.1 Tag Suggestion

As we have seen, social tagging systems provide a new way for Internet users to organise and share their own digital content and content from other users. Users are able to annotate each resource with any number of free-form tags of their own choosing without having to adhere to an *a-priori* set of keywords. Unfortunately the ease of use and freedom of word choice this allows comes at a significant cost. If each user is free to choose whatever tags she wishes then



Tags

Recommended tags: css typography webdesign tutorials css3

Figure 2.1: An example of tag suggestion on delicious.com

it is unlikely that other users will choose exactly the same tags to describe the same resource or indeed to tag similar resources they have found. Many studies have shown that obtaining high consistency among different taggers is very difficult to achieve [ZD69, Hoo65]. These factors result in the categorisation scheme displaying a number of highly undesirable characteristics such as polysemous and synonymous terms which make searching or browsing through the collection difficult and inaccurate.

This lack of a consistent and shared vocabulary also results in a large number of unique or "singleton" tags appearing in the folksonomy. Sigurbjörnsson and van Zwol investigated [SvZ08] the characteristics of a large sample of the Flickr database (which can be taken as a good reference point for most large-scale tagging systems) and found that the tag frequency closely follows a Zipfian distribution. This is where a small number of tags are used very frequently with tag use quickly tailing off leaving the so called "long tail" of infrequently used tags. Generally speaking, the tags at the extreme ends of the distribution are not particularly useful; the high-frequency tags are too generic and the singleton tags tend to be either compound phrases or misspellings and are likely to only be useful in very specific cases. The distribution of tags per resource was also found to follow a power law with a small number of resources being very thoroughly annotated and a large majority (64%) having only one, two or three tags.

To assist the user when tagging new resources, most of these systems offer some form of tag recommendation to increase the chance that a given resource is tagged and also to increase the average number of tags assigned to each resource in the system. These systems are primarily based on the observation that in many cases a user will tag resources with tags other users have already used, provided they agree with those tags. When a user goes to annotate a new resource they are presented with a list of recommended tags that they can choose to use or in some cases these suggestions may assist or prompt the user in coming up with their own tags. Figure 2.1 shows an example of tag suggestions on the popular social bookmarking site delicious. Users can either enter their own tags in the input field or click on one of the recommendations below to automatically add those tags.

A user study we conducted which involved users tagging images where only half of the users were given tag suggestions (in the form of a tag cloud) indicated that the suggestions served to increase both the quantity and the subjective quality of the tags [HBRE09]. The research also indicated that the users who were shown the tag suggestions more quickly converged on a group consensus on the most appropriate tags for the resource. Sood et al. [SOHB07] explain that providing tag suggestions “fundamentally changes the tagging process from generation to cognition,” serving to reduce cognitive load on users and expedite the tagging process. In many cases the tag suggestions can be improved further once the user has provided a few tags of their own and can be adapted to the user’s own interests and word choice.

2.1.1 Early Approaches

Despite their clear utility for improving social tagging systems, the literature on tag recommendation is still quite sparse, particularly in the case of personalised methods. Many early approaches tended to be based on a mixture of the most popular tags and tags which the user has used previously or by reusing existing techniques from information retrieval [Mis06, BWC07]. Recently more sophisticated systems have been proposed, focussing on methods derived from collaborative filtering and simple co-occurence data or making use of information other than the tags provided by users (for example the HTML content of web pages) [Sch06, GW08]. Research by Jäschke et al. [JMH$^+$07] has shown that these unadapted techniques are unable to perform well in real-world scenarios and in fact are unable to significantly outperform much simpler methods based purely on tag frequency. The method presented in [SvZ08] also uses simple co-occurence data but augments it by promoting tags based on two heuristic measures. The method boosts the ranking of tags which are both highly descriptive in that they are not in the head of the tag distribution (i.e. are not stopwords) and “stable” in that they are not only used by a very small number of users.

[SOHB07] focusses on tag suggestion for blog posts based on the tags used to annotate other posts in the blogosphere that have similar content. [HRM08] treats suggestion as a binary classification problem where each possible tag from the complete vocabulary either does or does not describe the resource. The algorithm also incorporates link data and content of web pages as features and uses Support Vector Machines to perform the classification task. This limits the application of this algorithm to social bookmarks as these features are not available for other resource types such as images, films and products.

2.1.2 Modern Approaches

In [JMH$^+$08] the authors reduce the tri-partite social tagging graph into three two-dimensional matrices and then use a collaborative filtering on these algorithm to generate suggestions. While [HJSS06] propose a method based on random walks around the folksonomy graph, much like the PageRank algorithm made famous by Google. The algorithm works by assigning importance weights to the links in the graph where importance is propagated around the graph via the random walk. For example a tag is important if it is used by important users and if it used to tag important resources or if it co-occurs with other important tags. A similar method, based on the older HITS algorithm, is detailed in [XFMS06]. Both of these systems suffer from their computation complexity which is a direct result of the sheer size of the tri-partite folksonomy graphs found in real-world tagging systems.

Many modern tag suggestions algorithms make use of some form of dimensionality reduction to improve the quality and variety of suggestions and to get round the problem of sparsity. For example [WZY06a] modelled broad folksonomic data using a simple Separable Mixture Model representation which reportedly worked well. However it makes the assumption that the probabilities of a user, a tag and a resource are all independent given a dimension d_α. It is also not an entirely generative model and does not make use of a Bayesian hierarchical structure when inferring parameters, meaning that it could easily suffer from problems of over-fitting. A similar model [PL07] was used to recommend resources to tagging system users, however this too suffers from similar drawbacks.

The work presented later in this thesis builds upon the knowledge gained from these previous attempts and also makes use of dimensionality reduction but approaches the problem from a more principled and Bayesian viewpoint.

2.2 Personalised Search

As highlighted in the previous chapter, term use in social tagging systems tends be very inconsistent between different users resulting in a large number of polysemous and synonymous tags. This has a highly detrimental effect on search performance unless the system deals with this inherent variation in some way and makes search in such systems a frustrating task. This problem is not restricted to the domain of social tagging and was identified early in the development of information retrieval systems [Yng59], however due to their unrestricted vocabularies and inherent data sparsity it is a more common issue in social tagging systems. This issue is compounded by the fact that the vast majority of search queries are short (usually less than three terms in length) and are frequently ambiguous in nature [HO06, TT10].

2.2.1 Search in Social Tagging Systems

In current social tagging systems, search algorithms tend to be rather simplistic in nature, often relying on simple term matching algorithms in order to rank resources given a query and seek to exploit the aggregated annotations across all users, the so called “wisdom of the crowds”. This simple approach to the problem fails to deal with the vocabulary problems noted above and can result in quite poor rankings, particularly when users make use of very specific or unusual tags. One potential method of reducing this ambiguity and thus improving search performance is to use some form of dimensionality reduction so that terms which frequently co-occur and are therefore likely to have a similar meaning, are in some way grouped together or implicitly linked. By doing so we can reduce the requirement on the user to choose exactly the same terms for a query as those used to annotate the relevant resources.

Consider a resource about a laptop computer which has been annotated by a knowledgeable user with the tags “macbook pro” and “core 2 duo”. A less knowledgeable user may be searching for this resource and may not know the specific terminology and as a result will use simpler search terms such as “laptop” and “computer”. Or, alternatively, the searcher may have a little knowledge of the terminology but misspells some of the query terms, for instance “macbookpro”. In a search system with no dimensionality reduction the relevant result will be ranked very low as its annotations do not contain the exact terms of the user’s search query. However a reduced dimensionality system does not rank resources based purely on matching terms, but does so by calculating a probability (or distance) of each resource given the query terms over the lower dimensional space. Since there is no requirement for the terms to match exactly and the system will have reduced all of these terms to the same dimension(s), it is highly likely that the relevant resource will be given a high rank for this query, thus allowing the user to fulfil their information need.

Another possible way of dealing with the inherent ambiguity of search queries is to attempt to personalise the search results based on the user’s preferences or interest profile. In the case of social tagging data we can build such user profiles implicitly by looking at the resources the user has bookmarked and the tags they have used to annotate these bookmarks; the user’s tagging history. Previous studies have suggested that while it can be difficult, if done correctly, personalisation can indeed improve the quality of search results [DSW07].

A classic example where understanding the user’s interests is of clear benefit is when the user enters a vague and highly ambiguous query. For example a user interested in astrology may want to find articles about the star sign *Cancer* and may simply choose to enter the query “cancer”. It seems a reasonable assumption that such a query would provide good results, however the word *cancer* has another very different meaning. At the time of writing, entering such a query on the Google search engine returns absolutely no results pertaining to the astrological meaning of the word within the first page of results. However in a personalised system the user's preference for astrology would cause results relating to this topic to be pushed up the rankings, making it much more likely that the user will easily find a relevant result.

Previous attempts have been made to improve search performance in tagging systems, however almost all large tagging systems on the web still use simple term matching or standard IR techniques. [KHS08] studied the performance of "traditional" search systems on tagging data and found that they performed poorly, suggesting that more novel approaches were needed to yield acceptable results from tagging data. As a result more successful algorithms are designed to work more in concert with the kind of data obtained from such systems. Work by Hotho et al. [HJSS06] utilised graph theory techniques based on the famous PageRank algorithm to rank documents. The authors conclude that enhanced search facilities are vital to support emergent semantics in tagging systems and found that their algorithm was good at identifying latent communities of interest.

[RHMGM09] investigate the use of tags from Delicious as additional source of data to assist in automatic clustering of web pages. Their results show that principled inclusion of tagging data can improve model quality and aid in the clustering process. They use both k-means and topic modelling based approaches and find that the latter significantly improves on the former indicating that such models are a good fit for tagging data. This work provides an interesting insight into how our own models may perform however it differs significantly from this work as it does not attempt to rank resources solely on tagging data and does not attempt to personalise the results.

2.2.2 Personalised Search

In more recent but similar work [VCJ10] the authors describe methods of deriving user profiles based on data obtained from social bookmarking systems to personalise search results on the Yahoo! search engine. However, again they do not attempt to apply this model to rank resources in the bookmarking system itself, they use it to *re-rank* the top URLs returned by the Yahoo! Boss

API based on the user preferences obtained from delicious data. Their results and methods are therefore not comparable with those described in this paper.

Closer to the work described in this thesis is [WCY$^+$10] where the authors also attempt to provide personalised rankings using social tagging data. We discuss their models later on and use the best performing one (when applied to our data) as a highly competitive baseline. In this case the authors use Language Modelling techniques to estimate probabilities of resources given tags and tags given users. They use the resulting parameters to rank resources given single term queries and compare various smoothing methods for obtaining these estimates.

Other uses for personalisation in social tagging systems have been investigated and several papers have looked at providing personalised tag suggestions to users when annotating resources, this includes work by Sigurbjörnsson et al. [SvZ08]. Work by Krestel et al. [KFN09] explored the use of topic models for tag recommendation and by extension to improve search results, however they did not make any attempt to personalise the recommendations.

Outside of social tagging, there have been a number of studies on the possibility of personalising search systems. For example Dou et al. [DSW07] investigated a number of methods for creating user profiles and generating personalised rankings using query logs. Their approach was to use a set of pre-defined interest categories and a K-nearest neighbour approach for clustering similar users. In this work we take a similar view that by reducing the dimensionality of the data we can get better results, however we use more principled techniques that do not rely on predefined categories but derive these from the data as part of the estimation process.

Teevan et al. [TDL08] investigated for what kinds of queries personalisation techniques most improved ranking performance. They found that how ambiguous a query is provides a good indication of how much benefit will be gained from personalisation. However for queries of low ambiguity (where all users tend to find the same results relevant) the personalisation can have a negative impact on performance. This work indicates that we must be careful when designing such systems to ensure that too much weight is not given to prior user preferences in deference to the unpersonalised document score.

2.3 Model-based Collaborative Filtering

Collaborative filtering systems can be placed in the context of information retrieval by considering that in a retrieval system items are "pulled" to users by the issuing of explicit search queries. Filtering systems on the other hand are described as "push" systems since they quite literally push those items at a user that they predict the user will like. Much early work was done in the 90s and the field has seen a resurgence of interest lately, primarily due to the Netflix prize [Kor08]. As mentioned in the first section, collaborative filtering algorithms can be generally classified into two distinct types: memory-based and model-based. Modern techniques tend to be model-based and are generally seen as a significant improvement over the older memory-based methods. In this work collaborative filtering algorithms are constructed via the model-based approach and therefore this survey of related work is restricted to similar methods.

In model-based collaborative filtering, typically the observed ratings are used to construct a model of the data by being decomposed into a sum of some biases. In the case of this work these include one for the user b_u, one for the item b_m and a third $b_{u,m}$; the joint bias caused by the interaction between user and item. More specifically it is surmised that the observed rating $r_{u,m}$ for an item m by a user u is a result of the mean rating over the entire data set μ perturbed by these three different factors plus some Gaussian error ϵ:

$$r_{u,m} = \mu + b_u + b_m + b_{u,m} + \epsilon \tag{2.1}$$

Examples of these models frequently use some form of dimensionality reduction to uncover latent factors and to calculate the joint bias $b_{u,m}$. These latent factors are constructed in a manner that best explains the training ratings and if we make the assumption that any further ratings will be drawn IID from the same distribution then the model should be able to predict new ratings well.

These model-based algorithms are able to overcome many of the scalability problems associated with the earlier, memory-based systems. This is particularly the case when real-time recommendations are required, which is obviously the most likely situation given the on-line nature of the systems where collaborative filtering is most often used. The most time-consuming task is the generation of the model itself, after which the task of new predictions is extremely quick due to the significant reduction in dimensionality afforded by the latent factors. With model-based systems the entire modelling operation can be completed off-line thus allowing for near-instantaneous real-time predictions as and when users need them.

2.3.1 Dimensionality Reduction

Many examples of this approach, including most attempts at the Netflix prize, use gradient descent algorithms to estimate a Singular Value Decomposition (SVD) of the original sparse ratings matrix [Pat07, SKKR00]. SVD is a technique derived from linear algebra used to represent a matrix A of real values as product of three simpler matrices usually denoted U, Σ and V, i.e. $A = U\Sigma V^T$. The columns of U are the left-singular vectors of the original matrix, the rows of V the right-singular vectors and Σ is a diagonal matrix of the singular values; essentially scaling factors for the singular vectors.

The resulting matrices are normally ordered so that the singular values in Σ are in descending order of relative importance where the first value in Σ, row in U and column in V represents the axis of greatest variance, the second being for the second greatest variance and so on. In its complete form the SVD of a matrix can be recombined into the exact original matrix with no loss of data. However, the rank of the matrices can be reduced to any number K, resulting in a reduced-dimensionality equivalent of the original matrix. This provides the best least-squares approximation of the original matrix and may uncover interesting relationships not easily discernible before the reduction in dimensionality.

If the matrix is complete (i.e. it is dense) then SVD has an analytic solution, however due to the incompleteness of the ratings matrix gradient descent methods are required to find a close approximation. The values computed for the SVD matrices are often regularised so as to prevent over-fitting and individual optimised biases for each user and item are commonly added to improve prediction performance. Goldberg et al. [GRGP01] instead apply a related technique called Principal Components Analysis (PCA).

2.3.2 Probabilistic Models

A large proportion of modern methods use probability theory to construct the models where observed ratings are assumed to arise from some latent variables which have to be estimated. In [Mar03], Marlin represents each user as a mixture of "attitudes" with each rating being generated by selecting one of these attitudes and then selecting the rating based on the ratings distribution for that attitude. Hofmann [Hof04] extends his earlier pLSI model to model ratings by again assuming that users have a distribution over "interests" or "attitudes" and that each rating is associated with a single interest drawn from the user's interest distribution. His work differs from that of Marlin [Mar03] however by then assuming that there is a rating distribution for each latent interest and item pair. So the observed rating is assumed conditional on both the latent interest of the user who rated the item and also on the item itself.

Other probabilistic approaches include [ZK07] in which the authors introduce a novel adaptation of the EM algorithm to learn the parameters of a prediction model for personalised content-based prediction. Stern et al. [SHG09] instead use Expectation Propagation and Variational Message Passing to learn a model using both ratings data and content. In other recent work Chen et al. [CCL$^+$09] compare the performance of Latent Dirichlet Allocation (LDA) [BNJ03] with association rule mining (ARM) for the purpose of community recommendation. This is a similar problem to rating prediction but instead involves the suggestion of online communities of interest rather than items. They show that LDA consistently outperforms ARM for this task, particularly when considering later recommendation. They also demonstrate that it is less likely to make extreme errors due to its Bayesian nature, certainly a useful property when recommending items. The next chapter discusses two probabilistic model-based collaborative filtering algorithms that can in some ways be seen as comparable to these models and draw on similar background theory. Blei [BNJ03] in fact uses collaborative filtering as an example of a problem for which LDA could be used and shows that it is able to outperform both probabilistic LSA [Hof01] and a simple unigram model.

2.4 Summary

This chapter has surveyed the previous work performed in these problem areas that motivates the later contributions of this thesis. In the next chapter I will present a short overview of the field of statistical machine learning, paying particular attention to unsupervised methods. Of particular interest are models involving latent variables which require the use of sophisticated sampling techniques required to work with. Furthermore the chapter will explain the Bayesian treatment of statistical inference and justify its application in learning the hidden semantics of social tagging data and for development of sophisticated collaborative filtering systems. The chapter will go on to describe two families of novel models that are used later in this thesis in a series of experiments with a view to solving these problems.

Chapter 3

Modelling Social Data

> "What we learn about is not nature itself, but nature exposed to our methods of questioning"
>
> *Werner Heisenberg*

This chapter provides a short introduction to the Bayesian methods of statistical modelling - in particular latent variable models - and how they can be used as powerful tools for modelling socially generated data. Before we can understand and use Bayesian modelling techniques we first have to understand what a probabilistic statistical model is, how one can be designed and how it can then be used. The chapter also briefly introduces the classical method of parameter estimation and contrasts this with the Bayesian treatment. In doing so, it suggests reasons why the Bayesian method is generally seen as being more principled and yields better results when inferring in real-world scenarios. Later the chapter describes more complex latent variable models and explains how their parameters can be estimated using Markov Chain Monte Carlo techniques. Finally latent topic models are introduced and then a series of novel Bayesian models designed for social data are described and derived. These models are used later in this thesis in a series of three experiments demonstrating their use and general applicability.

3.1 Data Modelling and Coin Tossing

Probabilistic data modelling refers to a process where we first construct a parameter-based model that we believe describes the outcomes of a set of experiments or data. These models are defined using one or more *probability distributions* that best explain the data which has been observed. Then using various mathematical and statistical methods we "fit" these parameters in such a way that the likelihood of the data we have actually observed being generated from that model is maximised. In a computer science setting this is frequently referred to as Machine Learning as we are in essence trying to get the machine to extract patterns and trends in order to gain a more complete understanding of what the data means [HTF08].

In classical (or frequentist) statistics, probabilities of events are interpreted as the frequencies of outcomes of an infinitely long-running experimental process. For example, in the simple - but very popular - example of flipping a coin the probability θ of the coin coming up heads be seen as simply the number of observations of heads divided by the total number of experiments, or in this case coin flips. This may seem quite intuitive and obvious but can actually be derived mathematically based on the underlying distribution and is known as a Maximum likelihood estimator (MLE). This process is modelled using a binomial distribution and a single coin flip can be modelled using a Bernoulli distribution where each coin flip is referred to as a single Bernoulli trial. The binomial distribution describes the probability of observing a number of "successes" (in this case, the number of heads) x in n experiments (coin flips) where the probability of success for each individual experiment is θ. The x and n values in this case are known as the sufficient statistics as they contain all the information required to describe the distribution and can be combined and written as the data D. The distribution can be written mathematically as the following function:

$$p(x; n, \theta) = \binom{n}{x} \theta^x (1-\theta)^{n-x} \tag{3.1}$$

The binomial distribution has a discrete and countable number of possible outcomes and is therefore known as a *discrete distribution*, the function above is known as a *probability mass function (PMF).* The first term in the function is the *binomial coefficient* and simply serves to ensure that the distribution over all possible outcomes sums to one, it is the remainder of the function that truly describes the form of the distribution. All probability functions have a similar form as it is always necessary that they sum to one. The mathematical form of this distribution is quite intuitive; it is simply the probability of success θ raised to the number of times a success is observed multiplied by the probability of failure (which must be $1 - \theta$) raised to the number of observed failures (which must be $n - x$). Distributions where the number of possible outcomes are not countable (because they can take on an infinite number of values) are known as *continuous distributions* and are described by *probability density functions (PDFs).* In this case the function does not describe the probability of a specific outcome value since that would always be zero but can be used to determine the probability of the outcome falling within a range of values by taking the integral of the PDF over a bounded interval.

Return to the problem of ML estimation, if we consider observed data $D = \{x_1, ..., x_n\}$ as being random independent, identically distributed (IID) draws[1] from some underlying and distribution F parameterised by θ, $x_i \sim F(\theta)$. Then we can calculate the *likelihood estimator* $\mathcal{L}(\theta|D)$ of some setting of parameter θ of this distribution conditioned on the observed data as follows:

$$\mathcal{L}(\theta|D) = \prod_{i=1}^{n} p(\theta|x_i) \tag{3.2}$$

If we then wish to calculate the best fitting value of this parameter given the data we find the value of θ that maximises the above likelihood estimator: $\hat{\theta} = \text{argmax}_\theta \ \mathcal{L}(\theta|D)$. Formally we say that the MLE produces the choice of parameters *most likely to have generated the observed data.* There are a number of techniques available for deriving the MLE for a given model, however generally speaking it can be obtained by calculating the derivative of the likelihood with respect to the parameter of interest and then solving the resulting

[1] Each draw is from the same distribution and draws are mutually independent (i.e. observing an outcome does not affect the probability of the next outcome).

equation when it equals zero. This is because finding the derivative and setting it to zero will return the value of the parameter at the stationary point of the distribution, i.e. where it is maximised. For reference the MLE for the binomial distribution, and therefore the most likely value for the parameter θ in this problem, is simply the number of observed heads divided by the total number of observations. For example if we toss the coin 20 times and observe 10 heads then the most likely value for the parameter θ is simply $\frac{10}{20}$, a half. In this coin flipping example the derivation of a Maximum likelihood estimator is simple and intuitive however this is not always the case.

3.1.1 Incorporating Prior Beliefs

The ML frequentist estimate is the best estimate we can make given the data we have been given, however it is not always very sensible. In many cases we are working with data which is quite sparse in nature and in some sense incomplete, we therefore may not have many observations from which to base our model. Imagine a case where we want to calculate θ, the probability of a coin giving heads, and so we flip a coin two times and observe two heads. The ML estimate would say that there is a 100% probability of this coin showing heads and conversely, a 0% probability of it returning tails. Do we honestly believe this is the case, or is it more likely that we simply have not observed the outcome of enough experiments to truly determine the most appropriate value for this parameter?

It is therefore practical to try to include some of our *prior* beliefs about the outcome of a coin toss. Our prior beliefs on the parameter θ can be incorporated in a distribution $p(\theta)$. In the coin tossing example we would use a binomial distribution for the likelihood function, $p(\theta|D)$, and could choose a beta distribution for the prior on θ, $p(\theta)$. The choice of the prior should represent our beliefs about the probability of observing heads *before* observing any data (coin tosses). The choice of a beta distribution is sensible in this case as it is conjugate to the binomial, therefore making computation more straightforward. Conjugacy is discussed in more detail in appendix A.

The beta distribution is a natural choice for a prior on the probability of observing heads for a coin toss experiment and has two parameters[2] α and β which essentially represent the prior number of observations we have made of heads and tails. If we strongly believe that the coin is fair and unbiased before observing any tosses then we could set our prior to be strongly peaked at $p(\theta = 0.5)$, for example by choosing a $beta(101, 101)$ distribution. In the coin flip setting this particular choice of hyperparmeters is analogous to saying that we have observed 200 coin flips before where 100 came up heads and 100 came up tails. However if we do not have strong prior beliefs and are happy to accept that the probability of heads is equally likely to be any value of θ from 0 to 1, we could choose a $beta(1, 1)$ distribution (this is the same as a uniform distribution where each outcome is equally likely).

In our coin tossing example, it is entirely possible that θ (the probability of heads) could be any value from 0 to 1 and we therefore can obtain the most pragmatic estimate if we use the entire $p(\theta|D)$ distribution. If we can achieve this then we are using all of the information about θ that we can get based on our observed data, without throwing any of it away. This is the overriding principle of the fully Bayesian method of estimation.

3.1.2 The Fully Bayesian Treatment

In Bayesian statistics, probabilities are interpreted as degrees of belief or measures of uncertainty rather than as being (essentially unknowable) parameters of some well defined, and ultimately deterministic, experiment. In many cases it is only really sensible to define probabilities as degrees of belief, rather than the average ratios of some repeatable process. For example consider the job of a juror; they are tasked with assessing whether or not the accused is guilty of committing the crime and so may need to ask, “given the evidence I have seen, what is the *probability* that the defendant is guilty?”. Clearly this is not an experiment that the juror can run many times to get the long-run outcome and therefore we must be able to consider a probability as a degree of belief in something.

In Bayesian statistics we always use a *prior* probability $p(\theta)$ in which we

[2] Note that parameters of priors are commonly referred to as hyperparameters.

encode our beliefs of the likely value of the parameter(s) before any data is actually observed. This $p(\theta)$ should crucially be some probability distribution and therefore will have a density over the range from 0 to 1. Given our *prior* distribution $p(\theta)$ and the likelihood of the observed data D conditioned on parameters θ ($p(D|\theta)$), we are concerned with generating a *posterior distribution* $p(\theta|D)$ over all possible values of the parameter of interest [BS94]. We can use Bayes' formula to construct this distribution as follows:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \tag{3.3}$$

Where $p(D)$, the *marginal probability* of the data, is the probability of witnessing the data D under all possible values of the parameter θ. It can be seen as a normalising constant ensuring that the posterior sums to 1 and can be calculated by *marginalising* over all possible values of the parameter θ:

$$p(D) = \int p(D,\theta)d\theta \ = \int p(D|\theta)p(\theta)\,d\theta \tag{3.4}$$

however if we are only interested in assessing each probability in this space in relation to the other probabilities we need not calculate the denominator. Since the $p(D)$ is not dependant on θ, the numerators will all be proportional to one another to within a constant. In this case we simply write:

$$p(\theta|D) \propto p(D|\theta)p(\theta) \tag{3.5}$$

Now that we have a posterior distribution over the parameter θ we can use this to estimate a "best fit" value for this parameter. We could, for example, follow the methods introduced earlier in the chapter and calculate the mode (maximal value) of the distribution $\hat{\theta} = \text{argmax}_\theta \ p(\theta|D)$. In doing so we would be calculating the "MAP" or Maximum a-Posteriori estimate and even though we are using a prior this is still a point estimate and therefore cannot be said to be a Bayesian estimate. To be fully Bayesian we must utilise all of the information from the distribution in a better way by calculating expectation of the distribution (mean value) given the posterior as follows:

$$\hat{\theta} = E[p(\theta|D)] = \int \theta \, p(\theta|D) \, d\theta \tag{3.6}$$

These concepts are perhaps best explained via the use of a simple example problem and a figure showing the various distribution involved and the parameter estimates obtained, this is shown in Figure 3.1. We will persevere with the coin example and see where it leads us. Let's say we have tossed a coin 10 times and observed a total of 7 heads and therefore there must have been 3 tails. The ML likelihood estimate for the parameter θ, the "fairness" of the coin is simply $\frac{7}{10} = 0.7$, quite intuitive but as we have only observed 10 coins tosses it may turn out that the real value of this parameter is very different. We can use the binomial distribution to visualise how likely each possible parameter is given the data we have observed, this is shown as the red dashed curve in Figure 3.1 and the ML estimate we calculated is the red dashed vertical line. Notice as described earlier in this chapter that this is also the stationary point of the derivative since the tangent line to the curve at this point will have a gradient of zero. Since we believe that our coin is quite fair we could choose to place a beta prior over the binomial, let's use $beta(11, 11)$, this distribution has a mean of 0.5 stipulating that we believe the coin to be fair and is shown by the blue dotted curve in the figure. Using Bayes' rule we can combine our prior and likelihood resulting in the posterior distribution shown by the black solid curve in the figure.

In Bayesian analysis we move from a prior distribution to a posterior by incorporating all of the information obtained from observations (the data). In cases where we have a lot of observations the data will "overwhelm" the prior (in the extreme case of infinite data we will essentially be left with the ML estimate), but will fill in where we do not have a lot of data. In fact the scientific method itself can be interpreted as simply an application of Bayesian inference [HU93]. Scientific theory involves updating probabilities about hypotheses conditional on any new observations or experiments; the more evidence we

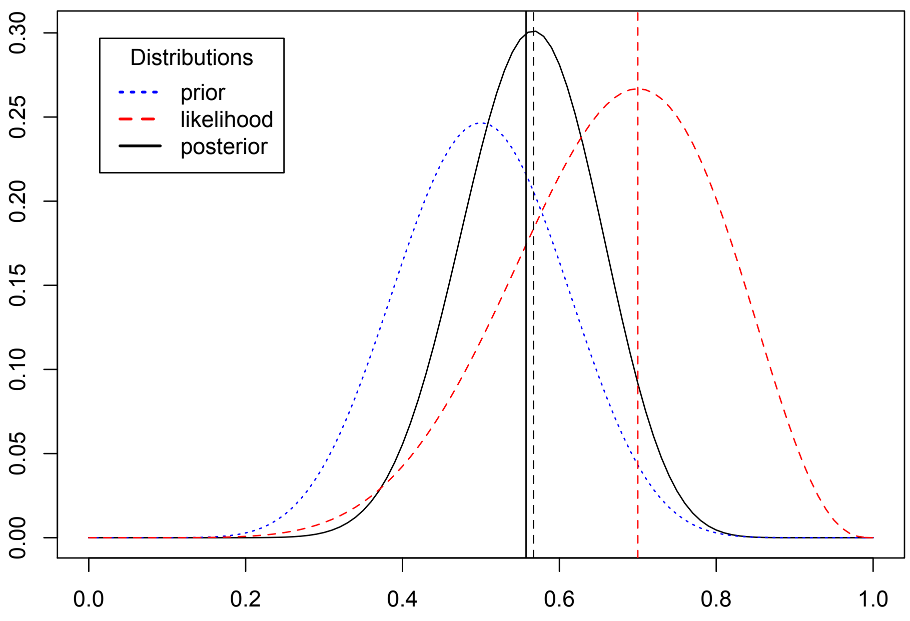


Figure 3.1: A simple example visualising Bayesian inference

have for something being true, the more we should believe that it is true.

We can see from the figure that the posterior is a compromise between the prior and the likelihood and that it has a smaller variance than either, this is a useful property and is not surprising given that the posterior combines the information from both sources, giving a better estimate. If we simply calculate the posterior mode we will get the MAP estimate, shown as a vertical dotted black line. We can get a more truly Bayesian estimate by calculating the expected value of the posterior (the mean) which in the figure is shown as a solid black line. Notice that the expected value is slightly less than the modal value because the posterior distribution is actually slightly positively skewed, a subtle but potentially important factor that is not incorporated in the MAP estimate. In this example the difference is not very significant, however in cases where the posterior is very complicated and multi-modal these two estimates can be very different and the benefits of an estimate over the whole distribution rather than a point estimate will be greater.

For this simple problem the posterior is quite easy to calculate and the result is easy to interpret however in many cases, particularly where latent variables are introduced, exact inference of the posterior is intractable. A number of techniques have been developed to approximate complex posteriors including Gibbs sampling [GS04] which is used for the work in this thesis and will be discussed in more detail later.

The entire process of Bayesian modelling can in fact be generalised into 3 main parts [GCSR04]:

1. Creating a *probability model* to describe the observed and unobserved quantities in a given problem. This model should be consistent with our understanding of the underlying problem and also the data collection method used.

2. Using Bayes' rule to estimate, and then carefully interpreting the *posterior distribution* - the conditional probability of the unobserved quantities give the observed quantities (the data).

3. *Evaluation of the final model* both in terms of statistical likelihood but also via empirical analysis and observation. How well does the model fit the data we have observed and does the output of the model "make sense?"

If we find by step 3 that the model we have selected does not fit the data observed particularly well or that inferences made based on the model do not appear sensible then we may wish to consider alternative models and follow through each of the 3 steps in the same manner. In many cases this results in an iterative process where a number of different models are estimated, evaluated and compared. It is worth noting that having a model with a good fit in terms of likelihood may not actually be the best for the problem at hand. It is therefore important to keep the original purpose of the work in mind and to evaluate the performance on the model by applying it to real data. Once we have our model in place we can use the posterior estimates to make inferences about unseen data, allowing for a number of useful statistical applications.

The Bayesian viewpoint of statistics has often been challenged due to its reliance of appropriate selection of a prior distribution, and in doing so introducing a certain amount of subjectivity into the application of probability. However this can be countered by the commonly held view that,

> You cannot learn in a vacuum, and cannot do inference without first making some assumptions.

The work described in this thesis follows the Bayesian viewpoint of probability in order to perform statistical inference as it benefits from a number of attractive properties, particularly when dealing with sparse and noisy data. I will discuss later the reason why I believe these features to be particularly advantageous when applied to real-world socially generated data, particularly when used to create complex latent variable models.

The experiment of flipping a coin serves as an excellent introduction to the key ideas behind statistical modelling however it is only natural that we want to proceed and use these techniques to model more interesting processes. We will now proceed by briefly explaining the concept of generative models and will show how latent (hidden) variables can be used in statistical inference.

3.2 Generative Models and Latent Variables

Earlier in this chapter we saw how it was possible to calculate the posterior distribution where we have a single conditional distribution. However, we can consider much more complex models where we are interested in working with a joint distribution of perhaps many conditionals or where the marginal distribution is highly complex. We are generally interested in considering models where there is a defined hierarchy of random variables with some structure of dependancy between them. In these cases our model may have a number of latent (or hidden) variables which are not actually observed and therefore must be inferred from the data.

A common and simple example is the case of heights in human populations, $p(h)$. The distribution of these heights would likely be bimodal[3] (having two peaks) where one peak is at approximately the mean height for a woman and another is at the mean height for a man. We prefer to model this complex

[3] Note that in practice it has been found that the distribution of human heights is not strictly bimodal. This is because a mixture of two Gaussians is only visibly bimodal if the difference between their means is less than two times their standard deviations. However for the purposes of this discussion we will assume that the distribution is clearly bimodal.

distribution as a mixture of two more simple distributions, where the complete marginal $p(h)$ is decomposed to become $p(h) = p(\theta_m)p(h|\theta_m) + p(\theta_f)p(h|\theta_f)$. Where $p(\theta_m)$ is the prior probability of a person being male, $p(\theta_f)$ is necessarily $1 - p(\theta_m)$, $p(h|\theta_m)$ and $p(h|\theta_f)$ are the distributions of heights for males and females respectively. These would be Gaussian distributions centred at the mean height for males and the mean height females and would each be parameterised by means μ_m and μ_f.

An example problem given this assumption might be that we have been given the heights of N people as our data and we wish to a) determine what the means of these Gaussians are and b) which class (gender) we expect each data point (person) to belong to. Since we have not been told the class of each data point in our data, we must introduce an auxiliary latent variable for each data point that indicates which class it belongs to.

For classification problems such as this there are two types of model that can be considered: discriminative and generative. In a generative model a full probabilistic model of all the variables is created from which a posterior can be estimated, yielding parameter estimates. On the other hand, discriminative models only provide estimates for the target variables conditioned on the observed training data. Generative models are generally able to deal better in situations of missing, partially labelled or entirely unlabelled data and are able to incrementally add new data and classes without needing to recalculate the entire model. In fact, discriminative models are inherently supervised and in many cases cannot be extended for unsupervised problems where no class labels are given in the training data [LB07]. These advantages are particularly beneficial for the kinds of problems addressed in this thesis and as a result generative models are used throughout. Blei also discusses the contrast between these model types in his thesis [Ble04] and also decides to use generative models for similar reasons.

In a generative model for this problem we can say that each observed data point x_i (the height of person i) was generated by the following process: first one of the two classes (male or female) is chosen at random based on the distribution over the two classes θ (which we would model using a binomial distribution) and is assigned to the variable z_i then the value of the data point

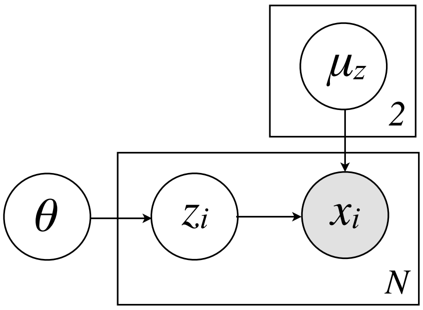


Figure 3.2: Graphical model for the distribution of human heights

is drawn at random from the class-conditional Gaussian distribution with mean μ_z. By constructing such a model we can see that parameter estimation can be loosely thought of as a reversal of this assumed generative process.

We can visualise these more complex hierarchies of dependency using graphical models, Figure 3.2 shows the graphical model for this problem. In these graphs each node represents a random variable, each edge a dependence between the random variables it connects and plates (squares) represent replication of the structure inside them. Each plate has a variable or number at its bottom-right corner denoting the number of replications of the encompassed structure, in this case the plate enclosing μ_z has cardinality two because they are two classes in our model. Random variables that are observed (the data) are shaded in and those which are unobserved (the latent variables) are unshaded. This technique for visualising graphical models is used to visualise the models described later in this thesis. Note that if we have been given the class of each data point then the z_i RVs would be observed and therefore shaded in and the parameter estimation procedure would be extremely simple.

3.2.1 Estimating a Posterior via Sampling

In cases where we have introduced unobserved latent variables we need to have some way to estimate the unknown parameter values, the posterior. If we choose to follow the fully-Bayesian route and place priors on the model parameters then this allows us to compute the entire posterior distribution. Unfortunately for many complex models, including those devised in this thesis, exact inference of the posterior distributions cannot be computed analytically in a sensible time frame and are therefore intractable. However a number of methods of approximating the posterior distribution exist including mean field variational inference [BNJ03] and Gibbs sampling [GS04]. This work makes use of Gibbs sampling methods to sample from the posterior which will be briefly described now.

Gibbs sampling is a Markov Chain Monte Carlo (MCMC) sampling method which allows us to work with the high-dimensionality probability distributions which typically arise in complex Bayesian models that we would otherwise be unable to construct or analyse. Monte Carlo techniques work by exploiting the observation that we do not necessarily have to be able to construct the full distribution of interest, we simply need to be able to draw samples from it that can then be used to form an empirical estimate. The Monte Carlo approach was originally devised by physicists to allow integration of very complex equations, for example let's say we wish to compute the integral of some complex function $f(x)$, i.e. we want:

$$\int f(x)\,dx \tag{3.7}$$

If this function is sufficiently complex that it cannot be integrated analytically then we may be able to use Monte Carlo techniques to approximate the integral. If we can decompose the integral into the product of a function $h(x)$ and some probability density function $p(x)$ then the following is true:

$$\int f(x)\,dx = \int h(x)p(x)\,dx = \mathbb{E}_{p(x)}[h(x)] \tag{3.8}$$

This shows that the integral can actually be expressed the expectation of $h(x)$ under the probability $p(x)$ and therefore if we can draw a large number N of samples at random from the distribution $p(x)$ then due to the law of large numbers the following holds true:

$$\mathbb{E}_{p(x)}[h(x)] \approx \frac{1}{N}\sum_{i=1}^{N} h(x_i) \tag{3.9}$$

To calculate the posterior distribution in a Bayesian model we need to be able to compute $\int p(y|x)p(x)\ dx$ and to accomplish this we can follow the principles of Monte Carlo sampling. If we draw many samples x_i from the prior $p(x)$ then the posterior can be approximated as follows:

$$p(x|y) \approx \frac{1}{N}\sum_{i=1}^{N} p(y|x_i) \tag{3.10}$$

Unfortunately in many cases (for example where we have introduced latent variables) this is not sufficient to estimate the posterior as we may not be able to directly calculate $p(y|x_i)$. Such problems necessitated the development of modern Markov Chain Monte Carlo methods. A Markov process describes random variables where the next state of the random variable is dependent only on its current state and a Markov Chain is simply a sequence of random variables generated from a Markov process. For example if we have a sequence of random variables where X_t denotes the value of X at time t then we can move from time t to $t+1$ via the following:

$$p(X_{t+1}|X_0, \ldots, X_t) = p(X_{t+1}|X_t) \tag{3.11}$$

Appropriately designed Markov Chains will eventually reach a stationary or equilibrium distribution and MCMC methods work by constructing a Markov Chain where the equilibrium distribution is (an approximation of) the distribution of interest, i.e. the posterior [GL97].

If we can construct such a Markov Chain then to sample from the posterior we simply have to run the chain until it has reached this state (when the chain is said to have converged) and then draw samples from the chain. These samples can be shown to be drawn from a close approximation of the posterior and can therefore be used to estimate parameter values of the model. The states of the chain before it has converged are known as “burn-in” and are simply discarded. When drawing from the converged chain it is common practise to only use every nth sample in order to prevent auto-correlation as samples close to each other in the chain are necessarily related, this is known as sample-lag.

In Gibbs sampling the next state in the chain is reached by sampling each

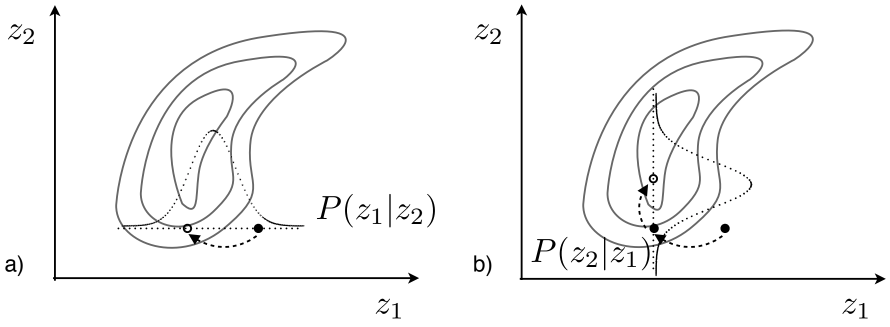


Figure 3.3: Gibbs sampling over latent variables

variable from its distribution when conditioned on the current values of all the other variables [GG84]. In the case of discrete latent variable models, each state of the Markov chain is an assignment of a class to each latent variable (i.e. to each z_i). Then for each latent variable in order a distribution is calculated over all of the possible latent classes conditioned on the current values of all the other latent variables $p(z_i|z_{-i})$ (where z_{-i} denotes all latent variables except for i). A value is then drawn at random from this distribution and this value is then allocated to z_i. This routine proceeds in order until all latent variables have been updated, this is said to be a complete "sweep" and is a single iteration of the Gibbs sampler.

The process can be more easily understood by referring to Figure 3.3 which shows the process for a simple example where there are only two latent variables. In Figure 3.3a we calculate the distribution of variable z_1 conditioned on the values of all the other latent variables which in this case is simply z_2, shown as a dotted line. A value for z_1 is then sampled from this distribution and the sampler proceeds to carry out the same process for z_2, shown in Figure 3.3b. Note that the conditionals can be any distribution, however for illustration purposes in this case we have chosen to show Gaussians. In many cases, particularly in Information Retrieval applications where we are dealing with textual data, these distributions will be discrete such as the multinomial distribution.

Gibbs sampling is a preferable alternative to methods such as Expectation Maximisation as it works by sampling from the entire posterior distribution rather than attempting to locate a stationary point and is therefore unlikely to get "stuck" in local maxima and does not require the use of additional machinery such as simulated annealing to get around this problem [SR93]. Furthermore Griffiths and Steyvers found that for large-scale hidden variables problems such as LDA that Gibbs sampling provides better performance in terms of convergence times than comparable algorithms [GS04].

In fact Gibbs sampling can be seen as a stochastic equivalent of the EM algorithm where the expectation and maximisation steps are replaced with sampling [Wal04]. Another benefit of this technique is the ability to quickly and easily "fold-in" new data into the model. To include this new data into the model we can simply run the Gibbs sampler over any new data, holding all of the pre-existing latent variables from previous runs of the sampler fixed. After the sampler has converged on this new data we can simply recalculate any parameter estimates we require for our model. Convergence on this new data usually occurs within less than 50 iterations, far less than required to sample an entire new model as it can leverage information from the already inferenced variables to more quickly hone in on the posterior.

Gibbs sampling is used extensively to calculate the posterior distributions of the models described later in the thesis. The reader is referred to [GL97] for more detailed information on MCMC methods including Gibbs sampling. The next section will briefly introduce Latent Dirichlet Allocation (LDA), a common and extremely popular example of a latent variable model used to uncover hidden topics present in document collections. LDA serves as a good starting point for understanding the novel models introduced later in this thesis.

3.2.2 Topic Models and Latent Dirichlet Allocation

Topic models attempt to probabilistically uncover the underlying semantic structure of a collection of resources based on analysis of only the vocabulary words present in each resource. This latent structure is modelled over a number of "topics" or dimensions which are assumed to be present in the collection. These topics are represented in the model as latent variables and each word position is assigned to one of these topics (like the two classes in the human height example). Since the number of topics chosen is generally much less than the dimensionality of the original data points such models provide a form of dimensionality reduction. This is of significant benefit as not only can it uncover hidden clusters in the data but can also markedly reduce the size of the model. This is similar to Singular Value Decomposition (SVD) in linear algebra which has been used in the past for similar problems in the field of Information Retrieval [DDF$^+$90] and as we shall see shortly, its output can be compared to that of SVD. However due to its probabilistic and generative nature this new approach to the problem is far more adaptable, principled and provides a more readily interpretable output.

In most cases the number of topics to use are chosen *a priori*, however recent work has investigated how this value might be inferred automatically based on the observed data, the most appropriate example for this work being Dirichlet Processes [TJBB06]. These processes add significant further complexity and as such it is generally acceptable to use empirical methods to choose the most optimal parameterisation. Topic models have been used for various problems including analysis of scientific papers [BL07], library books [MM07] and even text-based image retrieval [BJ03]. Examples of such models include probabilistic Latent Semantic Analysis (pLSA) [Hof01] which attempted to form a probabilistic interpretation of SVD and Latent Dirichlet Allocation (LDA) [BNJ03] which can be seen as a Bayesian interpretation of pLSA. LDA serves as an excellent starting point for building more complex models for example in [Wal06].

Figure 3.4 shows a graphical model diagram for LDA. Notice that it is not the "standard" LDA diagram in which a second plate is drawn to represent all of the samples (words and topics) from the same document. Instead, and equivalently in terms of the generative process, this representation introduces an observed variable d_i denoting the corresponding document ID for each word w_i in the corpus. This notation is used to facilitate easier comparison between the LDA model and the newer models introduced later. Note that for all of the models discussed in this thesis the subscript i always refers to a unique word position in the corpus.

LDA represents documents as random mixtures over latent topics which themselves are random mixtures over observed words in the vocabulary. So

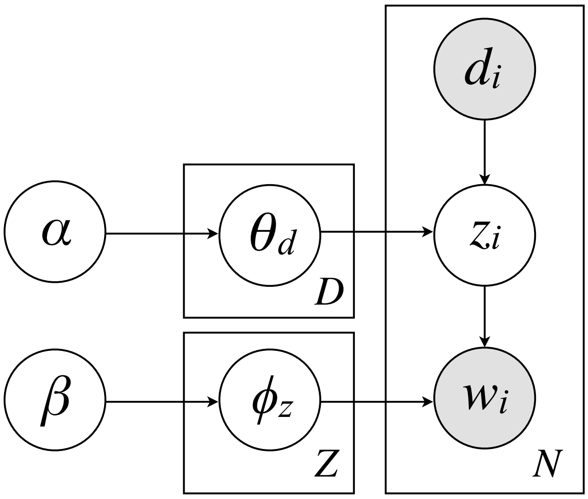


Figure 3.4: An alternate graphical model for Latent Dirichlet Allocation (LDA)

each document in the model is represented as a distribution over latent topics $z \in Z$ and has a distribution $p(z|d)$ and each topic is a distribution over words $p(w|z)$. The original data can be represented as a very large matrix of size $D \times W$ where each cell $i_{d,w}$ stores the count of word w in document d.

The model possesses a number of advantageous attributes; it is fully generative meaning that it is easy to make inferences on new documents or terms and overcomes the over-fitting problem present in models such as pLSI [Hof01]. Also since in LDA each document is a mixture over latent topics it is far more flexible than models that assume each document is only drawn from a single topic. The generative process for LDA can be described as follows:

1. For each document d a distribution $p(z|d)$ is drawn from $Dirichlet(\alpha)$

2. For each word w a distribution $p(w|z)$ is drawn from $Dirichlet(\beta)$

3. For each observed word position i

   (a) a topic allocation z_i is randomly chosen from the topical distribution $p(z|d_i)$ of the document the word position belongs to d_i

Symbol	Description
D	number of resources/documents
Z	number of topics
N	number of unique word positions
U	number of users
d_i	resource for word at position i
z_i	topic allocation at position i
w_i	lexical term (word/tag) at position i
u_i	user who contributed tag at position i
θ_d	distribution over topics for resource d
ϕ_z	distribution over words for topic z
ψ_u	distribution over topics for user u
θ_z	distribution over resources for topic z (TTM2 only)
α	Dirichlet prior over Θ
β	Dirichlet prior over Φ
γ	Dirichlet prior over Ψ

Table 3.1: List of notation for LDA and Tagging Topic Models

(b) a single word w_i is drawn from that topic's distribution over words $p(w|z_i)$

Note that, as with the vast majority of language models, the words are assumed to be independent and therefore the presence of one word does not effect the likelihood of observing another word. This simplifying assumption is also known as the "bag of words" model as it does not take word order or grammar into account. Also note that to truly make this generative model complete some distribution would be required from which to sample the lengths of the documents. In his paper [BNJ03] Blei uses a Poisson distribution to model this process but notes that this may not be entirely appropriate as it cannot accurately fit the distributions of documents lengths found in real-world datasets. In any case this is not necessary for model estimation purposes since the document lengths are known and it is therefore omitted from the following discussion of the models.

LDA is based around two parameters which are represented as two matrices

Φ and Θ containing estimates for the probability of a word given a topic $p(w|z)$ and a topic given a document $p(z|d)$. Thus each column of the respective matrices contains (estimates for) a probability distribution over words for a particular topic and over topics for a particular document, denoted ϕ_z and θ_d respectively. These two matrices can be compared to the U and V matrices derived from an SVD of the original tag data matrix, however in the case of LDA there is no requirement for a separate diagonal matrix Σ of scaling factors. In order to prevent over-fitting the data, LDA places a symmetric Dirichlet prior on both these distributions, resulting in the following expectations for the parameter values under the respective posterior distributions $p(\phi_z|\mathbf{w}, \mathbf{z})$ and $p(\theta_d|\mathbf{z}, \mathbf{d})$, where $\mathbf{w}$ is the vector of words occurrences w_i in the corpus, $\mathbf{z}$ is an assignment of topics to each word position z_i and $\mathbf{d}$ is the vector of documents d_i associated with each word position. Therefore given a complete set of topic allocations the parameters can be estimated in the following manner:

$$\hat{\phi}_{w|z} = \frac{N_{w,z} + \beta\frac{1}{W}}{N_z + \beta} \tag{3.12}$$

$$\hat{\theta}_{z|d} = \frac{N_{z,d} + \alpha\frac{1}{Z}}{N_d + \alpha} \tag{3.13}$$

Here $N_{w,z}$, $N_{z,d}$ and N_z are counts denoting the number of times the topic z appears (in $\mathbf{z}$) together with the word w, with the document d, and in total, respectively. W is the vocabulary size and Z is the number of topics. The hyperparameters α and β essentially act as a pseudo count indicating a relation to smoothing in language models. This allows the model to fall back on the priors in the event of sparse data.

As outlined earlier in this chapter, when designing and implementing any statistical model it is necessary to choose how to represent the data within the model in terms of probability distributions. The choice of distributions for LDA is fairly straightforward; both the words and the topic allocations are multinomial distributions which themselves are drawn from Dirichlet distributions. The multinomial distribution is a generalisation of the binomial discussed earlier to any number of dimensions. So for example in the case of the observed words in the corpus, they can be described as being distributed multinomial where the multinomial in question has dimensionality V, the size of the vocabulary. Each time a single term is used this can be seen as being a "success" for that word, in much the same way that a coin flip returning heads is a success. The Dirichlet distribution is the beta distribution generalised to any number of dimensions and describes a probability distribution over multinomials. Not surprisingly this combination of a multinomial distribution with a Dirichlet prior is a direct extension of the beta-binomial model discussed earlier generalised to an arbitrary number of dimensions. Both the multinomial and Dirichlet are described in more detail in Appendix A.

To estimate these parameter values we need to determine the topic allocation z_i for each word position w_i and to achieve this we can use Gibbs sampling. Each state of the Markov chain is (in this case) an assignment of a discrete topic (from 1 to Z) to each z_i, i.e. to each observed word in the corpus. The Gibbs sampling procedure for LDA involves iteratively updating the assignment of each topic z_i in the topic vector $\mathbf{z}$ by sampling a value from the distribution $p(z_i|\mathbf{w}, \mathbf{z}_{-i}, \mathbf{d})$, which is conditioned on the current assignment to all topic variables except z_i. (As before, the vector $\mathbf{z}_{-i}$ denotes all topic assignments except z_i.) In LDA the word assignment is conditionally independent of the document given the topic assignment:

$$p(z_i|w_i, d_i) \quad = \quad \frac{p(z_i, w_i|d_i)}{p(w_i|d_i)} \propto p(w_i|z_i)p(z_i|d_i) \tag{3.14}$$

In order to sample a topic allocation for each word position we need to be able to calculate the full conditional posterior distribution $p(z_i|\mathbf{w}, \mathbf{z}_{-i}, \mathbf{d})$. However, Gibbs sampling only requires that this be a function that is proportional to the true probability and therefore the expected value for this conditional distribution can be derived as follows:

$$\begin{aligned} \mathbf{E}[p(z_i|\mathbf{w}, \mathbf{z}_{-i}, \mathbf{d})] \quad &\propto \quad \hat{\phi}_{w_i|z_i}\hat{\theta}_{z_i|d_i} && (3.15) \\ &\propto \quad \frac{N^{-i}_{w_i,z} + \beta\frac{1}{W}}{N^{-i}_{z} + \beta}\frac{N^{-i}_{d_i,z} + \alpha\frac{1}{Z}}{N^{-i}_{d_i} + \alpha} && (3.16) \end{aligned}$$

The estimates $\hat{\phi}_{w|z}$ and $\hat{\theta}_{z|d}$ are calculated over $\mathbf{z}_{-i}$ rather than $\mathbf{z}$. So $\mathbf{z}_{-i}$

denotes the assignment of topics to all word positions (except the current topic z_i). In the full derivation $N_{w_i,z}^{-i}$ is the number of times word w_i is assigned to topic z and N_z^{-i} is the total number of words assigned to topic z (both excluding the current position, z_i). $N_{d_i,z}^{-i}$ is the number of times topic z occurs in resource d_i (excluding z_i) and $N_{d_i}^{-i}$ is the total number of words in resource d_i (less 1).

After sufficient iterations of the sampler, the Markov chain converges and the parameters of the LDA model can then be estimated from $\mathbf{z}$ using the estimators outlined in Equations 3.12 and 3.13. It can be assumed that the chain has converged when there is minimal change in the observed model likelihood over successive samples, in the case of LDA the likelihood is:

$$p(\mathbf{w}, \mathbf{z}|\Phi, \Theta) = \prod_i \sum_z \hat{\phi}_{w_i|z} \hat{\theta}_{z|d_i} \tag{3.17}$$

For increased accuracy, and to ensure that the resulting parameter values are sampled from a large proportion of the posterior, estimates can be averaged over consecutive complete samples of $\mathbf{z}$ from the Markov chain.

Note that in terms of implementation it is often preferable to represent the document data by using a more compact form of a “ragged” (non-rectangular) 2D array where each cell $w_{d,i}$ is now the word at position i of document d where the length of the 1D array w_d is the length of d. This representation has the added benefit of making the Gibbs sampling procedure easier as it simply needs to iterate over this array. Furthermore we can also represent the array of topic allocations in the same form where $z_{d,i}$ is the current topical allocation to word position i of document d. The Gibbs sampling method allows for a very compact and memory-efficient algorithm. It suffices to simply keep the word and topic arrays $\mathbf{w}$ and $\mathbf{z}$ and the counts of topics over documents and over words in memory.

3.3 Tagging Topic Models

The work in this thesis is concerned with the use of topic models to better understand and utilise the vast amounts of information available from social web sites. However, to use such models we must first have some form of

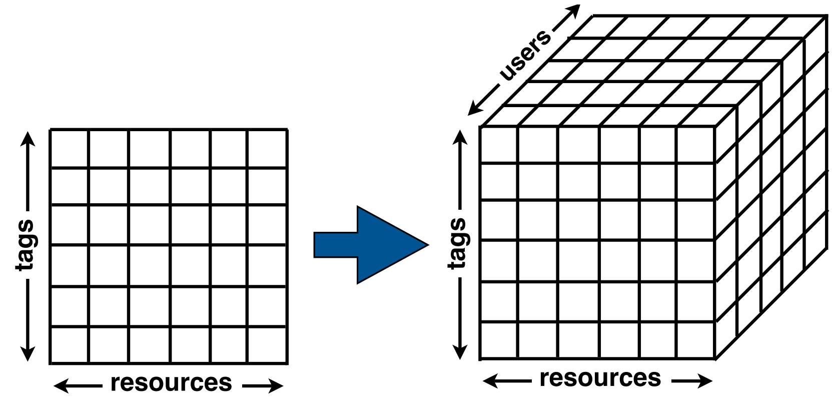


Figure 3.5: The problem of adding an extra dimension

document data from which to estimate our distributions. In the case of social systems where we have a small number of tags per user for some resources we can simply conflate all users' tags together to form a single "document" describing the resource. Doing so can potentially exploit the group consensus formed for more popular resources since tags chosen by multiple users will be counted in the model multiple times. In modelling this data we must also consider that we have another potential source of information: the user who submitted each annotation. The tagging topic models presented now and the later models for collaborative filtering are able to include this extra data to improve the accuracy of the estimations and to allow for modelling of user interests over the topic space.

Recall that social tagging data consists of 3 distinct entities: the resource being tagged, the user who tagged the resource and the tag itself. This is modelled as a tripartite graph with 3 disjoint sets of nodes: resources $\mathcal{D} = \{\mathtt{d}_1, \ldots, \mathtt{d}_D\}$, users $\mathcal{U} = \{\mathtt{u}_1, \ldots, \mathtt{u}_U\}$ and tags $\mathcal{W} = \{\mathtt{w}_1, \ldots, \mathtt{w}_V\}$[4] In this graph the edges between these nodes represent the individual annotations; a user $\mathtt{u}$ annotating resource $\mathtt{d}$ with tag $\mathtt{w}$. Each assignment of a tag to a resource by a user - each edge - is denoted as the relation $\mathcal{Y}$ and is typically called a tag assignment (*tas* for short). Therefore the complete folksonomy is actually

[4] Note that in order to remain in keeping with the notation from topic modelling literature we use the character d to denote resources and that for all intents and purposes the words *documents* and *resources* are interchangeable.

a quadruple $\mathcal{F} := (\mathcal{U}, \mathcal{W}, \mathcal{D}, \mathcal{Y})$. This data can be represented as a tensor as shown in Figure 3.5.

In [WZY06b] it is noted that tags are usually semantically related if they are used to describe the same resources many times. Correspondingly, resources are similar if they are annotated with the same tags and users share similar interests if their annotations share many related tags. These relationships can be mapped onto a conceptual space of Z dimensions (or in the topic modelling case, topics), that represent categories of knowledge. In this representation, each entity's component on a given dimension gives a measure of how similar or related it is to that category. This provides a framework for the discovery of meaningful relationships between entities and for reducing the dimensionality of the problem down as $Z \ll W$. To fully leverage social data we need to move from the two-dimensional matrix of LDA to a 3 dimensional tensor, $\mathbb{N}^{\mathcal{D} \times \mathcal{W}}$ to $\mathbb{N}^{\mathcal{D} \times \mathcal{U} \times \mathcal{W}}$. In using latent topics the dimensionality of this tensor is reduced down to $\mathbb{N}^{\mathcal{D} \times \mathcal{U} \times \mathcal{Z}}$.

The prime motivation for using topical models for social annotation data is that this intelligent reduction in dimensionality will deal with many of the polysemy and synonymy issues present. As we shall see, they provide a means in IR to match resources with user queries on a semantic meaning level, rather than purely by lexical co-occurence as similarity can be discerned based over several levels of co-occurence. Furthermore applying such models to social tagging data does not present the same issues of information loss due to the "bag of words" assumption as when they are applied to "normal" documents. This is because social annotations do not have any meaningful notion of word order, they are quite literally a bag of words with no grammatical structure possible. As a result, applying such a simplification to this data does not result in a loss as it would do were the tag data structured grammatically in some manner. Also words co-occurring in documents tend to only be strongly related in a local scope (in the same sentence or paragraph) and only very generally related over the document as a whole. While in the case of social tagging data all tags in a given social annotation can be assumed to be much more strongly related.

As noted in chapter 2 the vast majority of research involving the modelling

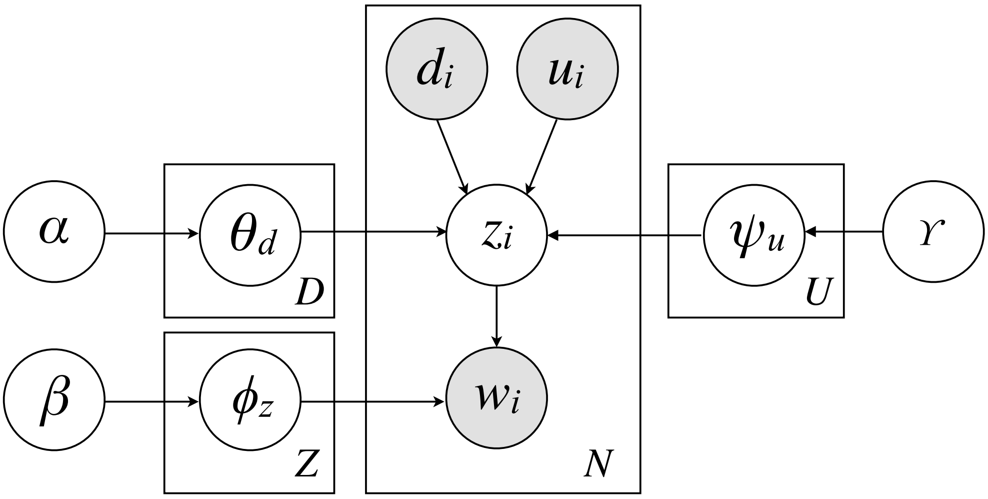


Figure 3.6: Graphical model of Tagging Topic Model 1 (TTM1)

of social tagging systems re-use mostly unadapted techniques and models from information retrieval and collaborative filtering. As we have seen, the structure and statistical properties of tagging data are very different to more traditional documents and therefore it is necessary to consider new models and techniques that are more thoroughly adapted to the data.

3.3.1 Tagging Topic Model 1 (TTM1)

In attempting to modify LDA to include user preferences the first, most natural step to take is to change the Θ matrix from representing the $p(z|d)$ to the $p(z|d,u)$; i.e. the joint probability of topic z given both resource d and user u. This new representation of users and resources over topics is a large, extremely sparse, 3D tensor $\in \mathbb{N}^{\mathcal{D}\times\mathcal{U}\times\mathcal{Z}}$. While this tensor is significantly smaller than the tensor representing the original data due to the dimensionality reduction over the topic space, the sheer size and inherent sparsity of this distribution still presents significant problems. Particularly due to the increased danger of over-fitting and the considerable amount of time required to fully sample the conditional distribution, not to mention the increased memory capacity required to work with it. Consider that for many combinations of users, resources and topics we will still have little or no information available from the corpus. As a result, for the majority of cases the estimate would be reduced to the symmetric un-informative prior over the distribution.

A solution to this problem is to make the simplifying assumption that the probability of a user and a resource are independent given a topic allocation. That is for each position in the corpus the probability of a topic given the resource the tag is assigned to is independent of the probability of the topic given the user who assigned the tag. The tensor is therefore split into a pair of two-dimensional matrices Θ, representing the $p(z|d)$ - as in LDA - and a new set of parameters Ψ, the $p(z|u)$ or probability over the topic space for each user. Recall that for the Gibbs sampling algorithm to operate we require a method of calculating the full posterior distribution, or at least a function proportional to it. To do this we first need a way to calculate the value of $p(z|\theta_d, \psi_u)$ which we can then use in our Gibbs sampling algorithm. Via direct application of probability theory and based on the assumptions stated earlier, the probability of a single topic assignment z given θ_d and ψ_u is:

$$\begin{aligned} p(z|\theta_d, \psi_u) &= \frac{p(z)p(\theta_d, \psi_u|z)}{p(\theta_d, \psi_u)} = \frac{p(z)p(\theta_d|z)p(\psi_u|z)}{p(\theta_d, \psi_u)} && (3.18) \\ &= \frac{p(z)[\frac{p(\theta_d)p(z|\theta_d)}{p(z)}][\frac{p(\psi_u)p(z|\psi_u)}{p(z)}]}{p(\theta_d, \psi_u)} \propto \frac{p(z|\theta_d)p(z|\psi_u)}{p(z)} && (3.19) \end{aligned}$$

In order to keep the model fully Bayesian a prior distribution γ can be placed over the user-topic distributions ψ_u. This gives the user-topic distribution a similar role to play in the generative story as the document-topic distribution. Therefore the prior in this case will also be Dirichlet meaning that the distribution ψ_u is assumed to be drawn from a symmetric Dirichlet parameterised by γ. This gives the following parameter estimation under the posterior distribution $p(\psi_u|\mathbf{z}, \mathbf{u})$:

$$\hat{\psi}_{z|u} = \frac{N_{z,u} + \gamma\frac{1}{Z}}{N_u + \gamma} \tag{3.20}$$

where $N_{z,u}$ and N_u are counts of the number of times the topic assignment z appears in annotations made by user u and N_u is the total number of annotations made by u, respectively. Examples of the Θ and Φ matrices outputted by a TTM model are shown below:

$$\Theta_{p(z|d)} = \begin{pmatrix} 0.33 & 0.1 & 0.4 \\ 0.33 & 0.8 & 0.1 \\ 0.33 & 0.1 & 0.5 \end{pmatrix} \qquad \Phi_{p(w|z)} = \begin{pmatrix} 0.2 & 0.9 & 0.2 \\ 0.3 & 0.05 & 0.5 \\ 0.3 & 0.05 & 0.3 \end{pmatrix}$$

In this example there are only 3 topics, 3 documents, 3 users and 3 words. The Θ matrix indicates that document 1 has a completely uniform distribution over the 3 latent topics, whereas document 2 draws predominantly from topic number 2. The first column of the Φ matrix shows that almost all words drawn from topic 2 will be word 1, with the other 2 words in the lexicon only having a probability of 0.05, which is likely to simply be the model falling back on the priors.

Intuitively we can therefore expect that document 2 is composed almost entirely of word 1, since it has a high probability of drawing from topic 2 and word 1 has a high probability of occurrence, given that one is drawing from topic 2. Note that the form of the Ψ matrix is similar to that of the Θ matrix, however each column instead represents a user rather than a document. Note also that in order to be valid probabilities, the columns of the matrices much sum to unity, however there is no requirement for the rows to also do so.

Since the estimate of the probability of a word given a topic has not changed in order to derive the complete Gibbs sampling equation $p(z|d)$ from LDA is replaced with new the joint estimate $p(z|\theta_d, \psi_u)$ derived above. The Gibbs sampling procedure of probability of a topic assignment z at position i in this model is therefore:

$$p(z_i|w_i, d_i, u_i) = \frac{p(z_i, w_i, u_i|d_i)}{p(w_i, u_i|d_i)} \propto p(w_i|z_i)\frac{p(z_i|d_i)p(z_i|u_i)}{p(z_i)} \tag{3.21}$$

$$\tag{3.22}$$

Thus the expected value for the conditional distribution can now be estimated as:

$$\mathbf{E}[p(z_i|\ldots)] \propto \hat{\phi}_{w_i|z_i}\frac{\hat{\theta}_{z_i|d_i}\hat{\psi}_{z_i|u_i}}{\hat{p(z)}} \tag{3.23}$$

$$\propto \frac{N^{(-i)}_{w_i,z}+\beta\frac{1}{W}}{N^{(-i)}_{z}+\beta}\left(\frac{N^{(-i)}_{d_i,z}+\alpha\frac{1}{Z}}{N^{(-i)}_{d_i}+\alpha}\frac{N^{(-i)}_{u_i,z}+\gamma\frac{1}{Z}}{N^{(-i)}_{u_i}+\gamma}\right)\hat{p(z)} \tag{3.24}$$

$\hat{p(z)}$ can be simply estimated as N_z/N (less the current topic allocation z_i), however this estimate could also be smoothed via the application of a Dirichlet prior. Also, since they are independent of the topic and are therefore constants, the denominators in the resource and user topic estimates can be removed from the calculation without affecting the sampling process. As with the more compact representation of LDA discussed previously the user data can also be presented as a ragged 2D array where each cell $u_{d,i}$ represents the user who assigned the tag as position i to resource d.

The complete generative model is shown in Figure 3.6. Notice that the model is extremely flexible on a per-resource-description level as each description is modelled over the entire topic space, this is also true for each user. Also for each annotation a three-layer generative process is performed with the topical distribution for each resource being sampled multiple times (once for each tag). The model, however, suffers from the fact that it is not entirely obvious what generative process could have produced such an output. This observation gives rise to the development of a second model which still models the entire folksonomy and makes similar independence assumptions, but also possesses a more intuitive generative structure.

3.3.2 Tagging Topic Model 2 (TTM2)

The previous section described TTM1 and indicated that this model's generative story (i.e. how we imagine that the data were originally generated) is a little unclear and does not intuitively fit with how we might expect social annotations to be generated. In both LDA and TTM1 it is assumed that each document in the collection "chooses" its own topical distribution θ_d, leading to an assignment of word positions in the document to topics based on this

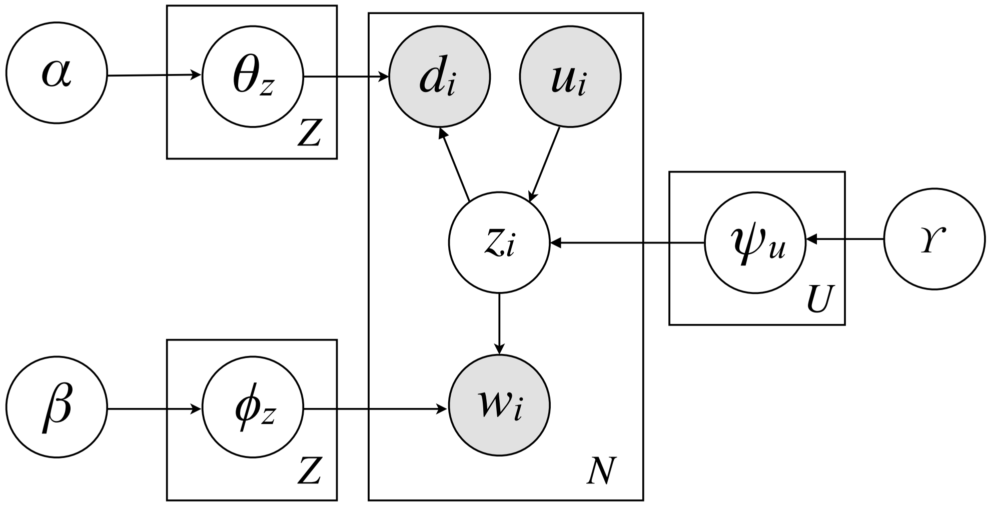


Figure 3.7: Graphical model of Tagging Topic Model 2 (TTM2)

distribution. In the case of TTM1 this is somehow also related to the user's topical distribution, however it is not clear exactly what this relationship may be.

Such a generative story for documents fits in well in a normal information retrieval setting where we are indexing the actual content of documents. However with social tagging data we are not using the content of the documents as features but rather the words (tags) chosen to describe resources by users. Therefore we propose an alternative model, shown in Figure 3.7, where the resource is chosen by the topic rather than the other way round. In this model the generative story for each individual word position i can be described as the following:

1. For each word position i, a topic allocation z_i is randomly chosen from user u's topical distribution $p(z|u_i)$
2. A relevant resource is drawn randomly from topic z_i's document distribution $p(d|z_i)$
3. Finally, a tag w_i to describe the resource is drawn from topic z_i's tags distribution $p(w|z_i)$

This generative story seems to be intuitively a better fit for annotations as the user initially chooses a topic (or topics) she is interested in and then based on those topics will find resources to bookmark and annotate. As the tags are a user's description of the resource we can further assert that the tags will be chosen from the same topical distribution as the document they describe. In this model Θ now contains probability estimates of the form $p(d|z)$ and each column θ_z is a probability distribution over resources (documents) for a particular topic. The expected value of these parameters under the posterior are calculated as follows:

$$\hat{\theta}_{d|z} = \frac{N_{z,d} + \alpha\frac{1}{D}}{N_z + \alpha} \tag{3.25}$$

Given this new parameterisation, the probability of a topic assignment z at position i in this model can be factorised much more cleanly as:

$$\begin{align} p(z_i|w_i, d_i, u_i) &= \frac{p(z_i, w_i, u_i|d_i)}{p(w_i, u_i|d_i)} \propto p(w_i|z_i)p(d_i|z_i)p(z_i|u_i) \tag{3.26} \\ & \tag{3.27} \end{align}$$

Finally the expected value for the conditional distribution is:

$$\begin{align} \mathbf{E}[p(z_i|\mathbf{w}, \mathbf{z}_{-i}, \mathbf{d}, \mathbf{u})] &\propto \hat{\phi}_{w_i|z_i}\hat{\theta}_{d_i|z_i}\hat{\psi}_{z_i|u_i} \tag{3.28} \\ &\propto \frac{N_{w_i,z}^{(-i)} + \beta\frac{1}{W}}{N_z^{(-i)} + \beta}\frac{N_{d_i,z}^{(-i)} + \alpha\frac{1}{D}}{N_z^{(-i)} + \alpha}\frac{N_{u_i,z}^{(-i)} + \gamma\frac{1}{Z}}{N_{u_i}^{(-i)} + \gamma} \tag{3.29} \end{align}$$

In this model only the denominator in the user-topic estimate can be safely removed from the calculation but it is included here for completeness.

For both of these models, the resulting reduced-dimensionality distributions over the complete folksonomy can then be used to uncover relationships between users, tags and resources and therefore make useful inferences about new data. Given that LDA can be said to be a Bayesian equivalent of the Singular Value Decomposition of a two-dimensional matrix, these models can be described as being analogous Bayesian equivalents of 3 dimensional tensor factorisation. These new models demonstrate how existing probabilistic models can be scaled up to provide useful inferential machinery in domains involving multiple levels of structure. However they also demonstrate that some care needs to be taken when choosing how to represent and model this new data. Later in this thesis two different uses of these models within social web sites are described and used to demonstrate their effectiveness for these tasks. For more information please refer to Appendix B where the TTM2 model is derived mathematically from its joint likelihood and from this the Gibbs sampling routine outlined above is also derived.

Note that during the initial investigation of models for this thesis an Expectation Maximisation-based alternative to Gibbs sampling was investigated for the TTM1 model. This model was very similar to the model presented by Wu et al. [WZY06b]. However in testing, it was found to be significantly slower than the Gibbs sampling version and had to be run several times in order to obtain a model with good fit as it tended to easily get stuck in local maxima. Due to the size of the datasets and the range of latent topics tested in the experiments presented later in this thesis, it would not have been a feasible alternative. In addition, perhaps because of its non-Bayesian nature, it was found to not be well suited to the sparse data typically found in social tagging systems.

The next section considers the more complex problem of adapting the topic modelling paradigm to collaborative filtering data where it is important to consider continuous distributions within the models, significantly deviating from the models described previously. This work also requires novel integration of Gibbs sampling with fixed-point optimisation to create a cohesive and powerful representation of ratings data. The next section details the choices required to develop latent variable models that are more suitable for ratings data and also are able to estimate an “interest” distribution for each user allowing for personalised predictions to be made.

3.4 Models for Collaborative Filtering Data

In order to implement a new latent variable model appropriate for collaborative filtering data it is necessary to choose how best to represent the latent factors and how to incorporate the ratings data, both in terms of their statis-

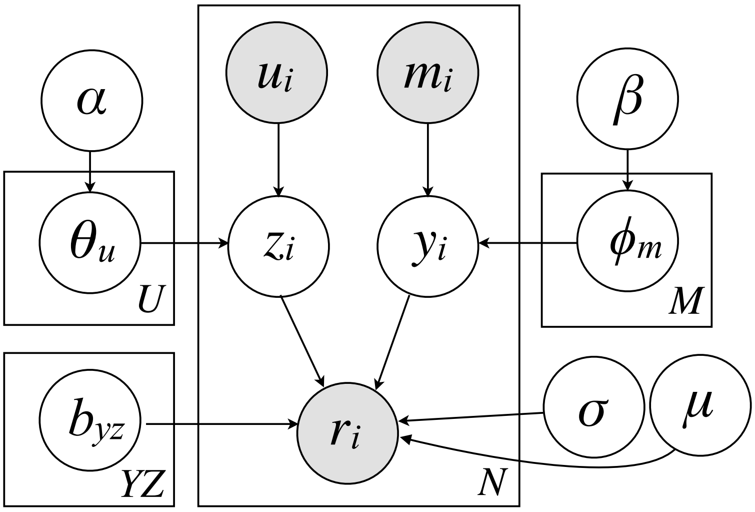


Figure 3.8: Latent Interest and Topic Ratings Model 1 (LITRM1)

tical distributions. Rating data obtained from collaborative filtering sites is similar to tagging data in that it has a familiar tri-partite structure. The first two elements of the data are the same; namely users and resources (hereafter referred to as items) and therefore we can use the same assumptions and distributions as we have in the previous sections. Namely that the distributions of topic allocations for resources and users will be distributed multinomial and will be drawn from Dirichlet priors. However it differs significantly in that the third of these elements is not a word drawn from a vocabulary but rather a numerical rating and that each user can only have a single link with an item (each user can only rate an item once). Since we are primarily interested in predicting ratings with the smallest possible error in aggregate it is sensible to consider models (and therefore distributions) that are continuous in nature. In doing so the predictions will not be constrained to be bound to the finite discrete values of the original ratings but will have the freedom to model the complex interactions of biases in the data at infinite granularity.

Symbol	**Description**
M	number of items
Y	number of topics/genres
Z	number of user interests
N	number of ratings
U	number of users
m_i	item for rating at position i
y_i	topic/genre allocation at position i
z_i	user interest allocation at position i
r_i	rating at position i
u_i	user who contributed rating at position i
ϕ_m	distribution over topics for item m
θ_u	distribution over interests for user u
b_m	bias due to item m
b_u	bias due to user u
b_{yz}	bias due to interest/topic pair yz
α	Dirichlet prior over Θ
β	Dirichlet prior over Φ
σ	standard deviation over all ratings
μ	mean rating

Table 3.2: List of notation for Ratings Models

3.4.1 Basic Generative Model (LITRM1)

Perhaps the simplest possible prediction algorithm one could imagine would be to use the mean rating over the training data, denoted μ (where $\hat{\mu} = \frac{1}{N}\sum_i r_i$), as a prediction for each item for every user, (i.e. $\hat{r}_{um} = \mu$). This overly simplistic model corresponds to a generative process in which each rating r_i is considered a normally distributed random variable with mean μ and standard deviation σ:

$$r_i \sim \mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.30}$$

This model makes a large number of assumptions and ignores a lot of the complexity in the data. It assumes that ratings are completely independent of both the item and user and that there are no interactions between the combinations of user and item that would affect the rating. We relax some of these assumptions and extend this model by following similar conclusions to both Hofmann [Hof04] and Marlin [Mar03], that the change in the rating is dependent on the user and that each user can be characterised by a distribution over a small number of latent interests.

In addition, (and in contrast to previous work) we then assume that the change in rating is also equally dependent on the items, which themselves can be characterised by a distribution over a small number of latent topics. For example in the case of movies this may be more intuitively thought of as their latent genres or for general items in a web store it could be the category/ies to which they could be categorised. This conjecture leads to a more useful generative model for personalised item filtering and ranking involving three random variables: a user interest z_i, an item topic (or movie genre) y_i and a rating r_i, where only the last variable, the rating itself, is observed. Following the models outlined earlier in this thesis we can then assume the user-interest and item-topic variables are distributed multinomial[5]. The same assumption could also be made regarding the ratings, as the original ratings assigned by users are indeed drawn from a discrete set. However as noted earlier the flexibility of the model can be increased, and also the granularity of its predictions, by instead modelling them as being drawn from Normal distributions. These assumptions can be summarised as follows:

$$z_i \sim Mult(\theta_{u_i}) \tag{3.31}$$

$$y_i \sim Mult(\phi_{m_i}) \tag{3.32}$$

$$r_i \sim \mathcal{N}(\mu+b_{y_i z_i}, \sigma^2) \tag{3.33}$$

Thus the model consists of a discrete probability distribution over interests for each user denoted θ_u, a discrete distribution over topics for each item denoted ϕ_m, a mean rating μ, a bias value b_{yz} for every pair of interests and

[5] The interest and topic variables are actually distributed according to a Categorical distribution, which is equivalent to a Multinomial distribution with a fixed count of 1. We use the term Multinomial in keeping with the literature.

topics, and a standard deviation parameter σ. A graphical model corresponding to this generative process is shown in Figure 3.8. The next question to address is what the parameters $b_{y_i z_i}$ being added to the mean rating μ to explain each observed rating r_i should be. We denoted these parameters as being biases, perturbing the rating away from the mean rating and allowing for much more statistical power than the extremely simple mean-rating model discussed previously.

The intuition for introducing the bias b_{yz} in this model is that we believe each interest and topic combination will likely have an effect on how the item is rated and that we can somehow capture this bias over the latent spaces. For example in the case of movies we might expect that a user who likes romance would give a horror movie a lower than average rating, meaning that the bias b_{yz} for this interest-topic pair would be negative. Similarly if the same user was to rate a romance movie then we would expect them to give a higher than average rating and the bias would therefore be positive. Since all of these biases would be calculated over the low-dimensionality latent spaces they will not be too sparse and should allow the model to generalise well to unseen user-item combinations, a key objective of any collaborative filtering model.

Given some estimates for these parameters, we can predict the rating for a user u and item m by calculating the expected value as follows:

$$\hat{r}_{um} = \mathbb{E}[r|u,m] = \sum_{y,z} \mathbb{E}[r|y,z] p(y|m) p(z|u) \tag{3.34}$$

$$= \mu + \sum_{y,z} b_{yz} \phi_{y|m} \theta_{z|u} \tag{3.35}$$

Here $\theta_{z|u}$ and $\phi_{y|m}$ denote probability of an interest given a user and a topic given an item respectively. This model is quite intuitive as it says that the rating given by a user to an item will be the product of a user's affinity for an interest, the item's probability of belonging to a topic and the average bias for that interest-topic combination, summed over all possible combinations of interests and topics.

We note that this new prediction model has far more flexibility than do "standard" Singular Value Decomposition (SVD) based ratings prediction algorithms, since the number of interests used to characterise users may be different from the number of topics used to represent items. This is not possible in the standard SVD approach, where the dimension of latent factors is necessarily the same for users and items. Moreover and more importantly, we have associated a possibly non-zero bias with every pair of interest topic dimensions. Thus not only is a positive bias associated with "corresponding"' interests and topics (e.g. the user-interest "horror" and the movie-genre "horror") but also a possibly negative bias with "non-corresponding"' interests and topics. For instance if a user's primary interest is "horror", they may still have a positive bias towards a "thriller" while having a negative bias against a "comedy". In SVD terms this is to some extent equivalent to replacing the diagonal singular-value matrix with a matrix containing non-zero off-diagonal values. These values then allow us to model both positive and negative correlations across different factors. Finally, by defining the predictions in terms of a generative model, we can interpret and explain the parameters of the model in a way that is not possible with SVD based prediction algorithms.

Given vectors of latent variable assignments $\mathbf{z} = (z_1, ..., z_N)$ and $\mathbf{y} = (y_1, ..., y_N)$, we can compute estimates of both the probability of an interest given a user $\theta_{z|u}$ and a topic given an item $\phi_{y|m}$. Following principles from LDA, and in keeping with Bayesian statistics, we place symmetric Dirichlet priors on both of these distributions, resulting in the following expectations for the parameter values under their respective posterior distributions:

$$\hat{\theta}_{z|u} = \frac{N_{zu} + \alpha\frac{1}{Z}}{N_u + \alpha} \tag{3.36}$$

$$\hat{\phi}_{y|m} = \frac{N_{ym} + \beta\frac{1}{Y}}{N_m + \beta} \tag{3.37}$$

Here N_{zu}, N_{ym}, N_u and N_u are counts denoting the number of times the interest z appears (in $\mathbf{z}$) together with user u, the number of times topic y appears (in $\mathbf{y}$) with item m, and the total ratings by user u and for item m respectively. Z is the number of interests and Y is the number of topics. The hyperparameters α and β act as pseudo-counts, allowing the model to fall back on the (uniform) prior probability in the event of sparse data, which is particularly useful in this setting where sparse data is common. They perform the same function as the hyperparameters in the tagging topic models and are in effect performing the same function. The only difference being that in this case they do not represent prior counts of word occurrences but instead prior counts of rating occurrences. Note that in calculating the values of these parameters we do not take into account the magnitude of each rating but we simply use a binary indicator of whether there is a rating or not.

In addition to estimating the distributions over interests and topics the model also needs an estimate of the bias for each interest and topic pair denoted b_{yz}. Given a complete set of assignments for these latent variables for each observed rating r_i an estimate of this bias can be calculated as follows:

$$\hat{b}_{yz} = \frac{\sum_{i:(y_i=y)\wedge(z_i=z)}(r_i - \mu)}{N_{yz} + \rho} \tag{3.38}$$

Here N_{yz} denotes the number of ratings for which y and z appear together and ρ is a smoothing parameter. This estimate is quite intuitive, it is calculating the mean perturbation from the mean rating for all ratings assigned to interest z and topic y. The smoothing parameter ρ is related to the variance of the zero mean Gaussian prior on b_{yz}, which keeps the model Bayesian and helps to deal with sparsity in the data[6]. Note that it would also be possible to estimate a variance parameter separately for each (y, z) pair, but this model instead makes the simplifying assumption that all biases have the same fixed variance.

In common with the other latent variable models investigated in this thesis, analytic inference of this model is intractable and therefore approximations of the posterior must be used. Gibbs sampling for this model involves sampling first z_i and then y_i for each rating r_i. To sample for z_i the distribution $p(z|r_i, y_i, u_i, \mu, \sigma, \mathbf{z}_{-i})$ is calculated, which is conditioned on the current assignment to all interest variables except z_i. Similarly for y_i a sample is drawn from the distribution $p(y|r_i, z_i, m_i, \mu, \sigma, \mathbf{y}_{-i})$. Note that the estimates for the parameters $\theta_{z|u}$, $\phi_{y|m}$ and b_{yz} depend on the interest and topic assignments $\mathbf{z}$ and $\mathbf{y}$, so when calculating estimates using Equations 3.36, 3.37 and 3.38, the i^{th} rating is removed from the sample. The conditional probability distributions

[6] The value ρ is equal to the ratio of the variances of the likelihood and the prior.

are then estimated as follows:

$$\begin{aligned}
p(z|r_i, y_i, u_i, ...) &\propto p(r_i|y_i, z)p(z|u_i) \\
&\propto \exp\left(\frac{(r_i-(\mu+b_{y_iz}))^2}{\sigma^2}\right)\frac{N_{zu}+\alpha\frac{1}{Z}}{N_u+\alpha} \qquad (3.39) \\
p(y|r_i, z_i, m_i, ...) &\propto p(r_i|y, z_i)p(y|m_i) \\
&\propto \exp\left(\frac{(r_i-(\mu+b_{yz_i}))^2}{\sigma^2}\right)\frac{N_{ym}+\beta\frac{1}{Y}}{N_m+\beta} \qquad (3.40)
\end{aligned}$$

Here $p(r|y, z)$ denotes the conditional probability density at rating r for the interest y and topic z. Since the algorithm only require estimates proportional to the true probabilities the normalising factor of the Normal distribution is not required. Therefore the first parts of Equations 3.39 and 3.40 are the unnormalised probabilities of a Normal distribution. This new model provides a method of predicting ratings by considering perturbations from the mean rating over a number of latent interests and topics. The next section describes an important extension of this base model that estimates individual biases for each user and for each item whilst still considering the bias over the latent topic and interest space.

3.4.2 Adding User and Movie Biases (LITRM2)

As noted in the previous chapter, the most successful models competing in the Netflix prize also estimate a bias for each user and a bias for each item as well as the bias due to the user and the item together. This is a sensible assumption as some users may naturally rate items higher than others and some may naturally choose from a lower baseline score. Similarly some items are intrinsically better than others and are therefore likely to be rated higher by all users, while the less quality items will be given a lower than average score by most users. While we would expect that these biases would be at least partially accounted for by the joint biases over the reduced genre and interest spaces it is likely that users and movies that give/have unusually high or low ratings (outliers) would affect the accuracy of the biases for other users. By calculating a separate bias for each user and item separately we

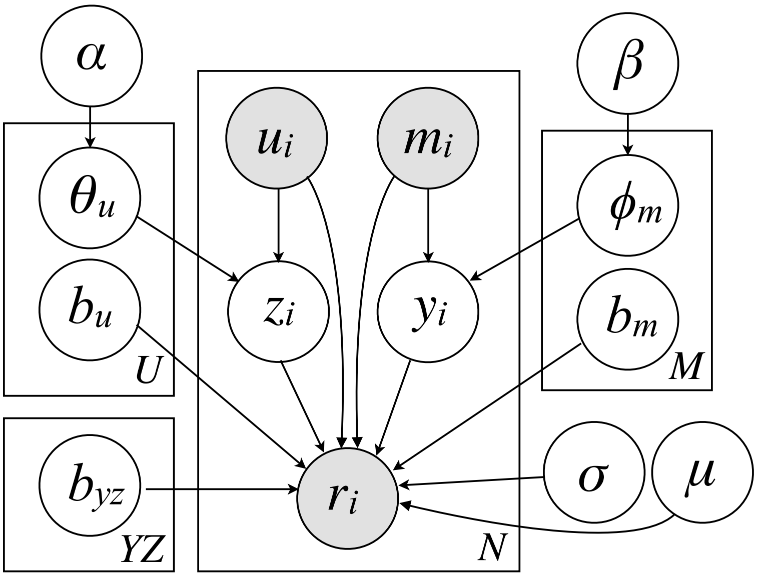


Figure 3.9: The extended Latent Interest and Topic Ratings Model (LITRM2)

effectively remove these eccentricities from the ratings, giving the joint biases the freedom to deal purely with the variations caused by observing the various interest/genres pairs. LITR2 is therefore an extension of the model described previously to also include these biases in order to improve prediction accuracy. The graphical representation for this model is shown in Figure 3.9.

The generative model is the same as the previous case, except that the mean of the Gaussian distribution that generates the rating r_i takes into account the user and item biases b_{u_i} and b_{m_i} as follows:

$$r_i \quad \sim \quad \mathcal{N}(\mu{+}b_{u_i}{+}b_{m_i}{+}b_{y_i z_i}, \sigma^2) \tag{3.41}$$

Given estimates for the parameters of this more complicated model, the predicted rating for a user u and an item m is now:

$$\hat{r}_{um} = \mathbb{E}[r|u, m] = \mu + b_u + b_m + \sum_{y,z} b_{yz}\phi_{y|m}\theta_{z|u} \tag{3.42}$$

Note that predictions under this new model and the previous model can both be viewed as perturbing the mean μ by a combination of biases. Both models add a bias for the likely interests and topics given the user and item pair, while the second model adds also explicit biases for the user and for the item.

Estimates for the parameters $\theta_{z|u}$ and $\phi_{y|m}$ are calculated as in the previous model, while the estimate for the bias b_{yz} must now include the effects of these extra biases as follows:

$$\hat{b}_{yz} = \frac{\sum_{i:(y_i=y)\wedge(z_i=z)}(r_i - (\mu + b_{u_i} + b_{m_i}))}{N_{yz} + \rho} \tag{3.43}$$

Furthermore estimates for the new user and item-dependent biases themselves must also be computed. The most obvious way to compute these biases is to take the mean difference of all ratings for a given user/item from the mean rating for all users/items. However since the model also includes a bias over the latent interests and topics spaces (denoted b_{um}) for each user-item pair, these estimates need to also include the effects of this bias in their estimators. The user and item biases are therefore estimated as follows:

$$\hat{b}_u = \frac{\sum_{i:(u_i=u)}(r_i - (\mu + b_{m_i} + b_{um_i}))}{N_u + \rho} \tag{3.44}$$

$$\hat{b}_m = \frac{\sum_{i:(m_i=m)}(r_i - (\mu + b_{u_i} + b_{u_i m}))}{N_m + \rho} \tag{3.45}$$

$$\text{where} \;\; b_{um} = \sum_{y,z} b_{yz}\phi_{y|m}\theta_{z|u} \tag{3.46}$$

Note that the Equations 3.44 and 3.45 are mutually dependent and thus an iterative fixed-point calculation is required to estimate the biases. Holding the joint b_{um} biases fixed this procedure converges very quickly and stabilises within less than ten iterations. Finally to complete the model estimation the distributions used for the Gibbs sampling routine must also be updated to include the new biases:

$$p(z|...) \propto \exp\left(\frac{(r_i - (\mu + b_{u_i} + b_{m_i} + b_{y_i z}))^2}{\sigma^2}\right)\frac{N_{zu} + \alpha\frac{1}{Z}}{N_u + \alpha} \tag{3.47}$$

$$p(y|...) \propto \exp\left(\frac{(r_i - (\mu + b_{u_i} + b_{m_i} + b_{yz_i}))^2}{\sigma^2}\right)\frac{N_{ym} + \beta\frac{1}{Y}}{N_m + \beta} \tag{3.48}$$

Since the user and item biases are not strongly dependent on the allocations of ratings to y and z we can simply estimate them after every k^{th} iteration of the Gibbs sampler and the algorithm with still converge. Not only does this speed up computation of the model but it also gives the sampler time to re-converge after changes to the user and item biases. In all the experiments performed later in the following chapters these biases are re-calculated after every 10 Gibbs iterations.

3.5 Unified Model and Latent User Communities

As mentioned at the start of this section, there are a number of clear similarities between the tagging models and the ratings models described in this thesis. They all model something as being a result of draws from distributions over some topic space dependent on a user and on a resource or item. It is therefore important to discuss the possibility of there being a single unified “core” model from which the 4 models presented could be derived.

This model would have the basic form of the first ratings model (LITRM1) where the resources in the tagging models and the items in the ratings models are modelled via the same distributions. Clearly in the tagging case this would require a joint distribution over words for each pair of topics and interests, the same as the joint biases for the ratings. Unfortunately this would further increase the sparsity of data for many of the estimates and may only be applicable in cases where there is a large amount of data. However it is possible that the extra flexibility afforded by this massive increase in parameters could further improve the accuracy of the models.

The modelling of user interests over a low-dimension topic space could feasibly allow for the identification of implicit user communities of practise. As discussed earlier, it is possible to discern communities of practise within a social system by isolating groups of users who have shown an interest in similar (or the same) resources or items. This assumption has been frequently used as a crucial part of memory-based collaborative filtering algorithms. However by having a representation of user interests over a latent topic space, we may be able uncover more subtle relationships in the data than is possible using simply co-occurence.

A straightforward method of identifying such communities would be to single-out groups of users who have high topical probability, under the model, for the same latent topic. Alternatively, if we were interested in identifying users who are in some sense similar to a given target user we could define a basic similarity metric by summing topic probabilities over all latent topics and then ranking the results in descending order. For example given a target user u^1 we could calculate a similarity measure for another user u^2 as follows:

$$sim(u^2, u^1) = \sum_{z=1}^{Z} p(z|u^1)p(z|u^2) \tag{3.49}$$

It may also be sensible to weight each product by the probability $p(z)$ of the topic so as to give a higher influence to frequently observed topics. These measures could then be used for a number of useful tasks, for example to suggest friends to users based on their shared interests or to suggest which latent topics would be of interest to a given user.

Both of these possibilities illustrate interesting possibilities for future work, other possibilities are briefly discussed in Chapter 7.

3.6 Conclusions

This chapter has described the mathematical and statistical theory underpinning the models used in this thesis. It discussed the ideas behind Bayesian probability and latent topic models and has motivated their use for applications using the sparse and noisy data obtained from social sites. The following chapters of this thesis will show by experiment on real (non synthetic) data that not only can these models be used to perform useful tasks on such data but that they are also able to out-perform other state-of-the-art methods in the field.

Chapter 4

Experiment 1: Tag Suggestion

> "Experiment is the sole source of truth. It alone can teach us something new; it alone can give us certainty"
>
> *Henri Poincaré*

In this first experiment the aim is to suggest relevant tags that a user may wish to use to annotate a resource. Given that the TTM models are personalised these suggested tags can by chosen based on both the tags already used to annotate the resource and the user's own profile. The basic problem of tag suggestion can be described as the following: conditioned on a *pseudo-query* q consisting of tags already chosen by the user, $q := \{w_1, \ldots, w_i\}$ the algorithm should rank the remaining tags in the vocabulary W in descending order of probability. In doing so it must in some way calculate $p(w|q)$; the probability of a word w given the pseudo-query q. Attempts can be made to improve the accuracy of predictions by also incorporating the user's interests and vocabulary choice into the estimate by calculating $p(w|q, u)$.

4.1 Suggesting Tags

Given some initial tags provided by the user for a given resource and the output from the topic model, the algorithm should predict which tags the user will enter next and offer them as suggestions. To do this it must first estimate a distribution over the latent topic space for the pseudo-query q comprising the tags supplied by the user. This can be calculated as a point estimate:

$$p(z|q) = \frac{N_z^{(q)} + p(z)\alpha_d}{N^{(q)} + \alpha_d} \tag{4.1}$$

where $N^{(q)}$ is the total number of tags in q and $p(z)$ is the relative frequency of topic z in the model. To calculate a value for the topic distribution $p(z|q)$, an estimate is required for the value of $N_z^{(q)}$, the expected count for the topic z in q. The expected value for $N_z^{(q)}$ can be calculated by summing over all tags in the pseudo-query as follows:

$$E[N_z^{(q)}] = \sum_{w \in q} p(z|w) N_w^{(q)} \tag{4.2}$$

where $p(z|w)$ can be calculated using the ϕ_z distribution from the model via Bayes' rule : $p(z|w) = \phi_{w|z} p(z)/p(w)$, and $N_w^{(q)}$ is the number of times tag w appears in the query q. Note that each $N_w^{(q)}$ will generally always be one since it is highly unlikely that a single user will use the same tag more than once to describe the same resource. Given this estimated distribution over the latent topics for q an estimate can be made for the probability of observing a new tag w:

$$p(w|q) = \sum_{z=1}^{Z} \phi_{w|z} p(z|q) \tag{4.3}$$

This returns the estimated probability of a term in the corpus given the pseudo-query, if this is calculated for $\forall w \in \mathcal{W}$ and then ordered by probability in descending order the top n terms can be chosen in this ranked list as tag suggestions. The tagging topic models benefit from also having estimates for user interests over the topics and this can be used to include the user's personal preferences in the suggestions. Based on the matrix Ψ from the tagging models, the personalised distribution over terms can be calculated thus:

$$p(w|q,u) = \sum_{z=1}^{Z} \phi_{w|z} \frac{p(z|q)\psi_{z|u}}{p(z)} \tag{4.4}$$

where u is the user who generated the tags for the pseudo-query (i.e. the user currently tagging the resource). This final distribution over terms indicates each term's probability given the previously observed terms and the topical interests of the user. These probabilities can then be used as a multiplier on traditional tag suggestion methods (such as the one outlined in the description of baseline method 3 below) and provides a smoothed, personalised weighting for each term.

4.2 Experimental Set-up

In order to evaluate the applicability of the tagging models to this problem I conducted experiments comparing them and LDA with 3 "baseline" methods through empirical evaluation based on held-out data from a real-life data set obtained from a large online social tagging system. This section outlines the experimental set-up in more detail, explains the various methods for tag suggestion and briefly describes the data set and the settings of parameters for the tagging topic models.

4.2.1 Evaluation Method

In order to evaluate the accuracy of the tags suggested by each algorithm some form of relevance judgement is necessary, for example a list of all accurate and useful tags for each resource. One method for doing this that has been utilised previously is a user study where users are asked if they think that tags suggested for each resource are relevant or not. This work does not follow this method as it is interested specifically in personalised results, therefore only the user(s) who originally tagged the resource can really say whether a tag is relevant or not. In this case a user study would likely provide an overestimate of the quality of the results and therefore it was chosen to evaluate the systems based only on the tags provided by the user on the live system. Given a set of l tags for a given resource m tags are chosen at random as input for the suggestion algorithms and the remaining l-m are used as the set of relevant suggestions. These resource are chosen from a set of held-out resources (i.e. resources that have not been used to train the model) and will give an estimate of the quality of the suggested tags. Since these assessments were derived from a sample of data from a working social network they are likely to more accurately reflect the performance of a live system. This segmentation of the data is described in Figure 4.1 where for each user there is a sequence of annotated resources consisting of tags. Some of these annotations are used for training and the rest are used for testing with each test-set annotation being segmented into pseudo-query terms, shown in blue, and relevant terms, shown in grey.

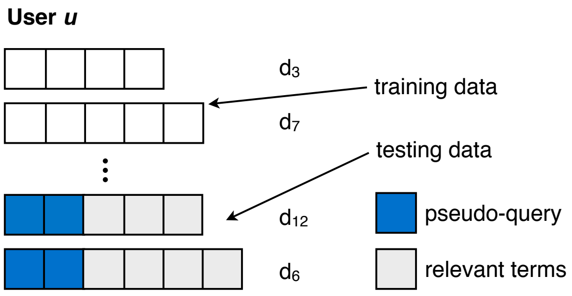


Figure 4.1: Data segmentation method for tag suggestion experiment

Since this experiment is intended to test the ability of various algorithms to return a good ranked list of suggested items the following evaluation metrics are used:

P@k - "precision at rank k" the ratio of suggested tags that are relevant, averaged over test resources. P@k is reported for k=1, k=5 and k=20.

S@k - "success at rank k" the ratio of times where there was at least 1 relevant item in the first k returned. S@k is reported for k=1, k=5 and k=20. S@1 and P@1 are the same and are therefore not reported separately.
$S@k = \frac{1}{|q|} \sum_i^{|q|} I(rank(d_i, q_i) \leq k)$
where I is the indicator function.

MRR = "mean reciprocal rank" the multiplicative inverse of the rank of the first relevant suggested item, averaged over test resources.
$MRR = \frac{1}{|q|} \sum_i^{|q|} \frac{1}{rank(d_i,q_i)}$

Note the choice to evaluate the precision and success metrics for k values up to 20 as this is the number of tags usually suggested on social tagging web sites. k values of 1 and 5 are the most commonly reported in other literature and people tend to pay more attention to the first few results in a ranked list. When training the systems 20% of resources chosen at random and held-out and then are "binned" into two sets; one (hereafter referred to as set1) containing documents with between 4 and 8 annotations and the other (set2) with 9 or more annotations. This allows for analysis of the performance of the models for both well annotated and poorly annotated resources.

4.2.2 Baselines

So that the results from the tagging models are compared to the algorithms already used in social tagging systems the above tests are run on 3 "baseline" methods, LDA as well as on both of the Tripartite Topic Models (TTM1 and TTM2). The first 2 of these methods simulate the tags that would be suggested on sites such as Flickr and Delicious and the final baseline method represents a slightly more sophisticated algorithm that has been proposed in previous literature [SvZ08, Sch06].

TopSys the simplest set of suggestions; the top k tags in the system by frequency of use.

TopUser the most frequently used tags by the user who tagged the resource, if more than 1 user has tagged the resource the union of all users' tags is used.

CoTag tag co-occurence using asymmetric normalisation, as used in previous research to find like terms in a folksonomy [Sch06]. This method is described in more detail below.

CoTag

The third baseline measure used is based on the concept of normalised tag co-occurence. The tag co-occurence between two tags i and j is simply the number of times those tags are both used to annotate a single resource, i.e. $|i \cap j|$ - the union of the two tags. These raw co-occurence numbers do not provide particularly useful information as they fail to take the frequency of individual tags into account. Therefore these raw values are commonly normalised by the frequency of the tags, this can be done either symmetrically or asymmetrically. The symmetric method is also known as the Jaccard coefficient and normalises the co-occurence frequency by the union of the two tag frequencies:

$$J(i, j) = \frac{|i \cap j|}{|i \cup j|} \tag{4.5}$$

The asymmetric method only normalises the frequencies by diving over the number of times the second tag is used:

$$Sim(i, j) = \frac{|i \cap j|}{|i|} \tag{4.6}$$

This can be interpreted as the probability of a resource being annotated with tag j given that it has also been tagged by i. If these values are summed for all terms in the psuedo-query then a ranked list of tags related to the pseudo-query can be obtained, this can be seen as similar to the result from the topic models described previously. This asymmetric method has been frequently used in previous research in order to find like terms in a folksonomy [Sch06] and so is used as a basis for the third baseline.

4.2.3 Data Set

Unlike more traditional forms of Information Retrieval, no standard data sets are yet available for the evaluation of social tagging systems. This does introduce the issue of acquiring sufficient data for testing, however it does mean that data collected can be drawn from a system that is currently in use, thus providing a more realistic setting for the analysis. For these experiments the tests were conducted on data provided by Bibsonomy[1] - a social bookmark and publication sharing system and a good example of a large, broad folksonomy

[1] Knowledge and Data Engineering Group, University of Kassel: Benchmark Folksonomy Data from BibSonomy, version of June 30th, 2007. http://bibsonomy.org/

and as such is ideal for the research aims of this work. The Bibsonomy data set shares similar characteristics with other large folksonomic data sets noted in previous research, most notably the tag use frequency follows a power law, as does the number of annotations per resource.

To filter out noise and to provide useful data for the evaluation methods any resources that have less than 4 annotations are discarded and similarly any tags that are used to annotate less than 5 resources are removed. This results in a data set of 36167 resources from 992 users with a total vocabulary of 5116 terms, 28143 (77.8%) of the resources fit into set1, the remaining 8024 (22.2%) fit into set2. Stratified random sampling was used to select test data resulting in a total of 7235 (20%) held-out resources with 5630 (79.4%) from set1 and 1605 (20.6%) from set2. In order to ensure that the results returned are not simply due to the held-out resources chosen all tests are performed over 10 different random folds. The unfiltered data set displays similar characteristics to those of other folksonomies analysed in related literature; the mean number of tags per resource is 3.27 (median 2), 68.6% of all resources have less than 3 tags. The filtered data therefore represents only 31.4% of the total Bibsonomy data set, highlighting the wide applicability of a good tag suggestion system.

Figure 4.2 shows histograms of resource description count (i.e. the count of all the tags, conflated over all users) and user count for the resources in the data set. To better show the distribution, those resources only tagged by a single user are not included in the user count plot. We can see that for both metrics the distributions exhibit quite consistent exponential decay, indicating that they may follow a Zifpian or Pareto distribution as would be expected for such data. This clearly demonstrates the need for both tag suggestion and use of robust and intelligent algorithms in such systems. Tag suggestion is likely to shift the length distribution so that in expectation, resources are more thoroughly annotated. Also, since for the vast majority of cases the data only contains 1 or 2 users' descriptions of resource, the "wisdom of crowds" effect will not often be present and therefore understanding of the semantics of the few tags available is key. It is in these cases that the topic models are likely to have the greatest advantage, particularly the tripartite models as they are able to also leverage understanding of the tagger's interests and word choice patterns.

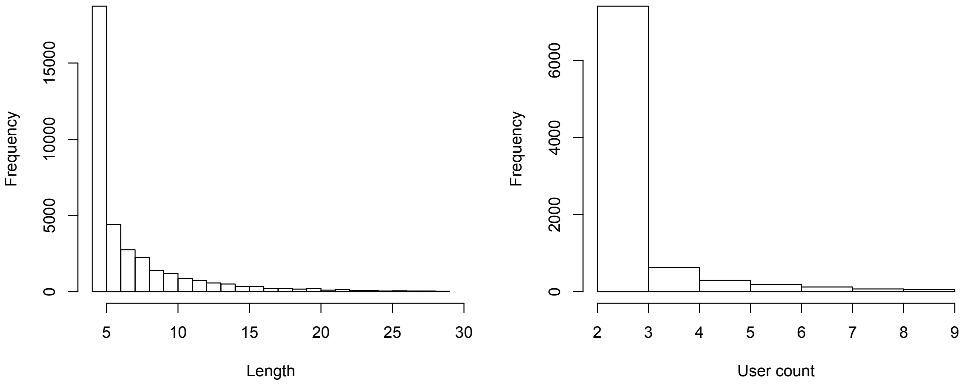


Figure 4.2: Histograms of resource description length and user count

4.2.4 Parameter Settings and Determining Convergence

One very influential parameter that must be set in any latent topic model is the value of Z; the number of latent topics in the model. In this analysis the results in terms of precision and recall for the tag suggestion algorithms are compared over a series of monotonically increasing values of Z. The value of Z where the delta improvement in metrics over the previous value is small is where the optimal value of Z lies. When run on the Bibsonomy data set a correlation was found between both metrics for the values over successive values of Z, with both indicating that around 200 latent topics provides the most optimal fit for the data using the tagging models. In general, topic models are not particularly sensitive to hyperparameter values, however it is still pragmatic to make sensible choices for them and investigate their effect of performance. Since all the hyperparameters in the models are symmetric Dirichlets in reality only a single value needs to be chosen for each which will then be applied over the Z dimensions of the model. Choosing large values will add significant smoothing and may cause the distribution over latent topics to be quite even, with all dimensions being assigned approximately the same probability mass of $\frac{1}{Z}$. On the other hand choosing a very small value will allow the data to overwhelm the prior much more easily, resulting in a more peaked distribution. It is generally advisable to choose values that mediate between these 2 extremes in order to allow the data to speak for itself whilst maintaining some smoothing and parameter tying in the event of sparse data. The concentration parameters α and β were set to 25.0 and $0.1W$ respectively, meaning that the α setting is slightly lower than is common in the literature [GS04]. It was found that a slightly smaller value provided better results, perhaps because the average length of a "document" (resource) in these systems is much less than in a more standard IR corpus. For both personalised models γ was set to 25 and in the TTM2 model α was set to $0.1D$.

The Rao-Blackwellised Gibbs sampling method proposed by Styvers and Griffths [Hei08, GS04] is used to sample the models. It is important when using methods such as Gibbs sampling to estimate a posterior distribution that the Markov chain is given enough time to "burn-in", i.e. when it begins to approach a stationary distribution [SR93]. To determine when the chains are beginning to approach a stationary distribution the perplexity[1] or log-likelihood of the model can be calculated given the currently sampled estimate every n iterations. If the chain is converging correctly these values should initially decrease quite rapidly, however as the chain approaches convergence the change (delta) in perplexity should become smaller until the deltas become negligible. For all of the topic model estimations the first 300 iterations of the chain were discarded and then averages were taken over samples of the chain thereafter until reaching 400 iterations. FIgure 4.3 shows the normalised perplexity scores for LDA and TTM1 over the iterations of the sampler. The tagging model takes slightly longer to converge initially, which is not surprising given its increased complexity, but eventually is able to attain a smaller perplexity than LDA, suggesting a better fit to the data. Both models appear to have converged well within the first 300 iterations indicating that the samples taken from the last 100 samples are being drawn from a close approximation to the posterior, perplexity within the kept samples is still changing by small amounts suggesting that the chain is mixing well and avoiding getting stuck in local maxima. Due to it having a very different representation of documents, perplexity results from TTM2 cannot be directly compared with those from

[1] The perplexity is monotonically decreasing in the likelihood of the data and is equivalent to the inverse of the mean per-word likelihood.

TTM1 and LDA. Figure 4.4 shows the perplexity of TTM2 over iterations of the sampler where we can see that the chain appears to have converged well within the first 200 iterations, is mixing well and in fact seems to converge much more quickly than for TTM1.

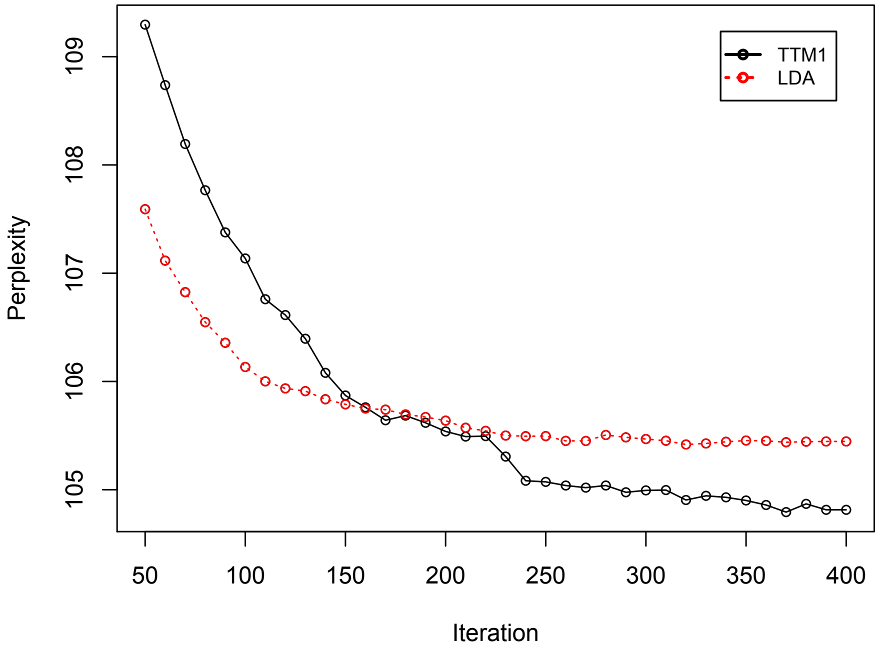


Figure 4.3: Perplexity over iterations of the sampler for LDA and TTM1

4.3 Results

This section presents the results from the series of experiments described in the previous section. It first looks at overall performance of the 5 tag suggestion methods for a “typical” scenario of a user providing 2 tags for the method to base their suggestions on, the difference in performance over the two resource “sets” is analysed and comments are made on how this is likely to relate to real-world performance. Finally the section looks at how varying the number of user tags provided affects the quality of tags suggested by the models.

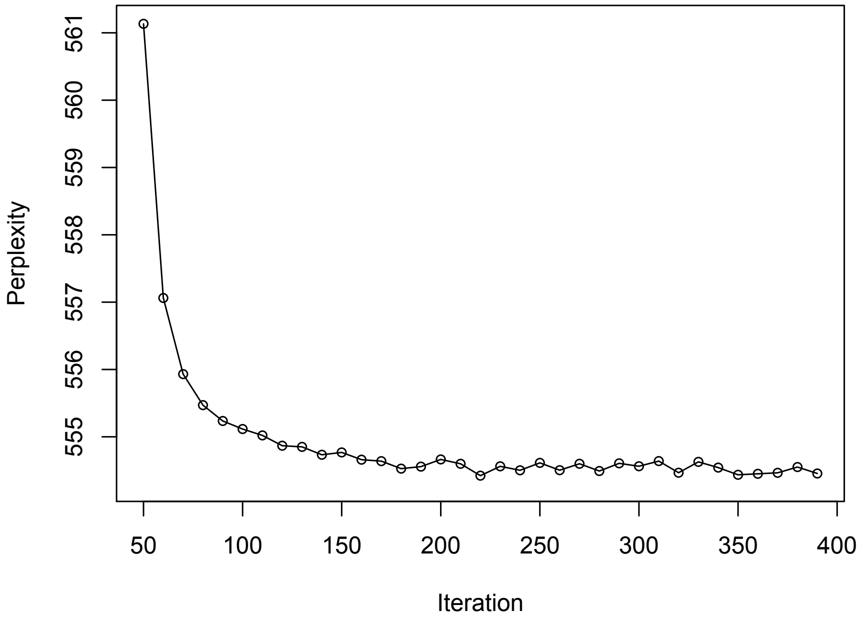


Figure 4.4: Perplexity over iterations of the sampler for TTM2

4.3.1 Tag Suggestion Performance

The results of the tag suggestion tests using resources from set1 (sparsely annotated resources) are presented in Table 4.1. Note that emboldened results indicate the best result achieved for that metric and * indicates a statistically significant improvement over CoTag at 95% confidence. The results for TopSys over all metrics are extremely poor (as expected), the results for TopUser are slightly better but still well below those returned by the other more sophisticated methods. Statistically significant improvements of both tagging topic models over both CoTag and basic LDA are observed over all metrics. These results show that the tripartite models are able to fit the available training data better than the other methods and therefore provide more useful and accurate suggestions. The larger improvements in precision and MRR indicate that the TTM methods are suggesting fewer incorrect tags and are returning more relevant tags at a higher rank than the other methods. The results also indicate that TTM2 is indeed a better model for tagging data than TTM1, especially where information available is sparse. Table 4.2 shows examples of the suggestions provided by TTM2 for 3 different resources, bold guessed terms are correct.

	TopSys	**TopUser**	**CoTag**	**LDA**	**TTM1**	**TTM2**
S@1	0.0490	0.2269	0.3449	0.3197	0.3736*	**0.4048***
S@5	0.1540	0.4495	0.5648	0.5494	0.6270*	**0.6515***
P@5	0.0353	0.1329	0.1786	0.1705	0.2029*	**0.2334***
S@20	0.3552	0.6853	0.7637	0.7583	0.8238*	**0.8332***
MRR	0.1023	0.2718	0.3608	0.3574	0.4056*	**0.4058***

Table 4.1: Results for sparsely annotated resources

Given	**Relevance judgements**	**Suggestions**
php, tool	php, software, tools, opensource, internet, design, free, resources development freeware php cms	**software**, internet, mysql, **design**, webservice
ocean, sea	flickr, photos, sky, ocean, sandy, beach, ocean, sky	**sky**, water, scenery, sand, **beach**
reference, podcasts	reference, education, free, learning, online, courses, podcasts, reference, education	**learning**, mp3, **education**, **online**, lernen

Table 4.2: Examples of tag suggestions made by TTM2

The results from resources from set2 (densely annotated resources), presented in Table 4.3, show that while the tagging models are still able to outperform other methods over all metrics, the improvements are smaller. In this case the difference in performance between CoTag and TTM1 is statistically significant for all metrics except for S@20. Again the greatest improvements are in precision and MRR, however all improvements over LDA and CoTag are smaller with the success metric being fairly similar for all 3 methods. This is likely because the small number of resources where the systems are unsuccessful are annotated with terms that have either not been used together before or do not exist at all in the training set. In this case the scope for performance improvement over the CoTag method is very small. For these resources TTM2 does not show improvement over TTM1 as it did for the sparser resources and for some metrics TTM1 actually returns a better score however none of these differences are significant. This is indicating that for resources where a large amount of data is available the benefits gained from personalisation are less significant.

	TopSys	**TopUser**	**CoTag**	**LDA**	**TTM1**	**TTM2**
S@1	0.1576	0.3499	0.6312	0.5879	0.6437*	**0.6441***
S@5	0.3829	0.5882	0.7811	0.7693	**0.8132***	0.8117*
P@5	0.1258	0.2436	0.4007	0.3796	0.4236*	**0.4241***
S@20	0.6593	0.8246	0.9376	0.9329	**0.9516**	0.9513
P@20	0.0749	0.1391	0.2022	0.1972	**0.2181***	0.2179*
MRR	0.2244	0.2788	0.3857	0.3890	**0.4125***	0.4118*

Table 4.3: Results for densely annotated resources

4.3.2 Varying the Number of Input Tags

Selected results from CoTag and TTM1 are presented in Table 4.4 for varying numbers of input tags where the number of input tags is indicated by the number in the square brackets. These results indicate that the performance of CoTag at 2 and 3 input tags is significantly better than with 1, however there is little difference between the performance with 2 or 3 tags. TTM performs well when only given a single input tag to infer suggestions from and its performance in terms of precision and MRR increases as the number of input tags increases. This is a very useful property as in many cases users will only supply a single tag so it is important that the method is able to make good suggestions based on such a small amount of information. Success@k metrics are not significantly different over varying numbers of input tags.

4.4 Conclusions

This experiment has demonstrated that the models for social tagging data derived in the previous chapter can suggest more relevant tags than current systems by comparing these to held-out tags from annotated resources. In terms

	CoTag[1]	**CoTag[2]**	**CoTag[3]**	**TTM[1]**	**TTM[2]**	**TTM[3]**
S@1	0.6058	0.6398	0.6186	0.6464	0.6594	0.6492
S@20	0.8986	0.9322	0.9388	0.9366	0.9522	0.9520
P@20	0.1936	0.2022	0.1948	0.2214	0.2245	0.2302
MRR	0.3494	0.4032	0.4061	0.3966	0.4172	0.4243

Table 4.4: Results from densely annotated resources from fold 10 for varying number of input tags

of precision, the use of the new models improves upon the suggestions provided by the CoTag method on sparsely annotated resources by between 7.87 and 30.6%, improves upon basic LDA by 11.4 to 36.9% and vastly outperforms the more common TopSys and TopUser methods. The results are particularly promising for sparsely annotated resources which are extremely common in tagging systems, indicating that suggestions from the tripartite models would work well in a live system. Analysis of the data obtained from BibSonomy indicated that this might be the case and highlighted the importance of this particular behaviour of the tripartite models as resource are frequently sparsely annotated. The significant improvements over LDA highlight that the user's tagging profile can be successfully incorporated into the model and used to improve estimation performance. For sparse resources where the user profile information is able to be brought to bear most effectively, TTM2 is shown to be a better model of the data than TTM1 and over some metric is able to deliver significant performance improvements. However for more densely annotated resources the differences between the 2 models are negligible and are well within the bounds of normal statistical variation.

Chapter 5

Experiment 2: Personalised Search

> "Facts are meaningless. You could use facts to prove anything that's even remotely true!"
>
> *Homer Simpson*

Following the successful results from the previous section this experiment takes the idea of personalisation further by attempting to personalise search results, again using a large sample of real data to validate the performance of the models. The experiment is similar to the previous one in that items are to be ranked based on both some new input data and the current user's topical interests, however in this case the models must provide personalised ranking for resources rather than tags. This problem is clearly drawn from the field of information retrieval and in particular, language modelling.

The field of Information Retrieval (IR) generally involves the ranking of a set of documents from a corpus given a textual search query supplied by a user. An ideal system would rank the documents in descending order of relevance to the query provided therefore maximising the chance of the user finding the one(s) that will best fulfil his information need. Viewing the problem from a probabilistic viewpoint we could also describe this ideal system as ranking the documents in descending order of the probability of utility to the query. In Language Models a separate model is constructed of the language contained within each document so that for each unique term in the corpus there is the probability of that term given the document $p(w|d)$. The assumption can then be made that each term in the query is drawn independently from this distribution and therefore the probability of the document given the query can be calculated in the following manner:

$$p(d|q) \propto p(d)p(q|d) \quad = \quad p(d)\prod_{w \in q} p(w|d) \tag{5.1}$$

This requires an estimate for $p(w|d)$, the probability of term w occurring in the language model for document d. This can be calculated via the principle of Maximum Likelihood thus: $p(\hat{w}|d) = N_{w,d}/N_d$ where $N_{w,d}$ is the count of term w in document d and N_d is the length of d. These raw estimates do not work well in practise due to the finite length of documents, particularly in the case of socially generated documents descriptions which comprise a small number of tags. It is therefore necessary to apply some form of smoothing to these estimates where for each term some "extra" probability mass is added to the term counts actually observed in the documents. These additional probability masses are usually in proportion to that term's frequency of occurrence in the overall corpus. Regardless of how this is calculated the ranking formula consists of the product of 2 distinct parts; a prior on the probability of the resource, $p(d)$, and the probability of the query given the resource, $p(q|d)$.

5.1 Ranking Resources

Following these assumptions, formulas for ranking resources using the parameters estimated in the topic models described in section 3.3 can be derived. As described above, given a query q the formula should return to the user a ranked set of resources ($d \in D$) according to their likelihood given the query under the model. In topical models, rather than using raw term probabilities, the $p(w|d)$ can be modelled by summing over the topics and can follow the assumptions in the generative process to devise an appropriate ranking algorithm. In the case of LDA this can be estimated as follows:

$$\begin{aligned} p(d|q) \propto p(d)p(q|d) &= p(d)\prod_{w\in q} p(w|d) && (5.2)\\ &= p(d)\prod_{w\in q}\sum_z p(w|z)p(z|d) && (5.3)\\ \textit{where} \qquad & p(d) = N_d/N && (5.4) \end{aligned}$$

Since the data also records which user has issued the query the formula for the tagging models can also include that user's preferences into the ranking. Therefore they rank documents according to their likelihood given both the query *and the user*, the ranking formula for **TTM1** is:

$$\begin{aligned} p(d|q,u) &\propto p(d|u)p(q|d,u) = p(d|u)\prod_{w\in q} p(w|d,u) && (5.5)\\ \textit{where} \quad p(d|u) &= p(d)\sum_z \frac{p(z|d)p(z|u)^{\pi_u}}{p(z)} && (5.6)\\ \textit{and} \quad p(w|d,u) &= \frac{\sum_z p(w|z)p(z|d)p(z|u)^{\pi_u}p(z)^{-1}}{\sum_z p(z|d)p(z|u)^{\pi_u}p(z)^{-1}} && (5.7) \end{aligned}$$

In the case of **TTM2** the $p(d|u)$ and $p(q|d,u)$ are as follows:

$$\begin{aligned} p(d|u) &= \sum_z p(d|z)p(z|u)^{\pi_u} && (5.8)\\ \textit{and} \quad p(w|d,u) &= \frac{\sum_z p(w|z)p(d|z)p(z|u)^{\pi_u}}{\sum_z p(d|z)p(z|u)^{\pi_u}} && (5.9) \end{aligned}$$

Again notice that the formulas above are the product of 2 parts: a *user-specific* document prior, $p(d|u)$, and the probability of the query given the resource and the user, $p(q|d,u)$. Notice also that a weighting parameter π_u has been introduced in the range zero to one on $p(z|u)$ so that its influence of the user's topical interests on the rankings can be varied. The intuition behind this being that resources likely tell us more about their own topic distribution than the users who annotated them.

5.2 Experimental Set-up

In order to evaluate the relative performance of the tagging topic models on real-world data a series of experiments were performed comparing them with LDA and 3 other baselines, 2 of which are extremely competitive and represent the best of traditional and more modern IR techniques. Data to perform this experiment was obtained by conducting a crawl of the popular social bookmarking site delicious, a process described in more detail later.

5.2.1 Evaluation Method

In order to generate queries to input into the ranking algorithms it is possible to use sets of tags from a social bookmarking system. To do this each bookmark in the test set (i.e. set comprising all tags for a resource contributed by a single user) can be treated as a pseudo query. Clearly, to evaluate success some form of relevance judgement for each pseudo-query is required and since it is known which resource was chosen for each bookmark a ranked resource can be said to be relevant if it is the same resource the user actually bookmarked. In keeping with the beliefs outlined in the previous experiment on tag suggestion, this method is chosen as we are interested in personalised results, therefore only the user(s) who originally tagged the resource can really say whether it is truly relevant to them or not. Again, this evaluation technique will more accurately reflect the performance of a live system and is likely to in fact give a slight under-estimate of the true performance.

In order to evaluate ranking performance the success at rank k (S@k)[1] and the mean reciprocal rank (MRR) can be calculated. These 2 measures were described in detail in chapter 4. Since the primary interest is of how well these models rank URLs, the S@k and MMR are reported up to rank 10 as they are the most commonly reported in other literature since people tend to only pay attention to the first page of results in a ranked list.

[1] Note that since one bookmarked URL per set of tags is available, precision at rank k (P@k) is equal to S@k/k and thus it is not reported separately.

5.2.2 Data Set

Again, there is no standard test set available for social tagging data and therefore it is necessary to obtain real data from a social tagging web site. In order to demonstrate the abilities of the models it is important to choose a broad folksonomy to draw the data from so that some information will be available in the training set for each resource since more than one user is able to annotate each resource. This also allows the models to demonstrate any performance gains achieved due to personalisation and will show that they are able to deal with different vocabulary being used to describe the same resource. For this experiment a crawler was written in Java to obtain a large sample of tagging data from the popular (and seminal) social bookmarking system delicious.

When crawling delicious for data it was important to ensure a random sample of recent data, to do this the crawler began by downloading the 100 most recent URLs submitted to delicious and recorded the usernames of the users who bookmarked them. It continued this process until a sample of 60,663 unique usernames had been collected. Then for each of these usernames the respective user's 100 most recent bookmarks were downloaded (as this is the largest number of recent bookmarks the delicious API will allow access to). Note that as 100 is the maximum number of bookmarks available via this crawling method per user not all users had this many bookmarks available resulting in 31% of the users having less than 100 bookmarks.

Each "document" (URL) is uniquely identified by computing a 32 bit MD5 hash of the complete URL, each URL and user in the data set was assigned a unique and anonymous ID number. To clean the resulting data set, only the URLs which had been bookmarked by more than 2 unique users were selected to ensure that all resources will always exist at least once in the training data. In order to give the systems reasonably complete user profiles to work from only the users who had bookmarked more than 60 unique URLs from the remaining data after the first pass were selected. Each remaining bookmark is a triple consisting of a URL identifier, a user identifier and a set of tags. The set of tags were parsed for each bookmark and finally all tags that appeared less than 2 times in the data set were removed.

Metric	Original	Reduced
users	60,663	9,587
URLs	476,248	111,232
vocab count	113,428	14,023
bookmarks	3,235,299	569,117
word occurrences	12,294,136	2,473,738
avg bookmarks/user	53.3	59.4
avg bookmarks/URL	6.79	5.1
avg annotations/URL	25.8	22.2
avg annotations/bookmark	3.8	4.3

Table 5.1: Counts and statistics for the original dataset created from the delicious crawl (Original) and after reduction (Reduced)

The dataset was separated into training and testing subsets by retaining the last 10% of bookmarks by each user for testing. Doing so ensures that the test data is distributed over users in the same way as the training data. Therefor the model is trained on the first 90% of all bookmarks tagged by each user, i.e. all of the tags that each user assigned to those resources contained within the first 90% of their complete set of bookmarks. This means that for each user 10% of his/her bookmarks and associated tags are unknown to the system and these can therefore be used to test the system. Given the stipulation that each individual resource must be bookmarked by more than one user this means that the system will still have some tags to describe each resource, however it will not have been trained on the tags assigned by the user doing the search.

The original data set and the resulting reduced set is described in more detail in Table 5.1, notice that the averaged statistics for both datasets are quite similar.

5.2.3 Baselines

In order to usefully evaluate the performance of the topic models they are compared with 3 different baselines; SMatch - which emulates the kind of simple matching formulas currently used when searching social tagging sites, Okapi BM25 - a popular and quite robust probabilistic retrieval framework and BayesLM - a competitive baseline Language Model with Bayesian smoothing. For each of the baseline methods any free parameters were optimised to ensure a fair and unbiased comparison with the topic models. Here I briefly describe the formulas for these models:

SMatch $score(d,q) = \sum_{w \in q} N_{w,d}$

BM25 $score(d,q) = \sum_{w \in q} IDF(w) . \frac{N_{w,d}(k_1+1)}{N_{w,d}+k1(1-b+b\frac{|d|}{avgdl})}$
where $IDF(w) = \frac{N-N_w+0.5}{N_w+0.5}$, $|d|$ is the length of resource d and $avgdl$ is the average length of a resource over the whole training corpus. k_1 and b are free parameters which were optimised to 2.0 and 0.1 respectively.

BayesLM $p(d|q) = p(d) \prod_{w \in q} \frac{N_{w,d}+\mu(N_d/N)}{N_d+\mu}$
where μ is the Bayesian smoothing parameter which was optimised to 0.75.

Note that BayesLM is the same as the non-personalised model used by Wang et al. [WCY$^+$10] except that it was adapted to deal with queries of lengths greater than one and can be described as a language model with a Bayesian prior on the term probabilities. The full personalised model described in the paper was tested as a baseline, but was found to perform extremely poorly. This is perhaps because this experiment uses a much larger data set with a vocabulary 14 times larger than theirs. In this case their choice to use raw tags as user profiles (rather than reduced dimensionality features as in this thesis) may have resulted in significant overfitting and poor generalisation. Therefore the results from their original personalised model are not reported.

5.2.4 Parameter Settings and Sampling

A large range of parameter settings for both the number of topics in each model were tested, (discussed further below), and similarly for the hyperparameter settings for each of the prior distributions. The same values for the hyperparameters as in the earlier tag suggestion experiment 4 were used. Again, none of the topic models were particularly sensitive to parameter value choice, provided one does not choose excessively low values, where almost no smoothing is being applied or in the other extreme very high values; smoothing out the information from the data completely.

For sampling the Rao-Blackwellised Gibbs sampler [GS04] is used. For all models the chain is sampled for 300 iterations in total, as this appeared to consistently give good convergence in terms of model likelihood, and the first 200 samples are discarded as chain "burn-in". The remaining 100 samples from the end of the chain were averaged over to obtain the final parameter values.

5.2.5 Sampling Using the Weighted User-topic Distribution

As noted in the *Ranking Resources* section above the intuition is that while giving equal weight to both the resource and user distributions within the models may work well for tag suggestion, this approach may not work quite so well for ranking resources. In this case we can expect the resource to convey more information about itself than the users who are annotating it, therefore in the ranking formulas a weight, π_u, is introduced on the user distribution to account for this. However the assumption that both the resource and user distributions are equally important is still being made in the sampling. Unfortunately incorporating such a weight into the sampling by simply raising the user distribution to a power will not have the same effect as it does in the ranking formula. This is because, in the experiments conducted, the Gibbs sampling routine still eventually tended towards the non-weighted full conditional distribution over successive iterations. Since the algorithm is always averaging over multiple samples from the full distribution it simply takes slightly longer to converge.

The solution to this problem is to only sample using the user distribution, ψ_u, on every k^{th} sample. By averaging over a large number of samples from the end of the chain this approximates a weight of $\frac{1}{k}$. In all of the experiments the parameter k was set to 5, resulting in an effective weighting of 0.2. This was found to have very little impact on the convergence time of the chain and has

Model	S@1	S@5	S@10	MRR@10
SMatch	0.0555	0.1372	0.1860	0.0900
BM25	0.1701	0.2975	0.3376	0.2238
BayesLM	0.1819	0.3299	0.3772	0.2440
LDA	0.1994	0.3397	0.3936	0.2579
TTM1	0.2030	0.3556$*$	0.4158$*$	0.2675$*$
TTM2	**0.2137†**	**0.3559$*$**	**0.4202$*$**	**0.2743†**

Table 5.2: Ranking performance of all models on the test data set

the added benefit of slightly reducing the average computational complexity of the sampling.

5.3 Results

Table 6.1 shows the results of the ranking experiments for all of the models, for all of the topic models the number of topics is set at 250. $*$ indicates the result is significantly better than LDA (paired t-test, 95% confidence, $p < 0.05$), † indicates the result is significantly better than TTM1 and LDA (paired t-test, 95% confidence, $p < 0.05$). Between the more "conventional" ranking methods the language model with Bayesian smoothing has the best overall performance and considering its relative simplicity, it performs very well. BM25 is clearly less suited to this kind of data than it is to more normal documents and the SMatch algorithm - unsurprisingly - returns particularly poor results.

Comparing the "conventional" models with the topic models results show that over all metrics the topic models perform significantly better than the baselines. This is in contrast to results from previous work into ranking using topic models [WC06] and perhaps highlights the difference between the "documents" constructed from social tagging data and much longer real-world documents more commonly discussed in IR literature. In the case of social tagging data, the topic model's generalisation of the data and ability to deal with some of the vocabulary problems noted earlier are much more beneficial than perhaps they are with more normal corpora.

Comparing the 3 topic models it is clear that both personalised models are able to outperform the unpersonalised LDA baseline. TTM1 outperforms LDA by a statistically significant margin on all but one of the metrics whereas TTM2 outperforms it significantly over all measures. Between the 2 personalised models TTM2, with its clearer and more straightforward modelling assumptions and ranking formula, is able to outperform TTM1 over all measures (and as a result also significantly outperforms LDA). TTM2 is able to outperform TTM1 by a significant margin on both S@1 and MRR@10 which, considering the task at hand (ranking of resources), are arguably the most important metrics. This is because a better Mean Reciprocal Rank indicates that the model is able to rank the relevant resources higher more often where the user is most likely to see and therefore click on them. This is confirmed by the significant improvement in S@1 score where TTM2 is more able to identify the relevant resource as being most likely given the user and query on the first attempt.

5.3.1 Varying the Number of Topics

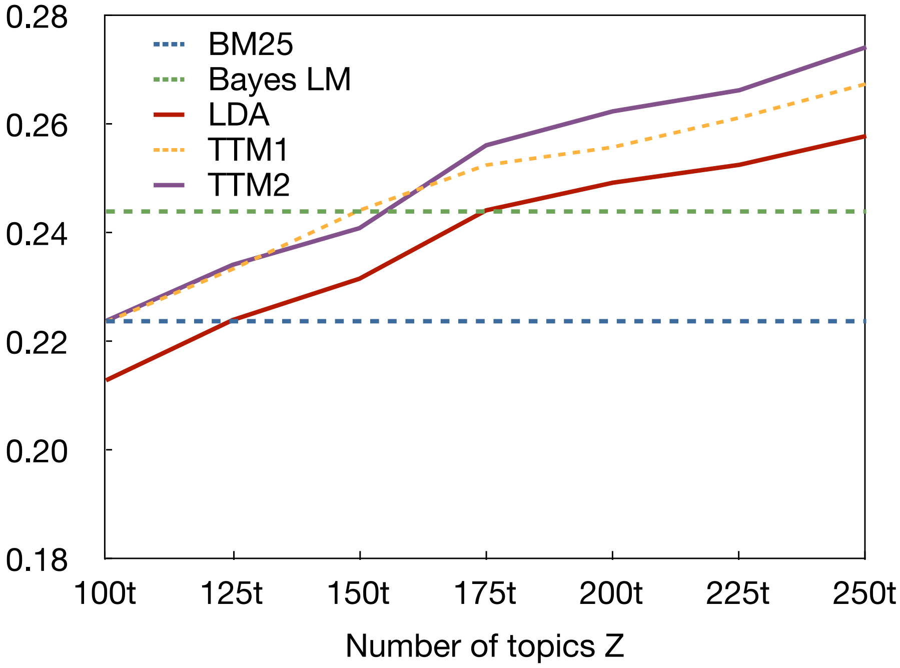


Figure 5.1: MRR@10 over varying numbers of topics

When using hidden topic models an important consideration is how com-

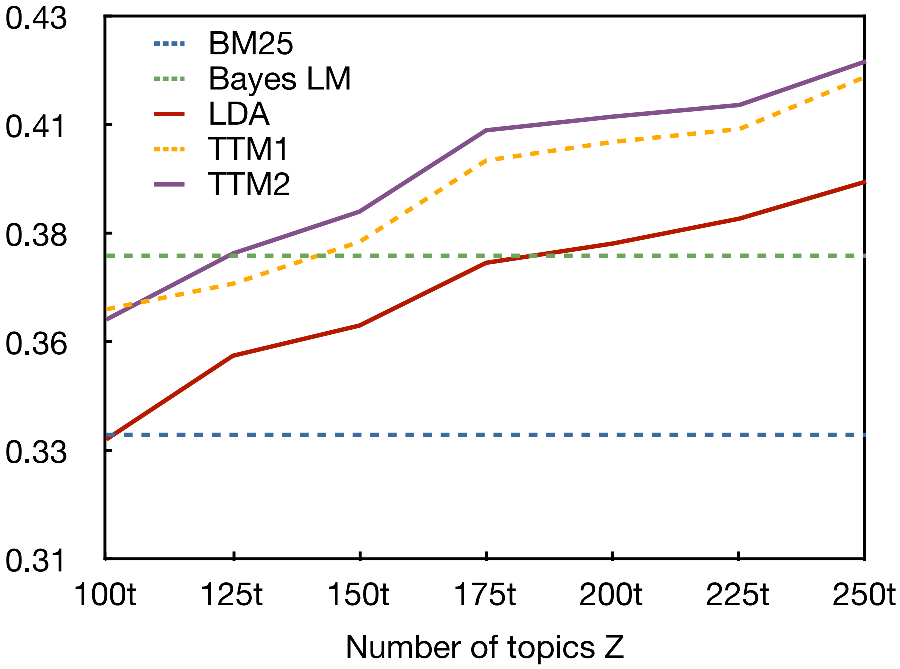


Figure 5.2: S@10 over varying numbers of topics

plex a model should be used in terms of the number of latent topics. Each model (in this case LDA, TTM1 and TTM2) can in fact be viewed as being a class of an infinite number of different models, where the complexity in number of topics is in the range $\{1, \ldots, \infty\}$. There has been a considerable amount of work published on so called non-parametric processes where the best model is inferred automatically based on the training data, the most appropriate for this work being Dirichlet Processes [TJBB06]. However these processes add significant further complexity and as such it is generally acceptable to use empirical methods to choose the most optimal parameterisation.

This work does not seek to optimise the models in terms of held-out likelihood but in terms of retrieval performance where these techniques may not be as appropriate. We would expect improvements in the held-out likelihood to taper off before improvements in retrieval performance do. Therefore parameters were estimated for the 3 topics models over different numbers of topics to see how retrieval performance was effected. Figures 5.2 and 5.1 show the results for the metrics Success@10 and MRR@10 for the 3 topic models over the range of topics from 100 to 250 with increments of 25. We also show the results from the 2 most competitive non-topic model baselines to allow direct comparison, SMatch is omitted from the figure as its performance is considerably worse than all the other models.

One can see quite clearly from the figure that as the number of topics is increased, the performance also increases, not a particularly surprising result. There appears to be a slight tailing off of performance improvement as the number of topics increases, however it is apparent that even better ranking performance could be achieved if the number of topics were to be increased even further. The increase in topic counts was stopped at 250 due to time constraints and because by this point it was clear that the topic models were outperforming all of the baselines. There is no reason why in principle the topic count couldn't keep increasing, however it can be expected that at some point performance would peak and the models could then be in danger of overfitting. Furthermore when using such systems a balance should be made between model complexity in terms of topics and ranking performance, since the amount of time required to rank resources using the models is linear in the number of topics.

Comparing between models, the data indicates that LDA needs approximately 175 topics before it begins to outperform BayesLM whereas the 2 personalised models only need somewhere between 125 and 150 topics, showing the advantage of incorporating the extra user data. The 2 personalised models have similar performance profiles over topics, however it appears that TTM2 begins to generally outperform TTM1 once it has enough topics to work with. This trend is particularly clear in the MRR figure where we can see that the 2 models only begin to diverge at around 175 topics and are fairly similar before this point.

5.3.2 Do We Have Enough Data?

As noted earlier in this chapter, due to restrictions imposed by the delicious public API, a maximum of only 100 bookmarks per user was available for download. Once all singleton resources and tags had been removed from the data set this left a fairly small profile for each user on which to build interest

	S@10		MRR@10	
Model	**0-60**	**60-80**	**0-60**	**60-80**
SMatch	0.1707	0.1667	0.0815	0.0811
BM25	0.3232	0.3271	0.2098	0.2180
BayesLM	0.3624	0.3776	0.2344	0.2291
LDA	0.3694	0.3941	0.2212	0.2534*
TTM1	0.3705	0.4175*	0.2361	0.2700*
TTM2	0.3719	0.4454*	0.2394	0.2804*

Table 5.3: Ranking performance over user profile size

profiles over the topic space (an average of 59.4 bookmarks per user). To investigate how performance was impacted by the size of the user profiles users were classified based on the number of resources they had bookmarked in the training data into two classes. One class containing users who had between 0 and 60 resources in their profiles and the other containing users with between 60 and 80 resources. Table 5.3 shows the results of this analysis. * indicates 60-80 class significantly different from 0-60 class ($p < 0.05$).

The results show, unsurprisingly, that the non-topic model baselines do not benefit from having more information about the user. There is no significant difference in results between the 2 bins for SMatch, SM25 or BayesLM. In contrast, all of the topic models appear to show better performance when ranking resources for users with longer profiles. For LDA, the difference between the S@10 values for the 2 bins is not significant, however for the MRR@10 metric it is significantly different.

This effect is far more pronounced in the personalised models, particularly TTM2 where the increase in both measures is very large when it has more information about the user. In fact the difference in performance between the 2 bins over both metrics for both personalised models is significant. This indicates that these models would perform even better if more information was available for the users, which would be the case were these techniques to be utilised on a live system. Note that results are not reported from users with more than 80 resources as it only covers a very small percentage of the total users (103 out of 9587).

Topic 1	Topic 2	Topic 3	Topic 4
france	car	cat	zoo
paris	cars	cats	animals
brittany	bike	kitty	animal
belgium	motorcycles	cute	nature
alps	japan	kitten	wildlife
europe	motorcycle	pet	bird
geneva	chiba	animal	birds
chamonix	auto	pets	fish
switzerland	motorsports	animals	aquarium
bourges	automobiles	feline	monkey

Topic 5	Topic 6	Topic 7	Topic 8
nyc	old	portrait	italy
newyork	window	selfportrait	travel
newyorkcity	rust	face	europe
manhattan	sign	people	rome
ny	wall	self	vacation
new	door	girl	italia
york	decay	eyes	roma
bigapple	shadow	me	holiday
brooklyn	blue	smile	florence
usa	car	man	firenze

Table 5.4: Most probable terms for 8 topics

5.3.3 Analysis of Topics

One of the key motivations for using topic models (and dimensionality reduction in general) is their ability to uncover relationships between terms and use these relationships to improve search results. As noted previously, in social tagging data there are many cases of polysemy and synonymy either due simply to word choice or due to people tagging in different languages or with different levels of knowledge. It is these significant and frequently observed differences in word choice combined with the short length of descriptions that make simple search algorithms less effective for social tagging data.

Table 5.4 shows the most probable terms (tags) for 8 different topics uncovered by the TTM2 topic model. Manual analysis of these topics suggests that they are extremely cohesive and that the model is indeed able to group together related terms and in doing so uncover the semantic meanings of those terms. Note that this cohesiveness is not necessarily expected as the models simply seek to minimise predictive error by maximising the likelihood of the data, a process that may not result in "topics" easily identifiable by human assessors. However, the fact that these models are able to return such coherent word lists is certainly advantageous.

The top terms in topic 8 show several examples of this kind of semantic grouping or clustering. For example the model has been able to group together the same words from different languages, in this case English and Italian, i.e. the pairs *Italy* - *Italia*, *Rome* - *Roma* and *Florence* - *Firenze*. It has also uncovered the important synonym relationship between the British English word 'holiday' and the US English equivalent 'vacation'. Similarly in topic 5 we can see that many different synonyms for New York have been clustered together. It is notable that all of these terms are never used to annotate a single resource, indicating that the model is able to uncover 2nd order co-occurence relationships. Other topics display evidence of basic level variation and synonymy, particularly topics 2 and 4 where a variety of related tags have been appropriately clustered together. Polysemy also appears to have been dealt with well in the example topics with the tags 'europe', 'animal' and 'animals' having high probability over more than one topic.

5.4 Conclusions

Chapter 2 discussed the problems facing ranking algorithms when dealing with social tagging data and Chapter 3 proposed the use of hidden topic models to deal with its inherent sparsity and vocabulary ambiguity. This highlighted the two most prominent issues resulting from this kind of data and indicated how such models might be able to at least partially overcome these obstacles. Reference to related work shows that topic modelling has been successfully used in this area in the past, however it has not been used to provide personalised search results based purely on tagging data.

Results from the experiments in this chapter showed that for social tagging data, the topic modelling approaches provided better resource rankings than even the most competitive baselines and outperformed them all by a statistically significant margin. They also demonstrated that the personalised tagging topic models were able to effectively leverage the extra user information to present better rankings than the unpersonalised LDA model. Over all measures the TTM2 model was able to significantly outperform LDA and was able to significantly outperform the less parsimonious TTM1 model on 2 key metrics. Further analysis of the results indicated that the performance of the tagging topic models could be improved further, relative to the other systems, if more data could be obtained for each user.

Chapter 6

Experiment 3: Collaborative Filtering

This experiment uses a large sample of ratings from a popular movie ratings web site to train the collaborative rating models introduced at the end of Chapter 3. The output from these models is used to attempt to predict unobserved ratings given by users. These predicted ratings are based on a number of biases as outlined in chapter 3 and the primary goal is to minimise the error of the predictions in the least-squares sense.

6.1 Predicting Ratings

The prediction problem is best described by saying that we would like to "fill in" the original sparse ratings matrix, extrapolating (or predicting) a rating $\hat{r}_{um}$ for every possible user-item pair from the limited data available. More practically we wish to define some function or model which will minimise the prediction error over the test data. Two metrics are commonly used to determine this average error; the Root Mean Squared Error (RMSE) and the Mean Average Error (MAE).

$$RMSE = \sqrt{\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (r_i - \hat{r}_i)^2} \tag{6.1}$$

$$MAE = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} |r_i - \hat{r}_i| \tag{6.2}$$

The RMSE is commonly used in statistics for measuring the difference between the set of values predicted by a model and the values actually observed from the system being modelled. It is a good measure of precision and is an unbiased estimator of the standard deviation of the predictions. Furthermore the measure assumes that errors are drawn from a Gaussian distribution which is the same assumption generally made in collborative filtering models, it is therefore a good choice of error function. The MAE is simply the mean absolute difference between the predicted rating and the actual rating, over the whole test set. We report both metrics as they provide different information regarding the performance of predictions: the RMSE penalises large errors much more than small errors while the MAE penalises all errors equally relative to their size.

6.2 Experimental Set-up

I now discuss the experiments performed on a large sample of rankings data from the MovieLens[1] movie rating web site, this data is freely available from the GroupLens website.[2] The data consists of 10 million ratings for 10,681 movies made by 71,567 users. The users are selected at random and have all rated at least 20 movies. Consequently the average number of ratings per user is 140 and per movie is 936. The ratings are all given on a scale of 0 to 5 stars with increments of 0.5 stars. The mean rating over all users and movies is 3.53 and the variance is 0.96.

This data set is separated into training and test sets by randomly choosing k percent of the ratings for each user to be kept for testing and used the remaining ratings to train the models. Doing so ensures that the test data is distributed over users in the same way as the training data. For these experiments k is set to be 20%. The results reported are based on predictions over all of the

[1] http://www.movielens.org/

[2] http://grouplens.org/node/73

test data, amounting to almost 2 million individual predictions. To evaluate the relative performance of the various models the evauations report both the RMSE and also the MAE as described above.

6.2.1 Baselines

In order to evaluate the utility of the new models it is important to choose suitable baseline methods with which to compare their performance. In this work the LITR methods are compared to 3 baselines from CF literature:

mean-r a naïve, simple baseline which returns the mean rating as an estimate for all user-movie pairs.

neighbourhood a nearest-neighbour method using Pearson correlation coefficient as the similarity metric with case amplification and significance weighting [BHK98]. This method represents a best performing example of earlier memory-based systems.

SVD a Singular Value Decomposition (SVD) model with user and movie biases providing a thoroughly modern and highly competitive baseline. This model is similar to the LITR models in that it reduces the dimensionality of the original ratings matrix down via the use of latent factors, however it is neither generative nor Bayesian. The decomposition of the matrix into latent factors is achieved via a gradient descent optimisation routine with added regularisation terms to reduce the likelihood over the model over-fitting the data. As in the LITR models, this gradient descent optimisation is interspersed with iterative fixed-point optimisations for the individual user and movie biases [Pat07].

6.2.2 Parameter Settings and Sampling

For the LITR models the concentration parameters of both α and β were set to 5, providing some light smoothing to the user-interest and item-topic distributions. The settings for ρ and σ were 0.5 and 0.1 respectively. The models

	Prediction error		**Improvement**	
Model	**MAE**	**RMSE**	**MAE**	**RMSE**
μ **rating**	0.8516	1.0521	-	-
n'hood	0.6582	0.8481	22.7%	19.4%
SVD	0.6516	0.8401	23.5%	20.1%
LITR1	0.6496	0.8384	23.7%	20.3%
LITR2	**0.6334**$*$	**0.8236** $*$	**25.6%**	**21.7%**

Table 6.1: Comparison of best results from each model

were not particularly sensitive to parameter values, provided excessively low or high values were not chosen.

For the SVD method the parameter values were optimised based on performance over a small sub-sample of the test set. The values obtained in doing this are very similar to the standard best performing parameters values as described in the literature [Pat07]. Specifically the learning rate was set to 0.002 and the 2 regularisation constants λ and $\lambda 2$ were set to 0.02 and 0.05. For the gradient descent algorithm prediction errors on a sub-sample of the test set were observed to stabilise after approximately 30 iterations, however to ensure convergence the process was allowed to continue until 50 iterations had elapsed. For the neighbourhood method the number of neighbours used for the estimates was set to 100.

Sampling in the LITR models is achieved via the use the Rao-Blackwellised Gibbs sampler [GS04] outlined in Chpater 3. For both models the chain was run for 300 iterations in total, as this appeared to consistently give good convergence in terms of model likelihood. The first 200 samples from the chain were discarded as "burn-in" and the remaining 100 samples from the end of the chain were averaged to obtain the final parameter values.

6.3 Results

The results from these experiments are summarised in Table 6.1. For latent factor/variable models the number of latent variables is set to 50, percentages indicate improvement over baseline. $*$ indicates a result significantly better than both LITR1 and SVD ($p < 0.05$), emboldened results are the best for that metric over all models. The results show that all of the methods significantly outperform the most simple choice of estimate: the mean over all ratings. The nearest-neighbour method performs surprisingly well, however the more modern model-based approaches are all able to outperform it by a large margin. During testing one of the main disadvantages of memory-based approaches was encountered as prediction using the neighbourhood model took orders of magnitude longer than any of the model-based approaches.

Comparing only the model-based approaches, the LITR1 model, which does not include individual biases for each user and item, is still able to outperform the SVD method, however not by a significant margin. The more complex LITR2 model on the other hand, which is able to leverage predictive power from the user and item biases as well as from the latent variable mixture of Gaussians, is able to outperform all of the other methods over both reported metrics by a statistically significant margin. In terms of MAE the LITR2 model outperforms SVD by 2.7%, and by 2% in terms of RMSE (paired t-test, 99% confidence, p-value = $4.5*10^{-05}$ and $1.2*10^{-05}$). Furthermore it improves upon the nearest neighbour approach by 3.8% for the MAE metric and 2.9% for the RMSE metric.

6.3.1 Varying the Number of Latent Factors

As with all reduced-dimensionality models the number of latent dimensions in the SVD baseline and both of the LITR model is an important factor and it is therefore sensible to look at how performance of the models vary as the number of topics is increased. By referring to the chart in Figure 6.1 we can see quite clearly that all of the model-based approaches increase in performance as the number of latent dimensions is increased. Notice that when the number of latent topics is set to 5 only LITR2 is able to outperform the memory-based nearest neighbour model, however as the number of factors is increased all of the model-based approaches begin to outperform it as the larger numbers of topics increase their flexibility in describing the relationships within the data.

Initially the performance of the LITR1 model appears to be quite poor

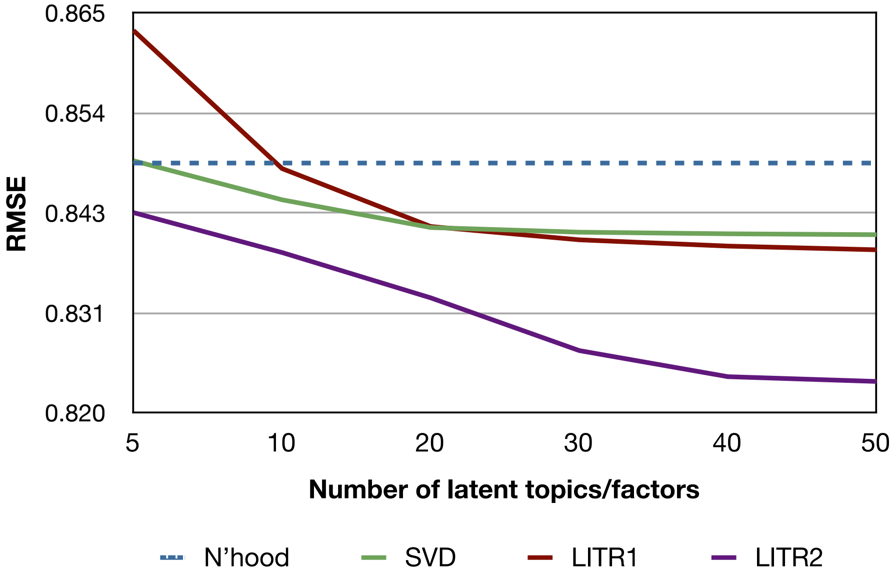


Figure 6.1: RMSE over different numbers of latent topics/factors

in comparison to the other latent variable models. This is because when the number of latent variables is small both SVD and LITR2 can rely on user and item biases to improve the prediction while LITR1 cannot. As the number of factors increases the performance of LITR1 approached and then eventually exceeds that of SVD, however it is still unable to come close to the performance of LITR2. The performance of all of the models appears to have reached a plateau by around 40 factors with any further improvements after this point being quite small. This is likely approaching the limit of how much of the variation within the data can be explained away via the reduced dimensionality spaces. Notice also that it appears that the extra modelling flexibility afforded by LITR2 allows its performance to continue to increase over a larger number of latent dimensions than either SVD or LITR1.

6.3.2 Performance for "Difficult" Users

As discussed earlier, an important consideration for any collaborative filtering algorithm is how well it is able to perform in the most difficult cases. Situations leading to difficult rating predictions are generally due to users or items with a very small number of ratings. Analysing the data set reveals that this situation

	Prediction error (RMSE)			
	≤ 20	≤ 50	≤ 100	all
SVD	0.9115	0.8840	0.8692	0.8401
LITR2	0.8536	0.8435	0.8379	0.8236
# users	9,404	33,965	50,297	71,567

Table 6.2: Comparison of results over different user profile sizes

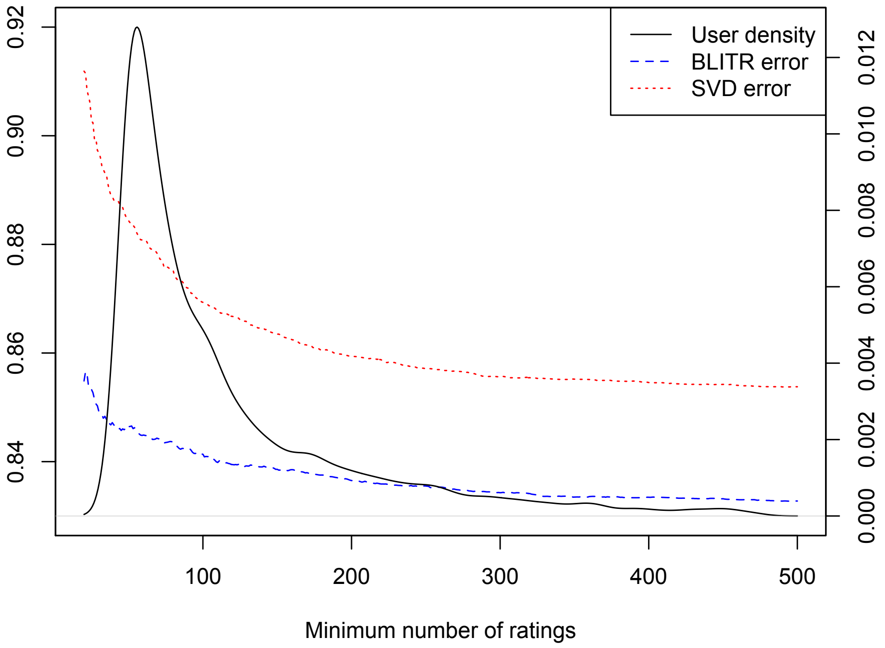


Figure 6.2: Prediction error and user count for varying profile sizes

is quite common; 13.4% of all users have 20 or fewer ratings and nearly half (47.5%) have 50 or fewer. Therefore it is expected that these will be the users for whom the algorithms struggle the most to make accurate predictions for.

Table 6.2 shows how the performance of the best performing baseline and the best of the new models (SVD and LITR2) vary over different user profiles sizes. The results show that the LITR model performs much better for smaller profiles (relative to its performance over all users) than SVD. The SVD model's performance decreases by 8.4% when dealing with small user profiles (20 or fewer ratings) whereas LITR2's performance only sees a decrease of 3.6%.

This result is perhaps illustrated more clearly in Figure 6.2, which shows the mean error over varying user profile sizes, for all users with a profile size smaller than or equal to the value on the x-axis. This is plotted for both SVD (dotted red line) and BLITR (dashed blue lines). The figure also shows on the right-hand y-axis the density of user counts over profile sizes (solid black line). We can see clearly that a large proportion of the users have a small number of ratings with very, very few having a large number. The maximum number of ratings for any user is 2876, 97% of users have fewer than 500 ratings and the minimum is 10 (this lower limit is imposed by the MovieLens web site). We can see from this plot that SVD's error for users with small profiles is quite high and that it fairly rapidly decreases as the profile size increases. On the other hand LITR has much smaller error for users with small profiles and is able to produce much smaller errors than SVD over the whole range of profile sizes.

This is an important outcome as it proves that the new LITR models perform much better when data is particularly sparse which is the most common case and the situations for which we are most interested in improving performance. This is likely to be at least partially a direct consequence of the Bayesian nature of the models; allowing them to cope better when there is little data available to base predictions on. It may also be because the LITR models are better at leveraging the limited information obtained from the small number of ratings that are available in these cases.

6.3.3 Variance of errors

The main focus of rating prediction is of course to make predictions with minimal error, however of course there will always be some error and it is not possible to always make perfect predictions. This being the case, a secondary focus is to try to ensure that when errors are made that they are not too large as this can frustrate and confuse users and even a single instance of poor prediction can cause a user to lose faith in the system's abilities. Figure 6.3 shows a density plot of the errors over the testing data made by both SVD and the LITR2 model. The plot shows that the errors made by SVD have larger variance (0.065 versus 0.045 for LITR2) and also have a much thicker tail at

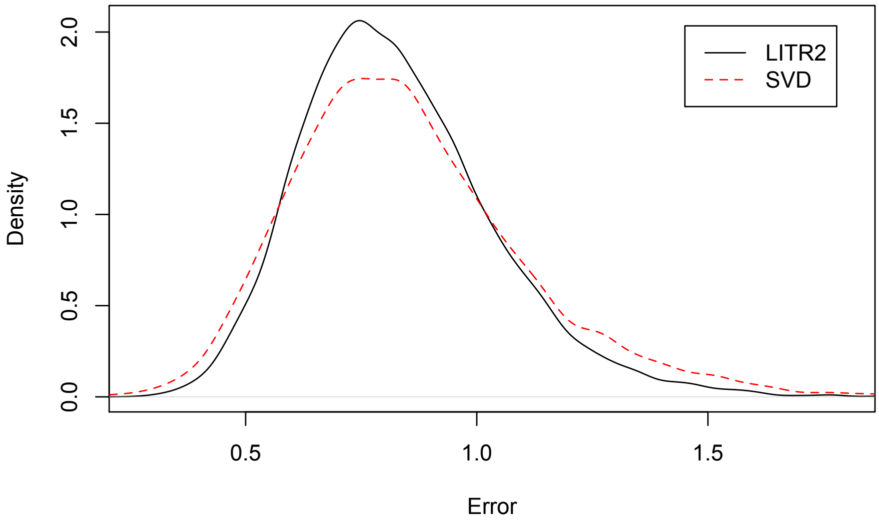


Figure 6.3: Density plot of errors

the higher end of the errors. This means that not only are the predictions made by LITR2 better in the expectation but they are also less likely to be extreme and as a result are less likely to frustrate users. It is also interesting to note from this plot that the errors are indeed Gaussian in nature, validating the assumptions made about errors in the models. This is confirmed by highly significant Pearson chi-square normality tests ($p \ll 0.0001$ for both SVD and LITR2).

6.4 Conclusions

This chapter has shown that the collaborative filtering models described in Chapter 3 are extremely competitive, with the extended model significantly outperforming the most competitive baselines. In comparison to more traditional methods (represented in this experiment by the neighbourhood baseline) it was found that the newer model-based methods outlined in this thesis represent a significant improvement in terms of both prediction accuracy and also computational complexity and prediction time. Investigation of how well the strongest baseline and the best of the two LITR models performed in cases where the user profiles were very short (where the user had rated very few movies) showed that the latter is able to cope far better. This is an important result as analysis of the data set showed this situation to be common and is where an improvement in performance is most noticeable to the user. Furthermore analysis of the residual errors showed that LITR2's errors had much smaller variance than those of SVD and as such it is much less likely to generate extremely erroneous predictions which could frustrate and confuse users.

Chapter 7

Conclusions

> "We can only see a short distance ahead, but we can see plenty there that needs to be done"
>
> *Alan Turing*

In this chapter I provide an outline of the main structure of the thesis and the main contributions of this work to the fields of Information Retrieval and Machine Learning and in particular to the narrower but growing field of social web search. I conclude the chapter and indeed the thesis by suggesting avenues of future work opened up by this research and consider how the models presented may be improved upon to give better results or to speed up computation.

The original aims of the work conducted throughout this thesis were to design models that could allow us to gain a better understanding of the complex dynamics of social data. Particularly by gaining some understanding of the semantic meaning of tags and of user interests to allow for personalisation. Chapter 1 began these investigations by discussing the main themes of social and collaborative data on the web; where it comes from, how it is structured, why it is useful and the motivations people have for actually taking the time to contribute to such systems. I outlined a number of posited benefits of this new form of data and surmised that it might be useful as a cheap source of communally-validated metadata. However in investigating previous research performed on social data I outlined a number of potential pitfalls of using such data including the vocabulary problem, an issue familiar to anyone working in the field of Information Retrieval, but one that is exacerbated by the sparsity and unrestricted vocabulary in this new setting. I concluded the chapter by introducing collaborative ratings, another form of socially contributed data now common on the web and discussed its various similarities and differences when compared to tagging data.

In Chapter 2 I grounded the investigation of appropriate models by outlining three different problem areas on social web sites and commented on related work in these areas, suggesting where improvements and refinements could be made. These were identified as key areas of research where progress would improve the experience of users and help to validate the suitability of any proposed models. Chapter 3 provided a short introduction to the main techniques of Bayesian statistical modelling and moved on to discuss contemporary methods of data modelling based on the idea of latent topics. In the second half of this chapter I motivated and introduced a series of four new statistical models; two for social tagging data and two for collaborative ratings data. I described their structure, assumptions made in their development and showed how their parameters could be estimated using Gibbs sampling and iterative fixed-point optimisation methods.

Chapters 4, 5 and 6 described a series of three experiments motivated by the earlier investigations of the literature that used the models derived to provide personalised tag suggestion, personalised search and personalised rating prediction. The results from these experiments demonstrated the applicability of the models to these key problems and showed that they were able to consistently outperform competitive baseline methods. The experiments all indicated that the models perform particularly well in the presence of sparse and noisy data, a key attribute when working with socially generated data. This evidence validates the earlier choice of Bayesian latent variable models for these problem areas and clearly illustrates that such models are well suited to this kind of data, perhaps much more so than for more traditional documents. Closer scrutiny of the results revealed that these new models are less prone to excessive error and are therefore more consistently reliable than previous methods. Evaluation of the topics generated showed strong evidence of semantic grouping not possible from simple co-occurence analysis with many synonymous terms automatically clustered together within the topic space. Furthermore the topical analysis showed many examples of polysemous terms appearing with high probability in several topics, demonstrating the potential of these models for lexical ambiguity resolution.

Interestingly it appears from the results of the experiments that the cleaner generative process used for the second tagging model (TTM2) does indeed yield better performance than that of TTM1. This may be a result of this generative process being closer to how one might expect the data were originally generated or it may be a result of the alternate parameterisation over resources instead of over topics. Another distinct possibility is that this alternate model places more importance on the user as being the driving force behind the generation of the observed tags, rather than the content and general themes of the original document.

7.1 Future Work

The work presented in this thesis opens up a number of possible directions for future work and the models presented can serve as a solid foundation for more complex hierarchical models incorporating other forms of data or further improving the fit of the models without loss of generality. Here I discuss a few examples of potential future work made possible by the results of this thesis.

The experimental work presented in Chapters 4, 5 and 6 showed that these new models can be applied to problems involving small to medium scale data sets without requiring an unreasonable amount of computation time. However it should be noted that these models are quite complex and rely on iterative approximation methods and are therefore computationally complex and non-deterministic in their convergence times. This means that in cases of very large data sets of many millions of data points the model estimation procedures described in this thesis may not be appropriate. A useful advancement of this work would be to investigate methods to speed up or parallelise the model estimation procedure so that they can be used for large problems or for near-real-time applications. Recent work by Porteous et al. [PNI+08] has shown that Gibbs sampling methods for LDA can be sped up significantly without a detrimental effect on the quality of the resulting parameter estimates.

More information could be introduced into the models, for example it is likely that in many cases a user's interests, or more specifically their currently "active" interests, will vary considerably over time. While the form of social data is normally described as being tri-partite it is actually true to say that for each annotation event we generally have a fourth aspect to consider: the timestamp when the annotation was made. Some early work has been conducted into models that are able to construct representations of topical variability over time including the Topics over Time model [WM06] and Dynamic Topic Models [BL06]. However at the time of writing these models are still in a fairly early stage of development, but could potentially be combined with the personalisation aspects of the Tagging Topic Models to form personalised, temporally-sensitive models of social data.

One of the main assumptions made in the models presented in this thesis was that the priors on the topical distributions should be symmetric and uninformative, serving only to smooth estimates and provide some parameter tying but not contributing extra information. There prior distributions could instead be made to be informative by either optimising them as part of the parameters estimation process of using them to introduce prior knowledge into the models. We may know in advance the prevalence of vocabulary words in predefined topics, for example by exploiting data available from some categorised data source or we may be able to determine a user's rough preferences by consulting them or deriving this information from other sources. This extra data would likely improve the reliability of estimates, especially in cases of sparse data where the prior is more influential, and would likely improve the convergence behaviour of the sampler thus requiring fewer burn-in samples to reach the posterior.

The collaborative rating models made the simplification of assuming that the Normal distributions describing the biases had the same fixed variances. These variances could instead be estimated during the sampling process with the expectation that doing so would further improve the fit of the model and therefore reduce the error of its predictions. During the development of these models several versions were tested that included these optimisations but were found to lead to over-fitting and generally poor prediction performance. This suggests that further machinery may be necessary to successfully augment the models in such a manner. A further possibility would be to leverage the tag data provided about the movies in the latest release of the MovieLens data set to make up for some of the sparsity in the ratings data. It may be possible, for example, to use this information to create an informative prior over $P(y|m)$, replacing the uniform prior currently used by estimating a distribution over topical genres from the tag data.

7.2 Main Contributions

The main contributions of this work can be summarised as follows:

- An investigation into social tagging and collaborative rating systems and the data they can provide, including discussion of the relationships between these two forms of socially-generated data.
- The development of two families of Bayesian hierarchical models designed specifically for these two forms of social data including algorithms for estimation of their parameters. The models represent significant developments over earlier work by incorporating more information about the users and by introducing more flexibility.
- In the case of the most sophisticated ratings model (LITR2), this work introduces the development of a novel method for parameter estimation via interleaved Gibbs sampling and fixed-point optimisation methods.
- A series of three separate experiments to determine the performance, limitations and behaviour of these models when applied to data obtained from real-world sources including comparison with competitive contemporary baseline methods. In all cases the models designed in this thesis were found to outperform the baselines.

- Novel evaluation methods for evaluating models and algorithms for the problems of social tag suggestion and personalised search of social bookmarks. This is necessary as at the time of writing there are no standard test sets available for the evaluation of social tagging systems.

- A discussion of the behaviour of these models on the various data and problem types in the experiments including the behaviour and characteristics of the Gibbs sampling algorithms and a discussion of the cohesiveness and interpretability of their outputs.

In summation the work of this thesis serves to advance understanding of how best to deal with the new forms of human-generated information now available in abundance on the Internet. It has shown that despite the extremely sparse and noisy nature of this data inferences can still be made allowing for the extraction of useful information. However, it has also illustrated the difficulties encountered in dealing with data of this nature and the importance of careful and reasoned model selection. It is highly likely that as the web continues to grow and evolve it will become ever more social giving us the opportunity to exploit interactions such as tags and ratings to form a better understanding of the contributed resources and of the users themselves. However in order to do this we need machine learning systems that can operate on the raw data, adapting to it as it evolves, and produce useful outputs. The work of this thesis provides a solid grounding in new and robust methods able to fulfil these goals and shows examples of how they could be used in real social systems on the web.

Appendix A

Distributions and Properties

Before deriving the TTM2 model in Appendix B it is useful to have an understanding of the 2 distributions used (the multinomial and the Dirichlet) and also a couple of important properties of these distributions. The multinomial distribution is a generalisation of the binomial distribution to an arbitrary number of dimensions and therefore models the probability of success in trials with more than 2 possible outcomes, the number of possible outcomes being the dimensionality K of the distribution. Note that in the modelling of text, such as in the models used in this thesis, we are actually using the categorical distribution which is simply the multinomial distribution over a single observation. The distribution is parameterised by the vector $\mathbf{p}$ which denotes the probability of observing each event and takes in a vector $\mathbf{x}$ denoting the number of times each value x_i appeared in the observation. The probability mass function for the multinomial is as follows:

$$Multinomial(\mathbf{x};\mathbf{p}) = \frac{n!}{\prod_{i=1}^{K} x_i!}\prod_{i=1}^{K} p_i^{x_i}$$

where $n = \sum_{i=1}^{K} x_i$, i.e. the total count of observations.

When dealing with Bayesian models it is convenient to make use of conjugate priors wherever possible as it makes the mathematical derivation much more straightforward. A prior $p(\theta)$ is said to be the conjugate of a likelihood $p(x|\theta)$ if the posterior distribution $p(\theta|x)$ is of the same form as the prior. The

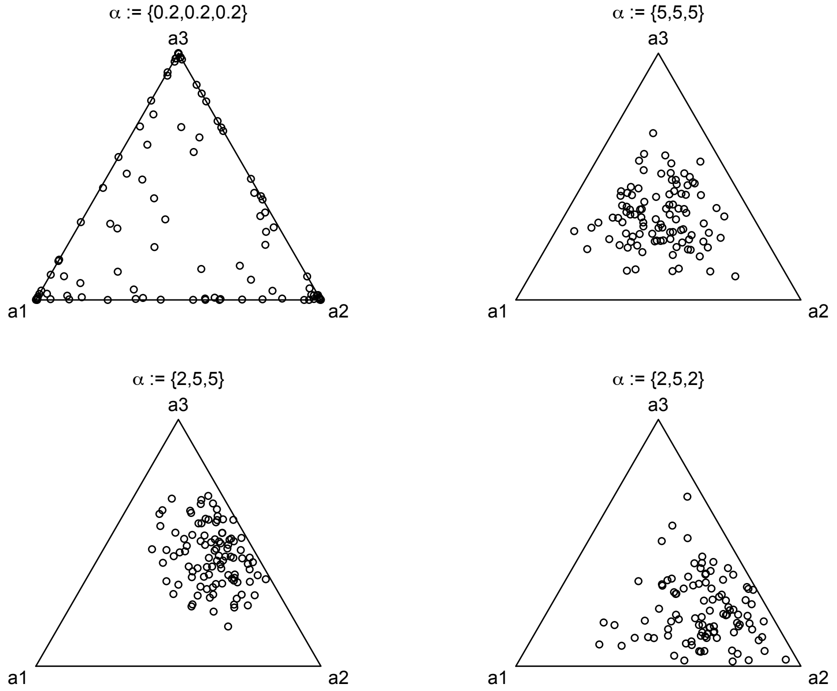


Figure A.1: Examples of random draws from 4 Dirichlet distributions

prior distribution for the multinomial is the Dirichlet distribution which unsurprisingly is a generalised Beta distribution and can be seen as a probability distribution over probability distributions. It is parameterised by a vector α of positive real numbers denoting that each possible outcome i has been previously observed $\alpha_i - 1$ times. It essentially returns the probability of a set of probabilities $\mathbf{x}$ given the parameters α which explains why it is a natural choice for a prior on the multinomial. Note that since it is itself a probability distribution, the vector $\mathbf{x}$ must sum to 1 over all i, that is $\sum_{i=1}^{K} x_i = 1$. Figure A.1 shows plots of 100 samples drawn from 4 different 3 dimensional Dirichlet distributions which are necessarily embedded on a $K - 1$ dimensional simplex where each point represents a 3D multinomial. Notice that if all α values are less than 1 then the distribution is in fact concave over the simplex and that as the α values increase the distribution becomes more centred, i.e. the probability mass become more equally spread across the dimensions. The probability density function for the Dirichlet is:

$$Dirichlet(\mathbf{p};\alpha) = \frac{1}{\mathrm{B}(\alpha)} \prod_{i=1}^{K} p_i^{\alpha_i - 1}$$

where the normalisation constant is the beta function which can be defined via gamma functions as:

$$\mathrm{B}(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)} \tag{A.1}$$

where $\Gamma(x)$ is the gamma function applied to x, this function is an extension of the factorial function, with its argument shifted down by one $(x-1)$, to real and complex numbers. This can be seen as necessary for normalising the Dirichlet by comparing the beta function to the normalisation constant for the multinomial remembering that the values of $\mathbf{x}$ in the Dirichlet are real numbers and not integers. Since the Dirichlet is conjugate to the multinomial, if we have a multinomial likelihood and a Dirichlet prior then the posterior will also be Dirichlet as follows:

$$\begin{aligned} p(\mathbf{p}|\mathbf{x},\alpha) &= \frac{1}{\mathrm{B}(\mathbf{x}+\alpha)} \prod_{i=1}^{K} p_i^{x_i+\alpha_i-1} \qquad \text{(A.2)} \\ &= Dirichlet(\mathbf{p};\mathbf{x}+\alpha) \end{aligned}$$

Given that equation A.2 is a valid probability distribution and all probability distributions must sum (in the case of discrete distributions) or integrate (in the case of continuous ones) to 1 then:

$$\begin{aligned} 1 &= \int \frac{1}{\mathrm{B}(\mathbf{x}+\alpha)} \prod_{i=1}^{K} p_i^{x_i+\alpha_i-1} \,\mathrm{d}\mathbf{p} \\ &= \frac{1}{\mathrm{B}(\mathbf{x}+\alpha)} \int \prod_{i=1}^{K} p_i^{x_i+\alpha_i-1} \,\mathrm{d}\mathbf{p} \end{aligned}$$

This implies that the following holds true:

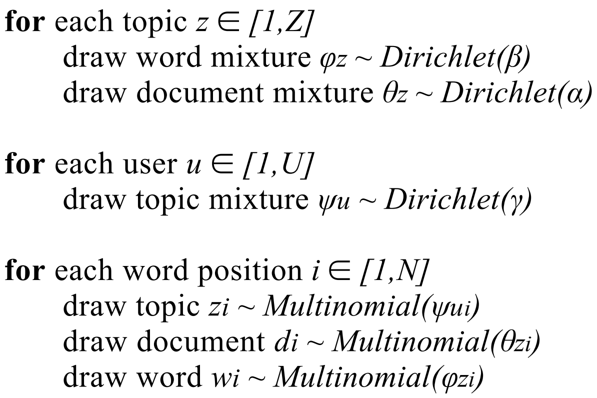
$$\int \prod_{i=1}^{K} p_i^{x_i+\alpha_i-1} \, \mathrm{d}\mathbf{p} = \mathrm{B}(\mathbf{x}+\alpha) \tag{A.3}$$

As we shall see, this is an extremely useful property and it allows us to reduce and solve seemingly very complex equations involving this posterior.

Appendix B

Derivation of TTM2

Before deriving the full likelihood and Gibbs sampling routing for this model it is worth referring to its generative process which is outlined in figure B.1.



```
for each topic $z \in [1,Z]$
    draw word mixture $\varphi_z \sim Dirichlet(\beta)$
    draw document mixture $\theta_z \sim Dirichlet(\alpha)$

for each user $u \in [1,U]$
    draw topic mixture $\psi_u \sim Dirichlet(\gamma)$

for each word position $i \in [1,N]$
    draw topic $z_i \sim Multinomial(\psi_{u_i})$
    draw document $d_i \sim Multinomial(\theta_{z_i})$
    draw word $w_i \sim Multinomial(\varphi_{z_i})$
```

Figure B.1: Generative model for Tagging Topic Model 2

To derive the complete model we begin by defining the joint distribution of the variables conditioned on their prior distributions, after doing this we proceed in the Bayesian style by integrating out the priors:

$$\begin{aligned} p(\mathbf{w}, \mathbf{z}, \mathbf{d} | \alpha, \beta, \gamma) &= p(\mathbf{w}|\mathbf{z}, \beta) p(\mathbf{d}|\mathbf{z}, \alpha) p(\mathbf{z}|\gamma) \\ &= \int_\Phi p(\mathbf{w}|\Phi, \mathbf{z}) p(\Phi; \beta) \, \mathrm{d}\Phi \int_\Theta p(\mathbf{d}|\Theta, \mathbf{z}) p(\Theta; \alpha) \, \mathrm{d}\Theta \\ &\quad \times \int_\Psi p(\mathbf{z}|\Psi) p(\Psi; \gamma) \, \mathrm{d}\Psi \end{aligned}$$

The above equation can be separated into 3 distinct parts that depend on Φ, Θ and Ψ respectively, since they are independent of each other, and as such they can be derived separately.

$p(\Phi;\beta)$ is comprised of Z Dirichlet distributions as follows:

$$p(\Phi;\beta) = \prod_{z=1}^{Z} p(\phi_z|\beta) = \prod_{z=1}^{Z} \frac{1}{\mathrm{B}(\beta)} \prod_{w=1}^{W} \phi_{z,w}^{\beta_w - 1}$$

and $p(\mathbf{w}|\Phi,\mathbf{z})$ has the following multinomial distribution:

$$p(\mathbf{w}|\Phi,\mathbf{z}) = \prod_{i=1}^{N} p(w_i|z_i) = \prod_{z=1}^{Z}\prod_{w=1}^{W} \phi_{w,z}^{N_{w,z}}$$

where N_w, z is the count of word positions where the word is w and the topical allocation is z. Now from the second part, the distribution of $p(\Theta;\alpha)$ is of the very same form as that of $p(\Phi;\beta)$, it is also distributed Dirichlet. Its likelihood can therefore be written as follows:

$$p(\Theta;\alpha) = \prod_{z=1}^{Z} p(\theta_z|\alpha) = \prod_{z=1}^{Z} \frac{1}{\mathrm{B}(\alpha)} \prod_{d=1}^{D} \theta_{z,d}^{\beta_d - 1}$$

The corresponding likelihood $p(\mathbf{d}|\Theta,\mathbf{z})$ is also very similar to that of the first part:

$$p(\mathbf{d}|\Theta,\mathbf{z}) = \prod_{i=1}^{N} p(d_i|z_i) = \prod_{z=1}^{Z}\prod_{d=1}^{D} \theta_{d,z}^{N_{d,z}}$$

Finally the third part of the equation, based on the distribution of topics given a user, $p(\Psi;\gamma)$ is also Dirichlet distributed:

$$p(\Psi;\gamma) = \prod_{u=1}^{U} p(\psi_u|\gamma) = \prod_{u=1}^{U} \frac{1}{\mathrm{B}(\gamma)} \prod_{z=1}^{Z} \psi_{z,u}^{\gamma_z - 1}$$

and the distribution of $p(\mathbf{z}|\Psi)$ is multinomial:

$$p(\mathbf{z}|\Psi) = \prod_{i=1}^{N} p(z_i|u_i) = \prod_{z=1}^{Z}\prod_{u=1}^{U} \psi_{z,u}^{N_{z,u}}$$

We can now take the product of the 3 parts and integrate over the priors:

$$\begin{aligned} p(\mathbf{w},\mathbf{z},\mathbf{d}|\alpha,\beta,\gamma) &= \int_{\Phi} \prod_{z=1}^{Z}\prod_{w=1}^{W} \phi_{w,z}^{N_{w,z}} \prod_{z=1}^{Z} \frac{1}{\mathrm{B}(\beta)} \prod_{w=1}^{W} \phi_{z,w}^{\beta_w - 1}\,\mathrm{d}\Phi \\ &\times \int_{\Theta} \prod_{z=1}^{Z}\prod_{d=1}^{D} \theta_{d,z}^{N_{d,z}} \prod_{z=1}^{Z} \frac{1}{\mathrm{B}(\alpha)} \prod_{d=1}^{D} \theta_{z,d}^{\beta_d - 1}\,\mathrm{d}\Theta \\ &\times \int_{\Psi} \prod_{z=1}^{Z}\prod_{u=1}^{U} \psi_{z,u}^{N_{z,u}} \prod_{u=1}^{U} \frac{1}{\mathrm{B}(\gamma)} \prod_{z=1}^{Z} \psi_{z,u}^{\gamma_z - 1}\,\mathrm{d}\Psi \end{aligned}$$

Combining like terms, using the properties of integration of a product and taking the products over Z and U out of the integrals, this can be rewritten as:

$$\begin{aligned} p(\mathbf{w},\mathbf{z},\mathbf{d}|\alpha,\beta,\gamma) &= \prod_{z=1}^{Z} \left(\frac{1}{\mathrm{B}(\beta)} \int_{\phi_z} \prod_{w=1}^{W} \phi_{w,z}^{N_{w,z}+\beta_w - 1}\,\mathrm{d}\phi_z \right) \\ &\times \prod_{z=1}^{Z} \left(\frac{1}{\mathrm{B}(\alpha)} \int_{\theta_z} \prod_{d=1}^{D} \theta_{d,z}^{N_{d,z}+\alpha_d - 1}\,\mathrm{d}\theta_z \right) \\ &\times \prod_{u=1}^{U} \left(\frac{1}{\mathrm{B}(\gamma)} \int_{\psi_u} \prod_{z=1}^{Z} \psi_{z,u}^{N_{z,u}+\gamma_z - 1}\,\mathrm{d}\psi_u \right) \end{aligned}$$

Using the property of multinomial distributions with Dirichlet priors outlined in appendix A in equation A.3 these integrals can be solved in closed form and thus simplified to the following:

$$p(\mathbf{w},\mathbf{z},\mathbf{d}|\ldots) = \prod_{z=1}^{Z} \frac{\mathrm{B}(\beta + \vec{Nw_z})}{\mathrm{B}(\beta)} \frac{\mathrm{B}(\alpha + \vec{Nd_z})}{\mathrm{B}(\alpha)} \prod_{u=1}^{U} \frac{\mathrm{B}(\gamma + \vec{Nz_u})}{\mathrm{B}(\gamma)} \quad \text{(B.1)}$$

where $\vec{Nw_z}$ denotes the vector over all words where each entry is the count of topic allocations z for that word. Likewise $\vec{Nd_z}$ is topic allocations over documents and $\vec{Nz_u}$ is users over topics. Bear in mind that each of hyperparameters

α, β and γ are also vectors, however in practise we are likely to simply use symmetric (uniform) distributions for these. To derive the appropriate Gibbs sampling equation for this problem we can use the chain rule of probabilities to obtain the full conditional probability:

$$\begin{aligned} p(z_i|\mathbf{w},\mathbf{d},\mathbf{z}_{-i};\alpha,\beta,\gamma) &= \frac{p(z_i,w_i,d_i|\mathbf{w}_{-i},\mathbf{d}_{-i},\mathbf{z}_{-i};\alpha,\beta,\gamma)}{p(w_i,d_i|\mathbf{w}_{-i},\mathbf{d}_{-i},\mathbf{z}_{-i};\alpha,\beta,\gamma)} \\ &\propto \frac{p(\mathbf{w},\mathbf{d},\mathbf{z}|\alpha,\beta,\gamma)}{p(\mathbf{w}_{-i},\mathbf{d}_{-i},\mathbf{z}_{-i}|\alpha,\beta,\gamma)} \end{aligned}$$

The numerator in the above equation is B.1 and the denominator is the same but with the counts from the ith position removed as follows:

$$p(\mathbf{w}_{-i},\mathbf{z}_{-i},\mathbf{d}_{-i}|\ldots) = \prod_{z=1}^{Z}\frac{\mathrm{B}(\beta+N\vec{w}_z^{-1})}{\mathrm{B}(\beta)}\frac{\mathrm{B}(\alpha+N\vec{d}_z^{-1})}{\mathrm{B}(\alpha)}\prod_{u=1}^{U}\frac{\mathrm{B}(\gamma+N\vec{z}_u^{-1})}{\mathrm{B}(\gamma)}$$

We can now expand all of the Beta functions as in A.1 to obtain the following:

$$p(z_i|\ldots) = \frac{\frac{\prod_{w=1}^{W}\Gamma(N_{w,z_i}+\beta_w)}{\Gamma(\sum_{w=1}^{W}N_{w,z_i}+\beta_w)}}{\frac{\prod_{w=1}^{W}\Gamma(N_{w,z_i}^{-1}+\beta_w)}{\Gamma(\sum_{w=1}^{W}N_{w,z_i}^{-1}+\beta_w)}} \cdot \frac{\frac{\prod_{d=1}^{D}\Gamma(N_{d,z_i}+\alpha_d)}{\Gamma(\sum_{d=1}^{D}N_{d,z_i}+\alpha_d)}}{\frac{\prod_{d=1}^{D}\Gamma(N_{d,z_i}^{-1}+\alpha_d)}{\Gamma(\sum_{d=1}^{D}N_{d,z_i}^{-1}+\alpha_d)}} \cdot \frac{\frac{\prod_{z=1}^{Z}\Gamma(N_{u_i,z}+\gamma_z)}{\Gamma(\sum_{z=1}^{Z}N_{u_i,z}+\gamma_z)}}{\frac{\prod_{z=1}^{Z}\Gamma(N_{u_i,z}^{-1}+\gamma_z)}{\Gamma(\sum_{z=1}^{Z}N_{u_i,z}^{-1}+\gamma_z)}}$$

Notice that if we sum $N\vec{w}_z^{-1}$ over all $z \in Z$ then it is exactly the same as $N\vec{w}_z) + 1$. This means that any factors where $z! = z_i$, $d! = d_i$ and $u! = u_i$ drop out and we get the following:

$$p(z_i|\ldots) = \frac{\frac{\Gamma(N_{w_i,z_i}+\beta_{w_i})}{\Gamma(\sum_{w=1}^{W}N_{w,z_i}+\beta_w)}}{\frac{\Gamma(N_{w_i,z_i}^{-1}+\beta_{w_i})}{\Gamma(\sum_{w=1}^{W}N_{w,z_i}^{-1}+\beta_w)}} \cdot \frac{\frac{\Gamma(N_{d_i,z_i}+\alpha_{d_i})}{\Gamma(\sum_{d=1}^{D}N_{d,z_i}+\alpha_d)}}{\frac{\Gamma(N_{d_i,z_i}^{-1}+\alpha_{d_i})}{\Gamma(\sum_{d=1}^{D}N_{d,z_i}^{-1}+\alpha_d)}} \cdot \frac{\frac{\Gamma(N_{u_i,z_i}+\gamma_{z_i})}{\Gamma(\sum_{z=1}^{Z}N_{u_i,z}+\gamma_z)}}{\frac{\Gamma(N_{u_i,z_i}^{-1}+\gamma_{z_i})}{\Gamma(\sum_{z=1}^{Z}N_{u_i,z}^{-1}+\gamma_z)}}$$

Finally, remember that these are count values, we can use the rule that for any y where y is a positive integer $\Gamma(y) = (y-1)!$ to derive the final Gibbs sampling equation:

$$p(z_i|\mathbf{w},\mathbf{d},\mathbf{z}_{-i};\alpha,\beta,\gamma) = \frac{N_{w_i,z_i}^{(-i)}+\beta\frac{1}{W}}{N_{z_i}^{(-i)}+\beta}\frac{N_{d_i,z_i}^{(-i)}+\alpha\frac{1}{D}}{N_{z_i}^{(-i)}+\alpha}\frac{N_{u_i,z_i}^{(-i)}+\gamma\frac{1}{Z}}{N_{u_i}^{(-i)}+\gamma}$$

Bibliography

[AKD07a] H.S. Al-Khalifa and H.C. Davis. Exploring the value of folksonomies for creating semantic metadata. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3(1):13–39, 2007.

[AKD07b] H.S. Al-Khalifa and H.C. Davis. Folksonomies versus automatic keyword extraction: An empirical study. Technical report, Learning Technology Research Group, ECS, University of Southampton, Southampton, UK, 2007.

[Ame07] Morgan Ames. Why we tag: motivations for annotation in mobile and online media. *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 971–980, 2007.

[AT05] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.

[BHK98] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI98)*, pages 43–52, 1998.

[BJ03] D. M. Blei and M. I. Jordan. Modeling annotated data. *Proceedings of the 26th annual international ACM SIGIR conference on*

Research and development in information retrieval, pages 127–134, 2003.

[BL06] David M. Blei and John D. Lafferty. Dynamic topic models. *Proceedings of the 23rd International Conference on Machine learning*, pages 113–120, 2006.

[BL07] D. M. Blei and J. D. Lafferty. A correlated topic model of science. *Annals of Applied Statistics*, 1:17–35, August 2007.

[BLCL+94] Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret. The world-wide web. *Commun. ACM*, 37:76–82, August 1994.

[Ble04] David Meir Blei. *Probabilistic models of text and images.* PhD thesis, Berkeley, CA, USA, 2004. AAI3183785.

[BM85] D C Blair and M E Maron. An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM*, 28(3):289–299, 1985.

[BNJ03] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2003.

[BP98] Sergey Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30:107–117, 1998.

[BP06] Eugene Barsky and Michelle Purdon. Introducing web 2.0: social networking and social bookmarking for health librarians. *JABSC*, 27:65–67, 2006.

[BS94] J.M. Bernardo and A.F.M. Smith. *Bayesian theory.* Wiley, New York, 1994.

[BWC07] Andrew Byde, Hui Wan, and Steve Cayzer. Personalized tag recommendations via tagging and content-based similarity metrics. *International Conference on Weblogs and Social Media (ICWSM)*, March 2007.

[BYRN99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval.* Addison Wesley Longman, 1999.

[CCL+09] Wen Y. Chen, Jon C. Chu, Junyi Luan, Hongjie Bai, Yi Wang, and Edward Y. Chang. Collaborative filtering for orkut communities: discovery of user latent behavior. *Proceedings of the 18th international conference on World wide web*, pages 681–690, 2009.

[CLP07] Ciro Cattuto, Vittoria Loreto, and Luciano Pietronero. Semiotic dynamics and collaborative tagging. *Proceedings of the National Academy of Sciences*, 104 (5):1461–1464, 2007.

[CW04] Y. Chen and J.Z. Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5:913–939, 2004.

[DDF+90] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[Dew76] M. Dewey. A classification and subject index for cataloguing and arranging the books and pamphlets of a library, 1876.

[DSW07] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. *Proceedings of the 16th international conference on World Wide Web*, pages 581–590, 2007.

[Fit06] D. Fitcher. Intranet applications for tagging and folksonomies. *Online*, 30 (3):43–45, 2006.

[FLGD87] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30 (11):964–971, 1987.

[GCSR04] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. London: Chapman and Hall, 2004.

[GG84] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[GH05] S. Golder and B. A. Huberman. The structure of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2005.

[GH06] S. Golder and B. Huberman. Usage patterns of collaborative tagging systems. In *Journal of Information Science*, volume 32(2), pages 198–208, 2006.

[GL97] D. Gamerman and H. F. Lopes. *Markov Chain Monte Carlo: stochastic simulation for bayesian inference*. London: Chapman and Hall, 1997.

[GNOT92] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35:61–70, 1992.

[GRGP01] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Journal of Information Retrieval*, 4(2):133–151, 2001.

[GS04] T. Griffiths and M. Steyvers. Finding scientific topics. *National Academy of Science*, 101:5228–5235, 2004.

[GW08] N. Garg and I. Weber. Personalized tag suggestion for flickr. In *The 17th international conference on World Wide Web (WWW)*, 2008.

[HBCR10] Morgan Harvey, Mark Baillie, Mark Carman, and Ian Ruthven. Tripartite hidden topic models for personalised tag suggestion. *Advances in Information Retrieval, 32nd European Conference on IR Research*, 2010.

[HBRE09] Morgan Harvey, Mark Baillie, Ian Ruthven, and David Elsweiler. Folksonomic tag clouds as an aid to content indexing. *2nd Annual Workshop on Search in Social Media (SSM)*, 2009.

[Hei08] G. Heinrich. Parameter estimation for text analysis. Technical report, Fraunhofer IGD, 2008.

[HJSS06] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. *Lecture Notes in Information Science*, 4011:411–426, 2006.

[HO06] Ben He and Iadh Ounis. Query performance prediction. *Information Systems*, 31, 7:585 – 594, 2006.

[Hof01] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196, 2001.

[Hof04] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, January 2004.

[Hoo65] R.S. Hooper. Indexer consistency tests—origin, measurements, results and utilization. Technical report, IBM, Bethesda, 1965.

[HRC10] Morgan Harvey, Ian Ruthven, and Mark James Carman. Ranking social bookmarks using topic models. *19th ACM Conference on Information and Knowledge Management (CIKM)*, pages 1401–1404, 2010.

[HRC11] Morgan Harvey, Ian Ruthven, and Mark James Carman. Improving social bookmark search using personalised latent variable language models. *Forth International Conference on Web Search and Web Data Mining (WSDM)*, pages 485–494, 2011.

[HRCC11] Morgan Harvey, Ian Ruthven, Fabio Crestani, and Mark Carman. Bayesian latent variable models for collaborative item rating prediction. *ACM 20th Conference on Information and Knowledge Management (CIKM)*, October 2011.

[HRM08] Paul Heymann, Daniel Ramage, and Hector G. Molina. Social tag prediction. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 531–538, 2008.

[HTF08] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2 edition, 2008.

[HU93] Colin Howson and Peter Urbach. *Scientific Reasoning: The Bayesian Approach.* Open Court Publishing Company, December 1993.

[Iiv95] Mirja Iivonen. Consistency in the selection of search concepts and search terms. *Information Processing and Management*, 31(2):173–190, 1995.

[JMH$^+$07] Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in folksonomies. In *Knowledge Discovery in Databases: PKDD 2007*, volume 4702 of *Lecture Notes in Computer Science*, chapter 52, pages 506–514. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2007.

[JMH$^+$08] Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in social bookmarking systems. *AI Communications*, 21(4):231–247, January 2008.

[KC06] M.E. Kipp and D. Grant Campbell. Patterns and inconsistencies in collaborative tagging systems: An examination of tagging practices. *American Society for Information Science and Technology*, 2006.

[KFN09] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent dirichlet allocation for tag recommendation. *Proceedings of the 3rd ACM conference on Recommender systems*, pages 61–68, 2009.

[KHS08] Beate Krause, Andreas Hotho, and Gerd Stumme. A comparison of social bookmarking with traditional search. *Advances in Information Retrieval, 30th European Conference on IR Research*, pages 101–113, 2008.

[KMM$^+$97] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40:77–87, 1997.

[Kor08] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

[Lan95] K. Lang. NewsWeeder: Learning to filter netnews. *12th International Conference on Machine Learning*, pages 331–339, 1995.

[Lan98] F. W. Lancaster. *Indexing and Abstracting in Theory and Practice.* University of Illinois, 1998.

[LB07] Julia Lasserre and Christopher M. Bishop. Generative or discriminative? getting the best of both worlds. *Bayesian Statistics*, 8:3–24, 2007.

[Lyn97] C. Lynch. Searching the internet. *Scientific American*, 276(3):52–56, 1997.

[LZ04] Jia Li and Osmar R. Zaïane. Combining Usage, Content, and Structure Data to Improve Web Site Recommendation. *5th International Conference on Electronic Commerce and Web Technologies*, pages 305–315, 2004.

[LZT09] Steffen Lohmann, Jürgen Ziegler, and Lena Tetzlaff. Comparison of tag cloud layouts: Task-related performance and visual exploration. *Human-Computer Interaction, Interact*, 2009.

[Mah08] Hosam M. Mahmoud. *Pólya Urn Models*. Chapman and Hall, 2008.

[Mar03] B. Marlin. Modeling user rating profiles for collaborative filtering. *17th Annual Conference on Neural Information Processing Systems (NIPS'03), 2003.*, 2003.

[Mat04] A. Mathes. Folksonomies - cooperative classification and communication through shared metadata. 2004.

[MBR98] Raymond J. Mooney, Paul N. Bennett, and Loriene Roy. Book recommending using text categorization with extracted information. *AAAI-98 Workshop on Learning for Text Categorization*, pages 49–54, 1998.

[Mis06] Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. *Proceedings of the 15th international conference on World Wide Web*, pages 953–954, 2006.

[MM07] David Mimno and Andrew McCallum. Organizing the OCA: learning faceted subjects from a library of digital books. *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 376–385, 2007.

[MNBD06] C. Marlow, M. Naaman, D. Body, and M. Davis. Ht06, tagging paper, taxonomy, flickr, academic article, toread. *Proceedings of the seventeenth conference on Hypertext and hypermedia*, 2006.

[Mor05] Peter Morville. *Ambient Findability What We Find Changes Who We Become*. O'Reilly Media, 2005.

[NmAYG+10] Michael G. Noll, Ching man Au Yeung, Nicholas Gibbins, Christopher Meinel, and Nigel Shadbolt. Telling expertise from spammers: Expertise ranking in folksonomies. *Proceedings of the 31st international ACM SIGIR conference on Research and development in information retrieval*, 2010.

[Pat07] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *KDDCup.07*, 2007.

[Paz99] Michael J. Pazzani. A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review*, 13(5):393–408, December 1999.

[PB97] Michael J. Pazzani and Daniel Billsus. Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*, 27(3):313–331, 1997.

[PL07] A. Plangprasopchok and K. Lerman. Exploiting social annotation for automatic resource discovery. *In Proceedings of AAAI workshop on Information Integration from the Web*, 2007.

[PLG11] Anon Plangprasopchok, Kristina Lerman, and Lise Getoor. A probabilistic approach for learning folksonomies from structured data. *Forth International Conference on Web Search and Web Data Mining (WSDM)*, pages 555–564, 2011.

[PNI+08] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. *Proceeding of the 14th*

ACM SIGKDD international conference on Knowledge discovery and data mining, pages 569–577, 2008.

[QCI04] L.M. Quiroga, M.E. Crosby, and M.K. Idling. Reducing cognitive load. *37th Hawaii International Conference on System Sciences*, 2004.

[RHMGM09] Daniel Ramage, Paul Heymann, Christopher D. Manning, and Hector Garcia-Molina. Clustering the tagged web. *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 54–63, 2009.

[Ric79] E. Rich. User modeling via stereotypes. *Cognitive Science*, 3, no. 4:329–354, 1979.

[RW03] K. Rodden and K. Wood. How do people manage their digital photographs? *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2003.

[Sch06] P. Schmitz. Inducing ontology from flickr tags. *Proceedings of the 15th international conference on World Wide Web*, 2006.

[SHG09] David H. Stern, Ralf Herbrich, and Thore Graepel. Matchbox: large scale online bayesian recommendations. *Proceedings of the 18th international conference on World wide web*, pages 111–120, 2009.

[Shi05] Clay Shirky. Ontology is overrated: Categories, links, and tags, 2005.

[Shi08] Clay Shirky. *Here Comes Everybody.* Penguin Group, 2008.

[Sin05] R. Sinha. A cognitive analysis of tagging, 2005.

[SKKR00] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John T. Riedl. Application of dimensionality reduction in recommender systems–a case study. *Proceedings of the ACM WebKDD Workshop*, 2000.

[SM95] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating "word of mouth". *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, 1995.

[SOHB07] Sanjay Sood, Sara Owsley, Kristian Hammond, and Larry Birnbaum. Tagassist: Automatic tag suggestion for blog posts. *International Conference on Weblogs and Social Media*, 2007.

[SR93] A. F. M. Smith and G. O. Roberts. Bayesian computation via the gibbs sampler and related markov chain monte-carlo methods (with discussion). *Journal of the Royal Statistical Society*, 55(1):3–23, 1993.

[SvZ08] Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. *Proceeding of the 17th international conference on World Wide Web*, 2008.

[TDL08] Jaime Teevan, Sasan T. Dumais, and Daniel J. Liebling. To personalize or not to personalize: Modeling queries with variation in user intent. *Proceedings of the 30th international ACM SIGIR conference on Research and development in information retrieval*, 2008.

[TJBB06] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[TT10] Sarah K. Tyler and Jaime Teevan. Large scale query log analysis of re-finding. *Proceedings of the third ACM international conference on Web search and data mining*, pages 191–200, 2010.

[VCJ10] David Vallet, Iván Cantador, and Joemon M. Jose. Personalizing web search with folksonomy-based user and document profiles. *Advances in Information Retrieval, 32nd European Conference on IR Research*, 2010.

[VG02] D. Vizine-Goetz. Classification scheme for internet resources revisited. *Journal of Internet Cataloging*, 5 (4), 2002.

[Voß07] Jakob Voß. Tagging, folksonomy & co - renaissance of manual indexing. *International Symposium of Information Science*, pages 234–254, 2007.

[Wal04] B. Walsh. Markov chain monte carlo and gibbs sampling. *Lecture Notes for EEB 581, version 26, April*, 2004.

[Wal06] Hanna M. Wallach. Topic modeling: beyond bag-of-words. *In Proceedings of the 23rd International Conference on Machine Learning*, pages 977–984, 2006.

[WC06] X. Wei and W.B Croft. Lda-based document models for ad-hoc retrieval. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, 2006.

[WCY$^+$10] Jun Wang, Maarten Clements, Jie Yang, Arjen P. de Vries, and Marcel J. T. Reinders. Personalization of tagging systems. *Information Processing and Management: an International Journal*, 460-1:58–70, 2010.

[WM06] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433, 2006.

[WZY06a] X. Wu, L. Zhang, and Y. Yu. Exploring social annotations of the semantic web. *Proceedings of the 15th international conference on World Wide Web*, 2006.

[WZY06b] Xian Wu, Lei Zhang, and Yong Yu. Exploring social annotations for the semantic web. *Proceedings of the 15th international conference on World Wide Web*, pages 417–426, 2006.

[XFMS06] Zhichen Xu, Yun Fu, Jianchang Mao, and Difu Su. Towards the semantic web: Collaborative tag suggestions. *Proceedings of the Collaborative Web Tagging Workshop at WWW Conference*, 2006.

[YGS07] Ching Man Au Yeung, Nicholas Gibbins, and Nigel Shadbolt. Mutual contextualization in tripartite graphs of folksonomies. *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference*, 4825:960–964, November 2007.

[Yng59] Victor H. Yngve. The feasibility of machine searching of english texts. *Proceedings of the International Conference on Scientific Information*, pages 161–169, 1959.

[ZD69] P. Zunde and M. E. Dexter. Indexing consistency and quality. *American Documentation*, 20(3):259–267, April 1969.

[ZK07] Yi Zhang and Jonathan Koren. Efficient bayesian hierarchical user modeling for recommendation system. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 47–54, 2007.

[Zol07] Alla Zollers. Emerging motivations for tagging: Expression, performance and activism. *Proceedings of the 16th international conference on World Wide Web*, 2007.