Personalised Access to Social Media

Maarten Clements

Personalised Access to Social Media

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 6 december 2010 om 12:30 uur

door

Maarten CLEMENTS

elektrotechnisch ingenieur
Geboren te Barendrecht

Dit proefschrift is goedgekeurd door de promotoren:
Prof. dr. ir. M.J.T. Reinders
Prof. dr. ir. A.P. de Vries

Samenstelling promotiecommissie:

Rector Magnificus	Voorzitter
Prof. dr. ir. M.J.T. Reinders	Delft University of Technology, promotor
Prof. dr. ir. A.P. de Vries	Delft University of Technology, promotor
Prof. dr. ir. G.J. Houben	Delft University of Technology
Prof. dr. M. de Rijke	University of Amsterdam
Prof. dr. ir. W. Kraaij	Radboud University Nijmegen
Prof. dr. A.P.J. van den Bosch	Tilburg University
Dr. M. Naaman	Rutgers University
Prof. dr. ir. H.J. Sips	Delft University of Technology, reservelid

ICT group
The research reported in this dissertation has been carried out in the Information and Communication Theory group of the Technical University of Delft. (2006-2009)

DMIR Lab
The research reported in this dissertation has been carried out at the Delft Multimedia and Information Retrieval Lab of the Multimedia and Signal Processing group of the Technical University of Delft. (2010)

I-Share
The research reported in this dissertation has been funded by I-Share, part of the national research program Freeband Communication.

ASCI
This work was carried out in the ASCI graduate school.
ASCI dissertation series number 218.

ISBN/EAN: 978-90-9025824-9

Cover: *Floating Heads* by Sophie Cave, photography by Maarten Clements, 2008, Glasgow.

Acknowledgements

Dankwoord

You are now reading a book that took me four years to write. I am simultaneously proud and anxious to give you this book. It seldom happens that one reveals his personal work to the public, including all its imperfections. I want to thank all of you for giving me the confidence to write this.

During my Ph.D. project I had the privilege to be supervised by two professors. Both of you have become more than supervisors and even more than colleagues, I have greatly enjoyed the last four years with you. Marcel, you were an important factor in my decision to start this project. I already enjoyed working with you on my master's thesis and it was good to know that I was in good company when I started my Ph.D. on a completely different topic. I truly admire your knowledge of the machine learning field and your skill to clearly visualise mathematical problems. Most of all I want to thank you for your ability to make me as productive as possible. You always kept a bird-eye view on the project and knew exactly when to push or relax me. In the last five years you have played an important role in my personal quest to find out what really motivates me in my work.

Arjen, your knowledge of the IR field has been incredibly useful for a retrieval rookie like me. With the many discussions in Delft and Amsterdam you have made me eager to learn and continue to work with great pleasure on data mining problems in IR. You do not only know the theory of IR, but also the people behind it. Because of your social network I have not been working on an island but felt part of a community from the start. Many thanks for your great passion and for helping me find my way in this interesting field.

For several chapters we have enjoyed the collaboration with others. Johan, many thanks for the inspiring discussions and your endless positive energy, I hope you'll never lose this and keep inspiring people to work on groundbreaking topics. Jun, you gave me a great start in the field of CF and language modeling theory, I enjoyed the short time we worked together. Roelof, Vanessa and Borkür, during the three months I spent in Barcelona I quickly lost count of the number of discussions we've had. I believe discussion defines the very core of research and you have been a truly inspiring team to work with. Pavel, unfortunately the time we worked together was way too short to execute all the ideas we discussed. Thanks for your help on Part III of this book. Jeroen, many thanks for your help on the efficient implementation of the peak finding algorithm used in Chapter 8.

After 5 years in the ICT group the people I have enjoyed working with have become too numerous to mention, but let's try anyway. I started in the bio-informatics group with my master's project. The great atmosphere really was one of my motivations to stay in the ICT group even though I would not be working directly with you. Ronald and Pien, as the *face behind the screen* you made office life a lot more fun. I can't imagine how many hours I have been staring in your direction. Many thanks for being there. Anja, Saskia, Laura, Ben, Robbert and Jan, it is just so much better when you are there, thanks for all the support. Inald, Arjen, Alan, Martha, Stevan, Carsten, Luz, Ewine, Yue, du Poppe und Christoph, it's great to be in your network of excellence. Thanks for the fun trips and fruitful collaboration. Marcel, Emile, Jeroen, Jeroen, Theo, Ewine and Cynthia, the organisation of the computer vision course was one of the most enjoyable obligations ever. Wouter and Saskia, in the last months you have been very important in providing the occasional brain reset, thanks for the enjoyable evenings.

All the Chinese, Turkish, South-African, cactus eating Mexican, Spanish, Italian, Polish, German and some more Dutch people I have not mentioned yet; I just want to thank you for being at one of the ASCI conferences, sports days, group trips or simply for drinking (bad) coffee or beer in the basement with me.

Especially in the early years of my project I often sneaked down to the 9th floor where the Tribler people live. Although you have never been able to convince the world that it needs a social file sharing system, you are definitely a fun and social group yourselves. Thanks for adopting me in some of your great discussions (including the beer and painting sessions).

Thanks to Arjen I had the opportunity to spend part of my time at CWI in Amsterdam. I am grateful to everyone in INS1 and INS2 for helping me to see my research in a broader context. Although we never really engaged in a project together, I learned a lot from you. Theodora and Roberto, thanks for the occasional discussions on my papers and code.

When I started this project in 2006, the IR research in Delft was almost completely absent and I was very glad to find many great IR students in other places scattered over the Netherlands. Thanks to all you guys and girls in Amsterdam, Enschede, Nijmegen and Tilburg. I especially remember some fantastic trips to Glasgow and Singapore, it would have been boring without all of you.

In 2009 I spent 3 months in Barcelona which is undoubtedly one of the greatest cities to live. The Yahoo! Lab provided a very inspirational environment with great research and amazing people. Next to my co-authors I owe a special debt of gratitude to Adam for helping me with the introduction of this thesis. Adam, I would have loved to have more time to get to know you.

Papa en mama, zonder jullie was ik hier nooit gekomen. Jullie hebben niet alleen alles mogelijk, maar ook nog makkelijk gemaakt. Bedankt voor alle steun in mijn leven. Ik heb van julie nooit geleerd wat stress is, en dat bevalt prima. Karen, ik begin me steeds meer te realiseren hoe belangrijk het is geweest dat ik in mijn leven altijd een vriendje bij me had. Ook al lagen we niet altijd op een lijn je bent altijd belangrijk voor mij. Oma's, jullie hebben nooit begrepen waar ik mee bezig was en ook dit boek zal jullie hier niet verder mee helpen. Ik kan jullie in ieder geval verzekeren: het gaat goed met mij. De rest van de familie allemaal bedankt voor jullie geweldige gevoel voor humor. Vooral mijn bonus broer en zus, Christa en Bart, jullie zijn tof maar dat spreekt voor zich.

Van 2000 tot 2007 heb ik mogen genieten van het meest briljante studentenhuis van Delft. Alle jongens van de Boerderie bedankt voor de geweldige tijd. De vele BBQs, kerstdiners en verdiepingsweekenden waren allemaal legendarisch. Tijdens mijn studie heb ik veel mooie mensen leren kennen, een paar daarvan hebben een blijvende indruk achter gelaten. Chris en Jo, van jullie heb ik geleerd dat samenwonen op 14m^2 best mogelijk is en nog gezellig ook. Sinds die tijd zijn we alleen maar dichter naar elkaar gegroeid. Met jullie is het gewoon altijd goed. Ik hoop dat jullie nog lang een belangrijk deel van ons leven uit zullen maken. Lennert en Ildi, wat is het leven fijn en wat kunnen jullie er goed van genieten. Op dit gebied kunnen we zeker nog van jullie leren en dat hopen we voorlopig ook te doen.

Ook buiten Delft blijken leuke mensen te wonen (wel significant minder). Kristel, Thirza, Ada, Yvonne, Marleen, Machiel, Edwin, Etienne, Alex en Hugo ik hoop niet dat ik nog hoef te vertellen dat ik jullie geweldig vind. Ondanks dat we veel tijd met z'n allen doorbrengen zijn jullie allemaal individueel onmisbaar voor me. Er is altijd te weinig tijd om met jullie samen te zijn, maar het maakt niet uit hoe lang we elkaar niet gezien hebben, het voelt altijd vertrouwd.

Een vriendin selecteer je doorgaans niet op haar familie. Mocht ik dit echter wel gedaan hebben dan was ik ongetwijfeld bij dezelfde familie uitgekomen. Ad, Rietje, Gineke en de rest van de Klerken (+ vrienden), een leukere schoonfamilie had ik niet kunnen verzinnen. Ik heb me bij jullie vanaf het begin thuis gevoeld. Bedankt voor alle gezelligheid en het vertrouwen in ons.

Bij het schrijven van dit boek heeft niemand mij zo veel steun gegeven als m'n vriendinnetje. Willemieke, met jou is alles leuker. Je tolereert me als ik weer eens 's avonds laat achter mijn computer zit en je help me op de goede momenten afstand te nemen van mijn werk. Door jou heeft mijn leven een doel en dat bereiken we elke dag.

Maarten Clements
Delft, October 2010

Contents

1 Introduction

Personality goes a long way
Jules Winnfield, Pulp Fiction (1994)

1.1 Information

1.1.1 The Early Days

The ability to exchange information among individuals has proven to be one of the most advantageous milestones in the evolution of the human race. The communication skills acquired over the ages have helped us to discover nutritious food sources, alert each other to threats and to collaborate in general. One of the earliest steps in the improvement of communication efficiency has been the development of speech. The difficulty to determine the exact origin of speech [77] however illustrates the problem of this communication method. Information transferred by speech is solely stored in the minds of individuals. Therefore, access to valuable information was limited to someone's social circle and much of the knowledge that was acquired diluted over time due to the limited capacity of human memory. When people started to collaborate more intensively, a method had to be devised that would enable the conversion from a concept stored in human minds to a more reliable and expandable platform.

Around the 4th millennium B.C. in Mesopotamia, formerly individual farmer communities started to group together in larger settlements. With fast growing inhabitant numbers and flourishing markets, these civic areas became increasingly difficult to manage. It is commonly believed that in this time the script that previously existed of simple drawings developed into a more complex system that enabled the registration of trading transactions and other administrative information. Inscribed in clay tables this information could easily be stored, retrieved and transported [82].

Only with hindsight we completely understand the ramifications of these events. The ability to store information on tablets allowed for practically unlimited capacity and perhaps even more importantly, the decoupling of information from the human mind. Information could be transferred independently from the writer, and the knowledge could be reconstructed in a different time and location.

It was soon realised that writing provided a great stimulus to the economic growth of the society, and so the following ages are packed with inventions that facilitated more complex language structures and more practical writing material. Not only was writing used to store administrative information, but long told stories that were up to now only transferred by oral history could finally be stored to stand the ravages of time. The rapid increase of stored information led to the emergence of great libraries in large cities and cloisters. A massive storage of information however, does not directly breed knowledge. Only when information can be effectively disseminated to the community, will knowledge be able to emerge and bloom.

In the 3rd century B.C. one of these massive libraries was located in Alexandria, a fast growing Egyptian city founded by Alexander the Great. The poet, teacher and scientist Callimachus was appointed as librarian and noted that a system needed to be developed to order the numerous scrolls in the library. Around 245 B.C. he started a massive effort to organize the library by authors and subjects. The complete catalogue or *pinakes* (tables), finished after the death of Callimachus, allegedly spanned about 120 volumes, and was a great source for Greek literary research in the years to come [38].

Retrieval of information from a manually indexed library results either in *total recall*, the retrieval of all available information on the requested topic, or a ranking based on the opinion of the librarian. In 1755 Diderot noted that even with extensive cataloguing the increase in information would eventually break the system:

> *"As long as the centuries continue to unfold, the number of books will grow continually, and one can predict that a time will come when it will be almost as difficult to learn anything from books as from the direct study of the whole universe. It will be almost as convenient to search for some bit of truth concealed in nature as it will be to find it hidden away in an immense multitude of bound volumes. When that time comes, a project, until then neglected because the need for it was not felt, will have to be undertaken."* [32]

Diderot proposed to use *Encyclopedia* as a vast compendium which would bring together and systematize all knowledge worth reading. Indeed the encyclopedia would prove its value, but we will see that indexing the full corpus of written information would remain possible for far longer than expected.

1.1.2 Faithful Servants

At the end of the 19th century the Belgian Paul Otlet saw that radical rethinking of information management was needed if the collective world knowledge would continue to move forward at its current pace. His vision was that libraries needed to be transformed into stations in an information network reaching around the world. New documentary techniques should provide fast and effective consultation. In this way, the network of offices would form a *mechanical, collective brain*. Otlet even imagined that the work-desk of the future might consist only of a screen and a telephone to request documents. A *telereading* machine would allow text to be read at a distance from one of the offices. So that, *"in his armchair, anyone would be able to contemplate the whole of creation or particular parts of it"* [97].

In 1895, Otlet met Henri La Fontaine and together they started to build the first of these great information *databases* that would later be renamed to *Mundaneum* in Brussels. This would be the central storage place where all original documents were stored and which provided copies to the other offices. By the outbreak of the war in 1914, over 11 million entries were recorded in the database and the *Universal Decimal Classification* was invented as a highly effective organisation scheme. Otlet set up a fee-based service sometimes referred to as an *analogue search engine* to answer questions by mail, by sending the requesters copies of the relevant index cards for each query. By 1912, this service responded to over 1,500 queries a year. Unfortunately, all the effort invested in the system would be in vain. Management issues and disputes with the government led to the a closing of the Mundaneum in 1934. The destructive effect of Second World War and Otlet's death in 1944 meant that many of his ideas and effort were never passed on to future generations [97].

Around the Second World War, the American inventor Vannevar Bush who had previously worked on the design of one of the earliest large scale analogue computers, was director at the U.S. Office of Science Research and Development. Recognising the massive amount of data that was generated by administrative reports and scientific research, he published the article *As we may think* in which he proposed a hypothetical solution that shared many similarities to the vision of Otlet:

> *"Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and to coin one at random, "memex" will do. A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory."* [13]

The proposed implementation came in the appearance of a desk with several buttons and levers and a translucent screen on which material would be projected. The content was stored on microfilm (allowing the storage of millions of books in the desk) and the mechanism inside the desk presented the requested documents on the screen. Based on the associative behaviour of human thought, Bush imagined that the machine should allow the user to make associative *trails* between the documents. These trails would enable instant retrieval of a related document to the one currently read by the user. It is unclear whether Bush was aware of the similarity of his ideas to the vision that was put forward by Otlet, but both ideas clearly contained many of the characteristics currently found in a personal computer and a cross referencing scheme that would later be one of the fundamentals of the *World Wide Web*.

In 1949, the Italian Jesuit priest Father Roberto Busa started a less futuristic but immense task of creating an *index verborum* of all the words in the works of St. Thomas Aquinas and related authors. In total these works contained about 11 million words in medieval Latin and instead of indexing all words as they appeared in the text, Father Busa had decided to produce a *lemmatised* version, where all inflections of the words are grouped into a single term. The invention of a machine commonly referred to as *computer* came to the attention of Father Busa, who turned to

Thomas J. Watson at IBM in search of support. Using punched cards to read the texts, the lemmatisation of the entire corpus was completed in a semiautomatic way while human employees dealt with word forms that the program could not handle [56].

In the same year, Sanford V. Larkey proposed to study the extend to which machines could be used to index and retrieve the scientific publications stored at the Welch Medical Library:

> *"The use of machine methods may appear somewhat Utopian but one must look to the possibilities of the future. . . . Machines can probably be designed to do what we desire but it must be determined how well they do it and if it is worth doing."* [73]

The final report of the project lists many possible applications of machines in information indexing and retrieval. Also, a suggestion is made for a central information center for all science, that would have complete coverage of all important scientific literature. It is however acknowledged that with the machines available back then the best result would be retrieval of a bibliographical reference instead of the original documents [55]. But even before computers could actually store large document corpora, significant steps would be made on the performance of retrieval systems.

Many of the foundations of modern information retrieval can be found in the SMART (Salton's Magic Automatic Retriever of Text) information retrieval system, developed at Cornell University in the 1960s by Gerard Salton and his colleagues. Although reading documents was still cumbersome and involved passing the documents on tape through the computer for each search[1], much of the theory that was later detailed in Salton's book *A Theory of Indexing* was proposed to generate a statistical relevance ranking of the available documents [109]. No longer would the opinion of a librarian determine the order of the retrieved information, but a statistical ranking objectively derived from the data could be presented to the user. In the following years, computers were enriched with hard drives, storage capacities continued to grow, and many libraries were equipped with retrieval systems.

Although collections could now easily be searched, the information was still only accessible to someone physically interacting with the computer. For collaboration, information needed to be transported to other computers on disks. The first ideas about a network to exchange content between computers are commonly credited to J.C.R. Licklider, from 1962 the head of the Information Processing Techniques Office (IPTO) at ARPA, the United States Department of Defense Advanced Research Projects Agency.

Licklider envisioned that everyday work could be made much more efficient by stimulating the *Man Computer Symbiosis* and collaboration could be improved by connecting the computers of different researchers in a network. Several years after Licklider left ARPA his ideas would result in the creation of ARPANET, a packet switching network that would later evolve into the Internet as we currently know it [75].

In 1980, Tim Berners-Lee at CERN (the European Organization for Nuclear Research) proposed a project based on hypertext which enabled access to documents

[1] http://blog.tomevslin.com/2006/01/search_down_mem.html, June 2010

stored at a remote location. By combining this work with the Internet in 1989 the *World Wide Web* was created, a framework that aimed to enable a universal linked information system [10].

By allowing documents to refer to each other on the Internet, it became much easier to retrieve information from remote sites. Not only for users, but computers could also use the hyperlink structure to learn about the web. Using web-crawlers the content of the Internet could automatically be collected and indexed by search engines [115]. With the proposal of new ranking algorithms like HITS [70] and Pagerank [99], a global prior ranking of the relevance of all available content on the WWW could be made. Web search engines became more effective and would soon be the mainstream method for searching information.

In 1999, DiNucci coined the term *Web2.0* for the transformation of the known Internet to a more dynamic variant. The web should be the *"ether through which interactivity happens"* [33]. This change of the Internet would appear not as a single invention, but several gradual changes to the implementation and perception of websites.

One of the prominent changes was that web designers started to *crosslink* almost everything on the page; next to links between different documents, it became possible to navigate from content to users and even between users. Websites were transformed in *social media* where people could share information and easily access and discuss each other's contributions.

For the retrieval of this content, traditional indexing methods did not suffice. Soon the volumes of provided content became too numerous even for a large team of librarians to annotate and categorise, and for contributions containing multimedia content, automated text based methods could not come to a rescue this time. The solution to disclose the growing media repositories would be found in the collaboration of the community. Interfaces were developed that allowed everyone to assist in the indexing or annotation of the provided content. This *collaborative index* ranged from textual tags to preference indications like ratings or buttons to express interest or disinterest in certain information.

By annotating content and indicating their preference, the users of social media started to leave many traces that could be used by the system to learn about their taste. Based on the user's annotation history, the system could be designed to provide *personalised information rankings*. Relevance was no longer a global notion, but dependent on the personal preference of each individual.

1.2 Scope

1.2.1 Social Media

The term *social media* is used to refer to websites that apply Web2.0 techniques to create a platform where users can observe each other's behaviour and social ties, and which provides efficient tools to communicate and collaborate.

In most social media, people can contribute *user generated content*, which can range from textual content (e.g. weblogs, encyclopedia articles) to different forms of multimedia content (e.g. video clips, photos, music). All this content is collaboratively indexed by the community and qualitative feedback can be given to either the contributer or the content itself. The system weaves all these contributions into an easily accessible web by providing cross links wherever possible.

Social media have emerged from different perspectives. Primarily two main classes can be identified: first, systems that start from the notion of a social network and stimulate the interaction between the users by allowing them to share content; examples include: Orkut, Friendster, MySpace, LinkedIn and Facebook. Second, networks that were primarily created for the distribution or management of content, and use the social network as an overlay that stimulates this content distribution; for example: YouTube, Flickr, LibraryThing, Del.icio.us, CiteULike. Both types of systems have emerged simultaneously over recent years, and both developments showed that only when both the social and content features are effectively implemented, have systems been able to satisfy a large user community for a long period.

The popularity of combined content sharing and social features can be explained by various arguments. As social media can be used to share activities or status updates, people can be easily updated on the current well-being of their friends. Hereby, the overall community interaction is stimulated, and it has become easier to maintain solid social relationships with many people. Clever matching algorithms make sure everyone is informed about the most relevant information and even stimulate the exploration of new content or social groups. Every single click generates a page full of information about people, objects or events. In this way, social media provide answers to and simultaneously stimulate the curiosity inherently present in the human race. Many users have even indicated that this instant gratification may lead to Internet addiction [122].

Social media are used to maintain both strong and weak social ties [41]. Naturally, people have just a handful of intimate friends (strong ties) and several hundreds of acquaintances (weak ties). While most people just focus on their closest friends, some users try to connect to as many people as possible and manage to obtain thousands of social ties. The resulting network is known to have a *small-world* structure, which is characterised by high clustering and short path length between any two selected nodes in the network [137]. This means that anyone in the world can easily be contacted through the friends of your friends [131; 6]. Not only has it become easier to maintain relationships within your current social circle, but social media actively stimulate the discovery of new relations. This feature can be very useful to get introduced to people or companies when searching for a new job. Easy distribution and access to each other's self generated content has even made some of the social media popular platforms to display artistic skills. The open community stimulates the recognition of these skills and social media have become a new doorway to instant fame.

The structure of social media does not only benefit its users, as the company that owns the website also gets its share. Common business models in current web services rely either on sales or advertisements. To ensure a web service's revenue, it needs to keep the users interested for as long as possible or provide them with (easy tools to find) the products that match their interest. If the user appreciates a service provided by the network, he might even pay for an account with more possibilities. Satisfying the users' expectations is a challenging task in the continuously evolving information age, that can only be accomplished if the system is tailored to suit each individual user's preferences. As the system aggregates all the user's interactions it learns more and more about the user's preference over time. This collection of user specific information allows the company to provide personal advertisements. If the right products are recommended to each user at the right time, an advertisement may not even be seen as an annoying distraction, but will be appreciated by the user [79].

1.2.2 Collaborative Indexing

With the introduction of collaborative annotations in social media, content indexing has shifted from objective statistical methods to a more subjective categorization. Everybody contributes to the description and organisation of the data. Actively or passively, everyone leaves traces that can be used to improve the index of the content. Some of these traces are subjective and therefore inherently related to the individual user, others objectively say something about the content.

Figure 1.1 shows the model of collaborative annotations that is used in this thesis. Because of subtle differences in social media design and user behaviour, no general model will capture all possible aspects that might be present in this data. The proposed model therefore is one of the many possible ways to represent this data, and is proposed as a guideline for the definitions used in this thesis rather than a ground truth. The entities that make up the data model are the *user* who interacts with the system, the *item* which might be any piece of information (e.g. a textual document, a movie or an image) and a *tag* which is a description attached to an item by a user. The tasks presented in this thesis will focus on the relevance prediction of one of these entities.

Opinions about the quality of content can be expressed through *ratings*. Different interface elements let users express a rating in various ways, ranging from 10 point scales to binary judgements in the form of *digg* or *like* buttons. A rating can also be derived from a mouse click or an actual purchase of the item. Ratings create a relation between user and item where the value of the rating determines the strength of the relation.

Tagging is used in social media to give users a way to annotate content with various types of descriptions. A tag assignment (*TAS*) creates a ternary relation between user, item and tag. Commonly, tags are keywords that the user considers representative of the topic of the items. These keywords can either consist of free text or be selected from a limited vocabulary in the system.

Metadata is the information that describes an item, like *length*, *author*, *video format* etc. The relation between tags and metadata is a much discussed topic. In this thesis a tag is seen as a piece of metadata that is promoted by a specific user, because he considers it relevant to describe his personal relation to the item. Thereby the user creates a direct preference relation to the tag. For example, the author of a book is commonly seen as metadata, but many users also add the name of the author as a tag to indicate that they are interested in this author. Some of the metadata, like the number of pages of a book, will hardly ever be promoted to a tag because it is

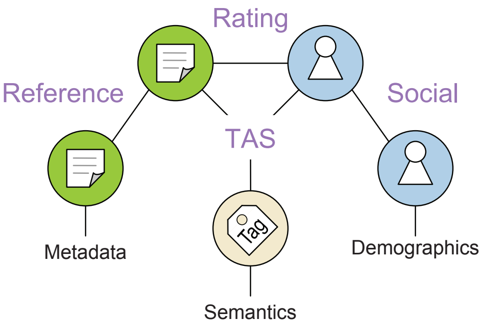


Figure 1.1: An overview of the relations between users, items and tags created by common annotation methods in social media. Next to collaborative annotations, many entities contain some static information indicated by *Metadata*, *Semantics* and *Demographics*.

unrelated to user preference. A tag can thus also refer to other types of descriptions besides keywords. An increasingly popular annotation is the geographical location of the item. Mostly photographers use these *geotags* to indicate that not only was the photo created at that location, but indirectly show that they have visited that place as well.

Although the semantics of a tag are dependent on the context in which it appears it is not modeled here as a relation between different entities. Tag semantics are derived from the user's language which has evolved over time and is not defined by the user in a social tagging system. All visualised relations in this model are based on annotations created and edited by the users of the system.

People are linked by the social network created from friendship relations or group memberships. The emergence and characteristics of this social structure have often been studied, also independently from social media [136]. The users who interact with the system often provide some *demographic* information about themselves, like sex, age or home town. This information only describes the user and does not link different entities in the system.

Finally, items in social media can be linked by *references* like hyperlinks or citations, or grouped in topical clusters. If the content is generated by the users themselves, these relations are also collaboratively created. Therefore these references are also visualised as collaborative annotation.

1.2.3 Thesis Demarcation

The social media studied in this thesis primarily focus on content distribution and management. The used data sources are derived from:

Movielens[2] A movie recommender website created by GroupLens Research at the University of Minnesota [105]. The data consists of 100,000 ratings on a scale from 1 to 5 from 943 users on 1682 movies [54], and is publically available from the Grouplens website.

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

LibraryThing[3] An online book catalog that allows its users to add tags, ratings and reviews to the books in their personal library. It is also possible to join topical groups and become friends with other users. The data collected for this thesis consists of 7.5 million annotations that contain one or more tags and a rating. Further details about the collection and statistics of this data set will be given in Section 4.3.2.

Bibsonomy[4] A website for categorisation and sharing of literature references and website bookmarks. A public data set from Bibsonomy was provided for the RSDC08 Discovery Challenge[5]. After preprocessing, this data contains about 214 thousand annotations with one or more tags. Further details about this data set will be given in Section 4.3.2.

Flickr[6] A photo management and sharing website where users can upload their photos and annotate them with tags and geotags. For Part III of this thesis a set of 43 million geotags from 126 thousand users is collected. Further details about this data set are given in Section 8.3. For Chapter 9 a large large collection of textual tags from Flickr was made available by Yahoo!.

The annotation methods that will be studied in this thesis are: *ratings*, tag assignments of textual *tags* and *geotags* and tag *semantics*. The informational value of the relations created by the annotations will be studied without making use of the content of the individual items. Therefore, many of the presented results will extend to other platforms that employ similar annotation methods for content description.

1.3 Contributions and Outline

The main contribution of this thesis is that it improves the understanding of collaborative annotation data by studying new and existing retrieval tasks and thereby reveals new opportunities for personalised information access. Based on recent developments in social media, retrieval tasks are defined that will improve the information accessibility in a *user centered way*. For each task, an appropriate relevance prediction method is either adopted from state-of-the-art work and adapted to the task, or an algorithm is proposed if a previously unaddressed task is studied. The parameters of the method are selected so that they relate to external factors that might influence the task. In this way, the optimisation of the parameter settings reveals insight in the data, which can be related to system design issues or user behaviour.

The deployed ranking models are used as a means to learn about the factors that contribute to the accessibility of the information in the system. By increasing the understanding of the collaborative annotation data, new possibilities for personalised

[3] http://www.librarything.com/

[4] http://www.bibsonomy.org/

[5] http://www.kde.cs.uni-kassel.de/ws/rsdc08/

[6] http://www.flickr.com/

retrieval are uncovered. It appears that the potential retrieval gain attained by personalisation is strongly dependent on the situation. Slight variations in task definition, system design or data characteristics determine the optimal approach and the extent to which the result can be adapted to the individual user. By iteratively changing the task definition and finding the optimal model parameters this thesis simultaneously finds the optimal personalised retrieval methods and increases the understanding of the data.

In previous literature, the study of personalised information retrieval was mostly focused on web search, the field of recommender systems dealt with making predictions based on ratings or sales data and social media studies were oriented towards the understanding of the collaborative annotation effort. Although all three fields aim at understanding and improving digital information exchange, the communities are largely separated. With this thesis the studies of personalised information retrieval, recommender systems and social media analysis have been brought closer to each other.

This thesis is divided in 4 parts based on the relations defined in Figure 1.1. Part I deals with ratings and provides several extensions to the field of collaborative filtering. Part II shows that many personalisation tasks emerge if textual tags can be assigned to the content. This part gives a strong contribution to the understanding of social tagging systems. In Part III new location-based tasks are proposed based on geotag data. In three chapters, this part gives an interesting insight into geotag data and methods to find relations in this data. Part IV shows how the collaborative annotation effort can be used to learn about the semantics of tags. All chapters directly relate to published scientific papers, each chapter is therefore self contained and can be read independently from the rest. The discussion in Chapter 11 gives a synopsis of the main findings, a discussion of the followed approach, some open issues regarding this work and a prospect of the future of information consumption.

PART I - Ratings

User-based collaborative filtering exploits a set of similar users to predict ratings for the target user. **Chapter 2** of this thesis extends the user-based collaborative filtering approach to perform in a distributed peer-to-peer environment. The social peer-to-peer client Tribler builds a semantic overlay based on users with similar preference profiles to enable accurate search and recommendations [102; 134]. The limitation introduced by this setting is that each user has only partial access to the ratings of other users, and needs a way to efficiently select the most useful peers to store in its local cache. The main contradiction in user-based collaborative filtering is that if two users are found with exact similar ratings, both of them will not be able to recommend anything to the other. To avoid filling the cache with users that are similar but not able to provide recommendations, chapter 2 introduces a model based on similarity, confidence and usefulness which is especially applicable in distributed recommendation settings.

Compared to rating prediction, the task of content ranking is only focused on correct prediction of the most interesting content (the top of the ranking), instead of predicting a relevance value for all available content. A different task also asks for different prediction methods. **Chapter 3** adopts a previously proposed modification [29] of personalised Pagerank [99] to address this task on a rated corpus from LibraryThing and a data set from Movielens. Graded relevance assessments in the form of ratings however can either indicate a positive or negative relation between the user and content. With a combination of two separate graphs, the notion of negative feedback is included in the personalised ranking. The results indicate that a user's positive and negative ratings are strongly interleaved, which can be explained by the selective process before a user decides to assess a certain item. Low ratings should therefore not be punished but exploited to find more relevant content.

Chapter 2 is published as: Maarten Clements, Arjen P. de Vries, Johan A. Pouwelse, Jun Wang, and Marcel J. T. Reinders. Evaluation of Neighbourhood Selection Methods in Decentralized Recommendation Systems. In *SIGIR07 Workshop on Large Scale Distributed Systems for Information Retrieval*, July 2007.

Chapter 3 is published as: Maarten Clements, Arjen P. de Vries, and Marcel J. T. Reinders. Exploiting Positive and Negative Graded Relevance Assessments for Content Recommendation. In K. Avrachenkov, D. Donato, and N. Litvak, editors, *WAW09: Proceedings of the 6th International Workshop on Algorithms and Models for the Web-Graph, LNCS 5427*, pages 155-166. Springer-Verlag, Berlin, Heidelberg, February 2009.

PART II - Tags

In **Chapter 4** a limited random walk is applied to study the influence of the design choices in collaborative annotation systems on popular retrieval tasks. A framework of 12 tasks is proposed and 4 different tasks are evaluated. For the tasks *Item recommendation*, *Personalized search*, *Tag suggestion* and *User recommendation* the value of a user's previous tags and ratings is evaluated and the number of steps made by the random walk model is used to study the optimal amount of smoothing with the background popularity. It is shown that when a site actively suggests tags to its users a more coherent corpus is created and a user's personal tags can be used for better personalisation.

For many tasks the combination of ratings and tags can improve the retrieval of relevant information. Also, it is shown that the combination of the annotations by multiple users improves the retrievability of the content. When a system only allows a single user to contribute tags to an item the content is less well described and more background smoothing is necessary. Both findings indicate that the aggregation of different representations of the same information (or polyrepresentation) results in an improved description of the intrinsic information need. This corresponds to the earlier observations that polyrepresentation can often enhance retrieval performance [60], and recommender systems [11; 133].

When a user queries a collaborative tagging site, the amount of information in the query determines the quality of the retrieved results. When more terms are appended to the query, the intent of the user becomes more obvious and the potential for personalisation decreases. **Chapter 5** studies the limits of personalisation in tagging systems with increasing query length. In the used tagging corpus, a query consisting of 4 or more tags is shown to be unambiguous and therefore personalisation and smoothing with a background model have no positive effect on the retrieved content.

Chapter 4 is published as: Maarten Clements, Arjen P. De Vries, and Marcel J. T. Reinders. The Task Dependent Effect of Tags and Ratings on Social Media Access. *ACM Transactions on Information Systems*, 28(4), October 2010.

Chapter 5 is published as: Maarten Clements, Arjen P. de Vries, and Marcel J. T. Reinders. The Influence of Personalization on Tag Query Length in Social Media Search. *Information Processing & Management*, 46:403-412, May 2010.

PART III - Geotags

The task of *location prediction* is relatively new. Only since GPS receivers have become integrated in mobile devices like cameras and mobile phones, has enough data been collected to study the travel history of individual users. In **chapter 6-8** a set of geotags collected from the popular photo sharing site Flickr is used to study whether location data can be used to predict which landmarks attract similar people. Compared to traditional recommendation problems, location prediction has to deal with continuously valued points in a 3D space, making similarity computation less trivial than in the usual discrete space constructed from a limited number of objects.

A grid based approach with Gaussian smoothing is used in **chapter 6** to find that similar locations can be identified by comparing the overlap in user visits to the prior visiting probability of a location. This method can be used by travel websites to show similar destinations for a given query location. In **chapter 7-8** both user-based and item-based approaches to location recommendation are proposed, using a Gaussian density estimation to compute distances in the continuous object space. The location similarity model proposed in **chapter 8** proves to be a versatile model that can accurately predict interesting locations at different scales. Using only the location history of the Flickr users, it is possible to relate semantically similar places at opposite sides of the world. Depending on the amount and coherence of the preference information contributed by the user, accurate personalised location recommendations can be made at any preferred scale in a previously unvisited region.

Chapter 6 is an extended version of: Maarten Clements, Pavel Serdyukov, Arjen P. de Vries, and Marcel J. T. Reinders. Finding Wormholes with Flickr Geotags. In Cathal Gurrin et al., editors, *ECIR 2010, LNCS 5993*, pages 658-661. Springer-Verlag, Berlin, Heidelberg, 2010.

Chapter 7 is published as: Maarten Clements, Pavel Serdyukov, Arjen P. de Vries, and Marcel J. T. Reinders. Using Flickr Geotags to Predict User Travel Behaviour. In *SIGIR10: Proceedings of the 33th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 851-852, New York, NY, USA, 2010. ACM.

Chapter 8 has been submitted as: Maarten Clements, Pavel Serdyukov, Arjen P. de Vries, and Marcel J. T. Reinders. Personalised Travel Recommendation based on Location Co-occurrence *ACM Transactions on Information Systems*. 2010.

PART IV - Tag Semantics

Since the introduction of tag functionality in social media *semantic tag analysis* methods have been proposed to understand the user dynamics in the system [49]. Although end users are not educated in the annotation of content, it is commonly believed that the aggregation of many tags will lead to an accurate description of the content [43]. The descriptions provided by the collaborative effort of the users do not only describe the content, but also contain information about the structure of the underlying language. It is interesting to investigate how semantic relations between tags can be learned from the statistics of the data.

Chapter 9 presents a new framework to establish the semantic specificity relation between two tags. Within this framework an extensive study of existing and new term specificity features shows that the combination of several features can be used to improve search and browsing tasks. Most previously proposed methods to determine tag specificity observe each tag in isolation and estimate their relation by comparing both individual measurements. In this chapter a method that takes both tags into account while computing their relation is shown to outperform the prediction based on other features.

Similar tags are often found by looking at the co-occurrence of two tags on the same content [8; 119]. **Chapter 10** shows that synonyms can be distinguished from otherwise similar tags by exploiting the user dimension as synonyms are rarely used by the same user. Automatic estimation of tag semantics is a useful addition to existing term hierarchies as an automatic method can deal with new emerging terminology and misspelled words, while a fixed word hierarchy can not. The collective intelligence created by the community effort can be exploited in many ways to discover new trends in user behaviour and personalisation opportunities for more effective information retrieval.

Chapter 9 is based on collaborative work with Yahoo! Research Barcelona: Maarten Clements, Börkur Sigurbjörnsson, Vanessa Murdock and Roelof van Zwol. Deriving Term Specificity from Social Tagging Data. *Yahoo! Research Barcelona*. 2009.

Chapter 10 is an extended version of: Maarten Clements, Arjen P. de Vries, and Marcel J. T. Reinders. Detecting Synonyms in Social Tagging Systems to Improve Content Retrieval. In *SIGIR 08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 739-740, New York, NY, USA, 2008. ACM.

Part I

Ratings

2

Evaluation of Neighbourhood Selection Methods in Decentralized Recommendation Systems

Recommendation systems are important in social networks that allow the injection of user-generated content and let users indicate their preferences towards the content introduced by others. Considering the increase of usage of these collaborative systems, it seems only a matter of time before the current centralized systems will be replaced by decentralized solutions. However, current collaborative filtering systems assume that recommendations can be based on the entire data collection in the network.

This work evaluates the performance of user-based collaborative filtering systems when only partial knowledge about the network is available at an end-user's computer. We propose a utility model that combines three important aspects of network users (similarity, confidence and usefulness) in order to create a semantic overlay network optimized for autonomous content recommendations. We compare different similarity functions on the most common dataset in collaborative filtering and we show the influence of the confidence and usefulness parameters on both dense and sparse data.

We find that the commonly used similarity function results in sub-optimal performance when used as updating criterion for locally stored rating profiles. We show that taking into account the level of confidence in the computed similarity can greatly improve recommendation accuracy, especially when a small user neighborhood is selected. Also, conventional methods select many users that cannot contribute to the recommendation, because they have rated too few items. The usefulness parameter that we introduce compensates for this problem, so that even a small local cache in very sparse data provides valuable recommendations.

This chapter is published as: Maarten Clements, Arjen P. de Vries, Johan A. Pouwelse, Jun Wang, and Marcel J. T. Reinders. Evaluation of Neighbourhood Selection Methods in Decentralized Recommendation Systems. In *SIGIR07 Workshop on Large Scale Distributed Systems for Information Retrieval*, July 2007.

2.1 Introduction

Over the last decade we have witnessed the rise of a new phenomenon on the Internet. Social networks that allow users to share their own content and browse through other peoples items are rapidly gaining popularity over conventional static websites. Social communities as YouTube[1], Flickr[2] and many more attract millions of users each day, and this growth in online collaboration is unlikely to stagnate in the coming years.

Meanwhile the largest part of Internet traffic is occupied by decentralized content distribution systems depending on peer-to-peer (P2P) technology. P2P protocols like Gnutella[3] and BitTorrent[4] have proven to be among the most competent methods to efficiently distribute large volumes of data to a large community of users. In order to cope with the increasing amount of data stored on local servers from currently popular web services, we believe that decentralized social networks will undeniably play an important role in the evolution of global content distribution.

One of the emerging technologies that aims to discover information in online databases is collaborative filtering (CF) [105]. In CF the history of network users is dedicated to generate recommendations for a certain target user. Recommendations have shown to be a useful addition to conventional search, because they allow users to discover content that matches their interest without having to type a specific query. In the last decade many new techniques have been developed to facilitate this recommendation. The most common approach to CF is often termed 'memory based' recommendation, which can be split in two different approaches: 1) User-based recommendation first identifies a set of common users by comparing rating or download profiles, and subsequently uses only the information from these users to predict a recommendation for the target user. 2) Item-based recommendation attempts to find items that have been downloaded by the same users as the items for which the target user showed interest.

Research in CF has focussed on recommendations on central databases, where the entire dataset is available at any time. Our work considers a P2P architecture instead, in which users have partial *local* information about the network, optimized for their personal recommendations. This setting is motivated by the work by Pouwelse et al. [102], who developed TRIBLER, a P2P system capable of social content discovery. TRIBLER uses an algorithm based on an epidemic protocol that can be summarized as follows. Each user maintains a set of references to the peers he has discovered and stores the rating history of the N most similar peers in his local cache. In our setting, users have rated a selection of movies on a discrete scale $r \in \{1,2,3,4,5\}$ ($r = \emptyset$ if no rating was given). A user iteratively connects either a random peer (*exploration*) or one of the most similar peers (*exploitation*). Connected peers exchange information about their locally stored rating profiles, and update their local similar peers cache, based on the combination of their individual caches (see Figure 2.1). This way, the locally stored neighborhood converges to the N most similar peers available in the

[1] http://www.youtube.com

[2] http://www.flickr.com

[3] http://www.gnutella.com we believe that decentralized social networks will undeniably play an important role in the evolution of global content distribution.

[4] http://www.bittorrent.com

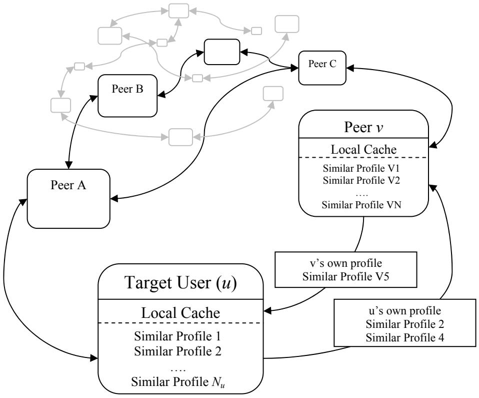


Figure 2.1: Each user maintains a local cache containing the rating profiles of N users. By connecting alternately the most similar or random random peers to exchange preference information, the list of most similar peers converges to the optimal peers from the network.

network. TRIBLER uses this social overlay network to increase both download speed and social features like content recommendations. This makes the selection criterion of similar peers a vital component of the system.

We perform a thorough survey of the most common similarity functions in CF in order to find the most suitable updating criterion for peer profile exchange. We then define a *utility* model that combines three aspects of network users with respect to recommendation (*similarity*, *confidence* and *usefulness*) and demonstrate how we can set the model parameters in order to converge to the ideal neighborhood for local storage in a decentralized network. In this work, we do not take important network aspects like *peer uptime* and *trust* into consideration for the selection criterion; we purely focus on the ideal setting for content recommendation. Because we do not experiment on an actual P2P system, our experimental results are based on the simplifying assumption that the target user's local cache always contains the correct N users identified by the utility model. Due to the rapidly changing nature of a true P2P system, the optimal set of users will in reality probably not be found. It has however been shown that in a small simulated P2P network full convergence can be reached [102]. Results in this chapter demonstrate that user selection in decentralized recommendation systems should not be based on similarity alone, but should take into account the confidence and usefulness aspects.

2.2 Related Work

In the most simple setup of a recommendation system the data consists of users and items with a simple binary (like/dislike) or continuous (rating) relation between them. User-based CF methods select the best N users that have given a rating for a certain item [12; 52; 105], and subsequently use only these rating profiles to estimate the rating of the target user for this item. This strategy however implies that we have to select a different set of users for each rating that we try to predict, which effectively requires access to the full user-item matrix. Since the creation of a semantic overlay by passing messages between peers slowly converges to the optimal configuration, this work assumes that we cannot update the set of similar users for each prediction. We select one set of N users to generate all predictions for the target user, resulting in a set of users that have not necessarily rated all the items we want to predict. Therefore, in our work it becomes important that we select users who have a large amount of given ratings, in order to be able to make predictions even for small N.

It has been shown that the creation of Semantic Overlay Networks (SON) can improve search performance in P2P networks. Semantic overlays can be created by clustering peers with similar content [30] or using a self organizing network based on a social model [16]. Usually these semantic overlays are built to optimize distributed search, while our utility model strives to predict autonomous recommendations.

Besides content search, SONs can improve the content distribution over the network. Pouwelse et al. [102] used this phenomenon to increase download performance and at the same time used the most similar peers to provide content recommendations for the network user. Up till now the optimization of the similarity function with respect to recommendations has been neglected in this system.

Ogston et al. discussed the value of recommendation in a decentralized network by comparing random user neighborhood selection to conventional CF selection methods [95]. They however did not adapt the conventional user selection method to achieve optimal performance in the decentralized situation. Item-based recommendation in decentralized networks has been studied by Wang et al. [134]. Because a P2P system is based on the connections between network users we however see a user-based recommendation as a more logical choice.

In this work we adopt a simple user-based recommendation scheme, and we extend the commonly used *similarity* function by integrating a *confidence* and *usefulness* factor. The confidence factor that we integrate in our model has been shown to improve recommendations in the work of Herlocker et al. [54], and was adopted by McLaughlin et al. [84] and Melville et al. [85]. However, we show in this work that they combined this factor with a similarity measure that demonstrates sub-optimal performance. Furthermore, we show that optimizing the weight of this confidence factor can increase the performance depending on the data and the used similarity function. After this, we show that if a small local cache is used for the recommendation of all items, our usefulness factor can improve the prediction, especially in sparse data sets.

2.3 Utility Model

2.3.1 Computation of Weighted Recommendations

User-based recommendation predicts a user's ratings on the basis of ratings made previously by the N 'most similar' users. In the original GroupLens system [105] as well as many recent works [85; 84; 135] the ratings of these users are combined by:

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v \in N_u | r_{v,i} \neq \emptyset} w(u,v)(r_{v,i} - \bar{r}_v)}{\sum_{v \in N_u | r_{v,i} \neq \emptyset} |w(u,v)|} \tag{2.1}$$

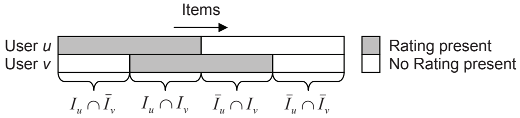


Figure 2.2: Two users with partially overlapping rating profiles. User u is the target user for which we want to recommend items that he has not rated yet. User v is a potential recommendation candidate who has a partial rating overlap with user u.

In which $\hat{r}_{u,i}$ is the estimated rating of the target user u for item i, $r_{v,i}$ is the rating of user v for this item, $\bar{r}$ is the mean of all the ratings given by a user and N_u is the selected set of nearest neighbors for user u, based on the similarity function $w(u,v)$.

It has however been shown by Herlocker et al. [52] that rating normalization by both the mean and standard deviation can slightly improve recommendations with respect to those based on equation 1. Therefore, we compute recommendations by:

$$\hat{r}_{u,i} = \bar{r}_u + \sigma_u \frac{\sum_{v \in N_u | r_{v,i} \neq \emptyset} w(u,v)(r_{v,i} - \bar{r}_v)/\sigma_v}{\sum_{v \in N_u | r_{v,i} \neq \emptyset} |w(u,v)|}, \tag{2.2}$$

which assumes that users' ratings not only differ by a certain offset, but also have a difference in standard deviation (σ).

Usually, not all users in the selected set (N) have given a rating for the items that we want to predict. Traditional CF methods simply select more users so that the predictions are always based on N ratings. Because we can not adapt the top-N for each prediction, we only sum over the existing ratings in the equations above. If the selected set of rating profiles from the N users is empty for a certain item, no collaborative recommendation can be given. In this case we predict $\hat{r}_{u,i}$ solely on the mean of all the target user's ratings ($\bar{r}_u$).

2.3.2 Utility Model for User Selection

The ranking of the most similar users to the target user is generally derived by computing the similarity $w(u,v)$ of the ratings that were given by both the target user (u) and the potential recommending user (v). This similarity is usually computed as the Pearson Correlation [105] or the Vector Similarity (VS) [12].

Figure 2.2 schematically represents two users' rating profiles, that have a partial overlap. The set of items rated by user u is represented by I_u. Based on this figure we define the value of a potential recommender as a combination of three aspects:

1. Similarity

   The most important aspect of the utility model is the *similarity* between the rating profile of the target user and the profile of the potentially selected peer ($w(u,v)$). Only the recommendations of similar users can give a good prediction of the rating from the target user. We state that the rating similarity between users can only be determined on the items that were rated by both users ($I_u \cap I_v$). Different similarity measures will be compared in section 2.3.3.

2. Confidence

   The *confidence* that we have in the similarity between users is related to the number of items both users have rated. If two users have rated many similar items the probability that the computed similarity is true becomes larger. Because the maximum rating overlap is bounded by the number of ratings that the target user has given, we express the confidence measure as $P(I_v|I_u)$; the probability that user v rated an item, given that user u has rated this item. This probability captures the intuition that user v should have rated many of the items that user u has rated, computed as:

$$P(I_v|I_u) = \frac{|I_u \cap I_v|}{|I_u|} \tag{2.3}$$

3. Usefulness

   We select the top-N similar users because they are expected to be good candidates for recommendation. However, this assumption holds *only* when these N users have actually given ratings for items that user u has not seen yet. In practice, data is sparse and the selected top-N often contains fewer than N ratings per item. For small N, this implies that the recommendation depends on very few ratings. In order to improve the reliability of recommendations for small N, we therefore introduce the *usefulness* of a peer as the probability that user v has rated an item that we want to predict for user u: $P(I_v|\bar{I}_u)$. This value can be computed as follows:

$$P(I_v|\bar{I}_u) = \frac{|\bar{I}_u \cap I_v|}{|\bar{I}_u|} \tag{2.4}$$

   Because we do not have local access to information about the entire collection of available items, the value of $|\bar{I}_u|$ is defined as $|I| - |I_u|$; the number of items that is rated by any of the users in the target user's local cache,

$$I = \bigcup_{v \in N_u} I_v \tag{2.5}$$

   minus the number of items rated by the target user itself ($|I_u|$).

We define the *utility model* as a weighted combination of these three aspects:

$$w^*(u,v) = w(u,v)^\alpha * P(I_v|I_u)^\beta * P(I_v|\neg I_u)^\gamma \tag{2.6}$$

where the factors α, β, γ are used to control the influence of the three aspects individually. We use a multiplication of the elements, because in the usage of the weight factor in Eq. 2.2, relative differences between user weights are more important compared to absolute differences. In this way, if any of the elements of the utility model is doubled, also the influence of this user in the recommendation is doubled. To adjust the relative differences between the different elements, we use the weight factors as exponents. We will use the value of $w^*(u,v)$ as the new weight factor in Eq. 2.2.

2.3.3 Similarity Measures

For the computation of the user similarity in our utility model we will consider two different similarity measures:

1. Full Pearson correlation

   We can compute the Pearson correlation between entire rating profiles by first *imputing* (ascribing) each users average rating at the positions where no rating was given,

   $$r_{v,i} = \begin{cases} \bar{r}_v & \text{if } r_{v,i} \in \emptyset \\ r_{v,i} & \text{otherwise} \end{cases}$$

   and then computing:

   $$w(u,v) = \begin{cases} \rho_{u,v} = \frac{\sum (r_{u,i}-\bar{r}_u)(r_{v,i}-\bar{r}_v)}{\sigma_u \sigma_v} & \text{if } \rho_{u,v} > 0 \\ 0 & \text{otherwise} \end{cases}$$

   Here we do not take the negative correlations into account, so all negative outcomes are set to 0. In Breese et al. [12] the imputation method is termed *default voting*.

2. Overlap Pearson correlation

   A similarity measure used in most memory based CF approaches [54; 52; 84; 105] is the Pearson correlation between only the overlapping part of the rating profiles. This method ignores the ratings that were given by only one of the two users under comparison, and can therefore give incorrect normalizations.

Vector similarity (cosine correlation) and the Spearman rank correlation coefficient were compared to the Pearson correlation coefficient in the work of Breese et al. [12] and Herlocker et al. [52] respectively, but both appeared to perform significantly worse than the Pearson correlation. Cosine correlation only differs from the Pearson correlation by the fact that it ignores the difference in rating offset between users. We believe this offset is a natural phenomenon that is inherently present in user provided ratings. We will therefore not consider this measure in this work. The Spearman rank correlation coefficient first ranks all ratings and subsequently computes a Pearson correlation of the values of the ranks. Tied ratings get the average of all the rank values that belong to the same rating. We follow the assumption of Herlocker et al. [52], that because we have only 5 distinct ratings, many ties will occur and this measure will not improve over the normal Pearson correlation.

2.4 Experiments and Results

2.4.1 Data

We have conducted our experiments on the Movielens[5] data set, which consists of 100,000 ratings for 1682 movies by 943 users. This results in a dataset with a density

[5] http://www.grouplens.org/

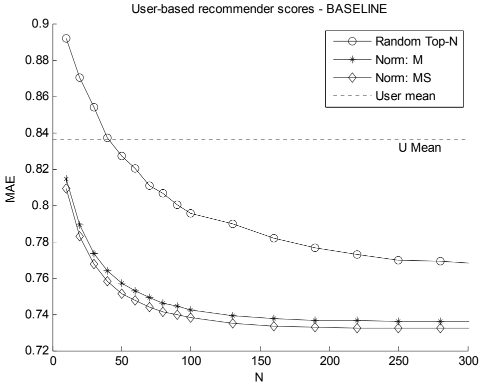


Figure 2.3: The baseline MAE scores. We show the MAE achieved with normalization by mean (Norm: M) or both mean and standard deviation (Norm: MS). Also, the MAE for random peer selection and the MAE when we estimate the ratings by the target user's mean rating are shown.

of 6.3% (about 1 out of 16 items has been rated). Ratings have been given on a scale of 1 to 5, 1 being dreadful and 5 excellent. The data has been split into 5 different training and test sets, where the training sets contain 80% of the data and the test sets 20%. The splits have been made in a way that all users and items maintain a part of their ratings in every set.

In our social P2P setting we see this data as user injected content that has been rated by multiple users over the network. The rating profiles are distributed over the network by the algorithm described in section 2.1.

With regard to most collaborative databases, the Movielens data is relatively dense. The recently published data set from Netflix [47] has a density of 1.2% and we expect that sites that allow people to share their own content (e.g. Tribler, YouTube, Flickr) are even more sparse. We therefore also evaluate the effect of sparsity on the optimal parameters for our model, by removing increasing parts of the Movielens data.

2.4.2 Evaluation Metric

We use the mean absolute error (MAE) as evaluation criterion. This metric is defined as the average absolute difference between predicted rating and the actual rating.

$$MAE = \frac{\sum_{j=0}^{M} |\hat{r}_j - r_j|}{M}, \tag{2.7}$$

where M are all the predicted ratings.

This metric has often been used in the evaluation of CF algorithms, we therefore see it as an adequate assessment that makes our results comparable to other findings.

2.4.3 Baseline Scores

Figure 2.3 shows the difference between recommendation predictions normalized by mean (Equation 2.1) or by both mean and standard deviation (Equation 2.2). Here

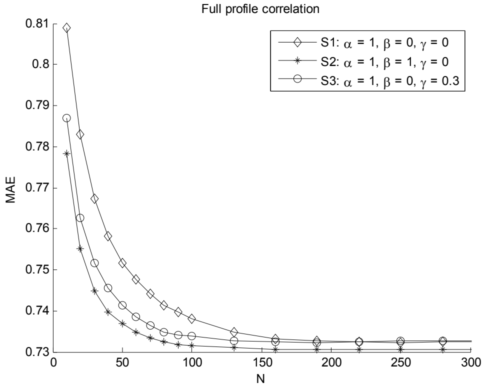


Figure 2.4: Mean absolute error for different settings of the utility parameters, using the full correlation similarity function. Setting 1 (S1) shows the results when only the similarity function is used. In setting 2 and 3 the utility model is respectively optimized for either β or γ.

the full profile correlation is used as a similarity measure and both the confidence and usefulness factor are set to zero. We see that compensating for a difference in rating spread improves the recommendation scores. We will therefore only use Equation 2.2 for recommendations in the rest of the chapter.

If no recommendation can be given for a certain item, because none of the N selected similar users has given a rating for this item, we estimate the rating by the mean rating of the target user. The MAE achieved by always imputing the user's mean as estimation is also shown in Figure 2.3.

To compare the neighourhood to that of a p2p system that does not utilize the users' taste profiles, we show the scores achieved with randomly selected peers, contributing equally to the estimation of the rating in Equation 2.2 ($w^*(u,v) = 1$). This recommendation score eventually converges to the one where all predictions are based on the average rating of the items.

Traditional CF research, in which the top-N selected users is allowed to change for each recommendation [52], often shows an optimal value of N after which the result starts to decline. We notice that this optimal value is not present in our results. This difference is explained by the fact that due to missing data the number of ratings that contribute to our recommendations (*effective* N) is significantly smaller than the N users we select. Therefore, when we increase N we are improving the prediction for some items and at the same time we downgrade the prediction for others, because the real number of contributing ratings differs per item. According to the results of Herlocker et al. [52], where exactly N ratings are used in each recommendation, the optimal number of contributing ratings is around 20-60. If we select a neighborhood of 100, some predictions will be relying on fewer than 20 ratings, while others are computed on more then 60. Eventually, the MAE curves stabilize for large N, because the weight function ($w^*(u,v)$) only decays for increasing N.

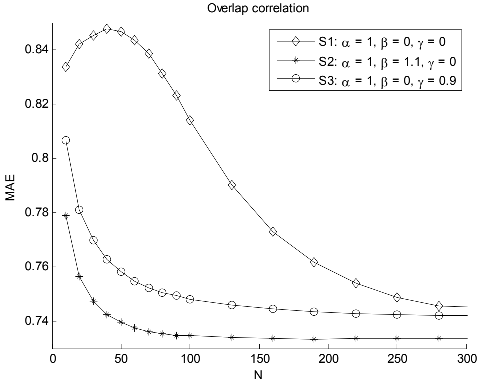


Figure 2.5: Mean absolute error for different settings of the utility parameters, using the overlap correlation similarity function.

2.4.4 Utility Model

Figures 2.4 and 2.5 show the MAE, using respectively the full correlation and the overlap correlation as similarity measures. Here α has been set to 1 while β and γ are individually optimized to achieve the lowest error. This means that we set either β or γ to zero, and find the optimal setting for the other parameter. We explicitly state that these results are obtained with retrospective experiments – parameters have been optimized on the test set – so the results demonstrate only that the introduced parameters *can* improve recommendation scores. Because the optimal parameter settings differ for each data set the model has to be optimized in each new setting.

We observe that both the confidence factor and the usefulness factor can improve the results, especially for small N (Fig. 2.4 and 2.5, setting 2/3). In the P2P setting this means that we can achieve the same error while storing much less data on the user's local machine.

Optimizing the results over all three weight factors (setting 4) did not improve the MAE over the results we obtain by just increasing the confidence (setting 2), we therefore do not show this result in Figure 2.4 and 2.5. In Section 2.4.5 we will show that if the data is more sparse the addition of the usefulness factor does improve the results over the combination of only similarity and confidence.

Comparing the full correlation measure with the correlation on the overlapping part of the ratings shows that the full profile correlation achieves slightly better results (minimal MAE 0.7335 vs. 0.7305). If the correlation is computed on only the overlapping part of two users' ratings, the normalization step in the correlation measure can produce unexpected results, because in a sparse data set the overlapping part usually constitutes only a fraction of a user's ratings. In Figure 2.6 we show two partially overlapping profiles, where user 1 clearly dislikes the items both users have rated and user 2 clearly likes these items. A computation of the correlation on the overlapping part would however result in a similarity close to one, while a full correlation correctly normalizes the profiles maintaining the distance between these items.

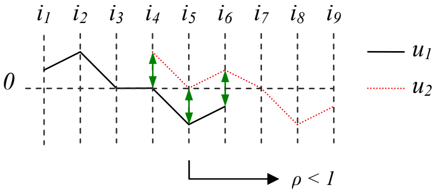


Figure 2.6: Two partially overlapping rating profiles can get a very high correlation on the overlapping part by accident. When the correlation between the entire profiles is computed, the profiles are correctly normalized so that the distance between different ratings is preserved.

Furthermore, we notice that without the influence of the confidence and usefulness factor (setting 1) the MAE for the overlap correlation first increases before it starts to improve. The full profile measure always improves when the neighborhood size increases. This difference is caused by the fact that the overlap correlation can give a high similarity score to users with only a few commonly rated items. The full correlation measure however, depends already on the length of the overlap, because after the normalization step the correlation measure is reduced to:

$$\rho_{u,v-\mathit{full}} = \frac{\sum_{i \in I} \tilde{r}_{u,i} \cdot \tilde{r}_{v,i}}{|I|}, \tag{2.8}$$

where $\tilde{r}_{u,i}$ is the rating of user u for item i after normalization by the user's mean and standard deviation. I are the items rated by any of the users in the top-N of the target user, u is the target user and v is the potential candidate for the recommendation.

Because of the normalization the imputed mean ratings at the not rated positions evaluate to zero, therefore the inner product of the rating profiles is exactly the same as the inner product of the overlapping part of the profiles:

$$\rho_{u,v-\mathit{full}} = \frac{\sum_{i \in (I_u \cap I_v)} \tilde{r}_{u,i} \cdot \tilde{r}_{v,i}}{|I|}, \tag{2.9}$$

where $I_u \cap I_v$ is the overlapping part over the two users' rating profiles.

Because the factors $|I|$ and $|I_u|$ are both constants with respect to the target user, we can see that Eq. 2.9 is proportional to the overlap correlation times the confidence factor ($\beta = 1$):

$$\rho_{u,v-ol} \times P(I_v|I_u) = \frac{\sum_{i \in (I_u \cap I_v)} \tilde{r}_{u,i} \cdot \tilde{r}_{v,i}}{|I_u \cap I_v|} \times \frac{|I_u \cap I_v|}{|I_u|} \tag{2.10}$$

Concluding, the full profile distance differs from the overlap correlation measure by the difference in normalization and $\beta = 1$.

Another observation is that in the computation of the full profile correlation *short* rating profiles are normalized by a smaller standard deviation than *long* profiles. Users who did not rate many items are therefore favored with respect to frequently rating users. This effect is contrary to that of our confidence and usefulness factor that aim at selecting users with many ratings. Because users with a short rating profile

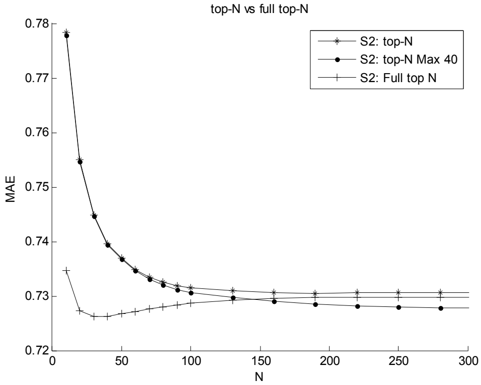


Figure 2.7: This figure shows the MAE when all ratings of the selected N users are included in the recommendation, and when the number of ratings is limited to 40. *Full top-N* shows the MAE when exactly N ratings are used for each recommendation.

are usually bad recommendation candidates, the full profile measure can still benefit from the extra confidence or usefulness factor ($\beta, \gamma > 0$).

In section 2.4.3 it was suggested that the optimal number of contributing ratings lies somewhere between 20 and 60. Figure 2.7 shows that if for each user the number of locally stored ratings is limited to 40 per item, the MAE slowly converges to the optimal score obtained when the recommendation is computed on exactly N ratings (*Full top-N*). In this way, less data needs to be stored locally and better recommendation performance can be reached for large N. We do however notice that the number of positively contributing ratings depends on the size and nature of the network. This number should therefore not be taken as optimal value in a different network or database.

2.4.5 User Selection in Sparse Data

The usefulness factor was introduced in order to select peers that have given a rating to many items in the network. In this way the recommendations will depend on more information and therefore be a more accurate prediction of the target user's rating. Because the Movielens data is however relatively dense, we do not suffer much from very empty rating profiles. To test the influence of the usefulness parameter on more sparse data, we have subsequently removed 0%, 25%, 50%, 75% and 90% of the ratings from half of users in the training set. We then compute the average MAE achieved for small top-N ($N \in \{10, 20, 30, 40, 50\}$), and we show the difference in mean MAE for increasing values of γ, compared to $\gamma = 0$. Figure 2.8 shows that the influence of the usefulness parameter increases with increasing sparsity. Also, the value of γ for which the optimal result is reached slightly increases.

As another argument for our usefulness parameter we show the number of ratings that contribute to the recommendation for different numbers of N in Figure 2.9. For neighborhood sizes $N \in \{10, 20, 30, 40, 50\}$ this figure shows the median of the number of users that has rated the items for which we want to predict a rating (*effective*

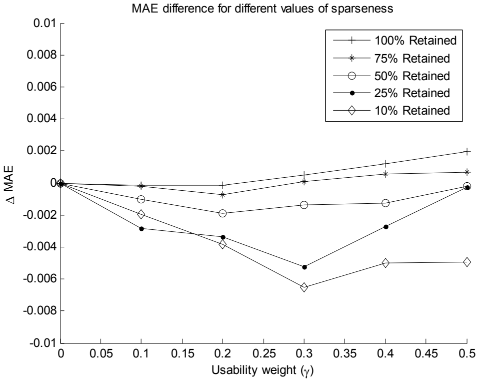


Figure 2.8: For different values of usefulness (γ) we show the difference in MAE compared to $\gamma = 0$. The MAE is here computed as the mean of the MAE for $N \in \{10, 20, 30, 40, 50\}$. We compare the results for different levels of sparsity and notice that the addition of the usefulness factor has much more influence when the data is sparse. Here we have set α,$\beta = 1$.

N). We show the *effective* N for the Movielens data without rating removal (*Dense data*) and utility parameter setting 2, for 90% removal with utility parameter setting 2 (*Sparse:* $\gamma = 0$) and 90% data removal with utility parameter setting 4 (*Sparse:* $\gamma = 0.3$).

The figure shows that in the original (dense) data set about 40% of the selected users contribute in the recommendation for a certain item, when a small neighborhood is considered. In the sparse data, about 20% of the users in the selected neighborhood have given a rating for the target item. If random users would have been chosen these values would have been around 5% for dense data and 2.8% for the sparse data. This demonstrates that the weight function without the addition of the usefulness parameter already selects users that have rated a similar set of items as the target user, which indicates a common interest in movies. An increase in γ as expected selects a neighborhood with more ratings, which explains the MAE improvement when a small neighborhood is selected in a sparse data set.

2.5 Conclusion

The selection of a peer neighbourhood, containing similar users, is a vital function for a social p2p client, since the taste neighbourhood can improve both download speed and content recommendations. In this chapter, we have investigated the performance of common collaborative filtering techniques for deployment in a decentralized environment. In this setting each user has a locally optimized cache of rating profiles for recommendation. The weighing factor of the collaborative filtering system is seen as the resource selection method to create the social overlay network and the updating criterion for the user's local cache. After the selection of a user's neighborhood, this set of users is fixed and can not be updated for each individual prediction. This setting results in a varying number of ratings per prediction and it can even occur that no

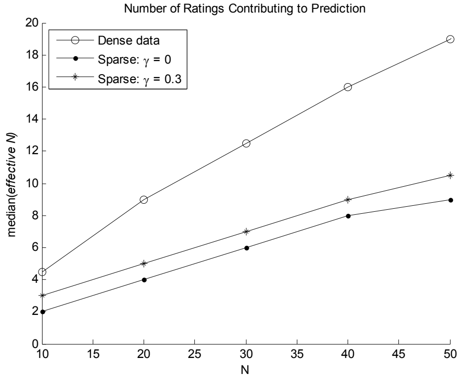


Figure 2.9: The real number of peers that contributes to the recommendation for different numbers of N. Results are shown for *Dense* data (α,$\beta = 1$; $\gamma = 0$), *Sparse* data without usefulness factor (α,$\beta = 1$; $\gamma = 0$) and *Sparse* data with usefulness factor (α,$\beta = 1$; $\gamma = 0.3$). The full profile correlation is used as similarity measure.

collaborative recommendation can be made for a specific item (because no users with ratings have been selected for this item).

We have proposed a *utility model*, that extends the common similarity functions by including a variable confidence and usefulness factor. The confidence factor compensates for the fact that similarities based on many ratings are more reliable than those based on only a few. The usefulness factor increases the number of ratings that contribute to the recommendation, by favoring users with many given ratings over those who just rated a few items. We have compared two similarity functions, and we have shown how our utility model can improve the recommendation accuracy.

Most user-based CF systems use the Pearson correlation, computed on the overlapping part of the users' rating profiles, as a similarity measure. Often, this similarity is combined with a confidence factor ($\beta = 1$). We have shown that the full profile correlation (with imputed mean ratings at the empty positions) can improve the prediction results, because this measure normalizes the rating profiles on all the ratings given by the users. In the computation of the overlap correlation, normalization inaccuracies can occur because the overlapping part constitutes only a fraction of the users' ratings.

Although the full profile correlation already encapsulates a weighing factor that compensates for the length of the overlapping part of the rating profiles, we have shown that this measure can still benefit from the additional confidence factor. We are not aware of any other work that combined the full correlation method with an extra confidence factor, we therefore suggest that this method can also improve recommendations in a centralized system (where the top-N can be adjusted for each recommendation).

In a sparse data environment the predicted recommendation often depend on a small portion of the selected peers. In order to provide enough locally stored ratings to estimate the taste of the target user, we have introduced the usefulness factor. We have shown that this factor becomes important if the data is very sparse and the locally stored rating profiles cannot be updated for each prediction.

Finally, if the optimal number of positively contributing ratings is known for a specific dataset, no more than this number of ratings should be maintained locally in order to optimize recommendations and minimize the local cache. The true number of useful peers is however highly dependent on the size and the nature of the network. Therefore, a thorough investigation of the network aspects is essential for a good recommendation system.

3

Exploiting Positive and Negative Graded Relevance Assessments for Content Recommendation

Social media allow users to give their opinion about the available content by assigning a rating. Collaborative filtering approaches to predict recommendations based on these graded relevance assessments are hampered by the sparseness of the data. This sparseness problem can be overcome with graph-based models, but current methods are not able to deal with negative relevance assessments.

We propose a new graph-based model that exploits both positive and negative preference data. Hereto, we combine in a single content ranking the results from two graphs, one based on positive and the other based on negative preference information. The resulting ranking contains fewer false positives than a ranking based on positive information alone. Low ratings however appear to have a predictive value for relevant content. Discounting the negative information therefore does not only remove the irrelevant content from the top of the ranking, but also reduces the recall of relevant documents.

This chapter is published as: Maarten Clements, Arjen P. de Vries, and Marcel J. T. Reinders. Exploiting Positive and Negative Graded Relevance Assessments for Content Recommendation. In K. Avrachenkov, D. Donato, and N. Litvak, editors, *WAW09: Proceedings of the 6th International Workshop on Algorithms and Models for the Web-Graph, LNCS 5427*, pages 155-166. Springer-Verlag, Berlin, Heidelberg, February 2009.

3.1 Graded Relevance Assessments

The popularity of online social media encourages users to manage their online identity by active participation in the annotation of the available content. Most of these systems allow their users to assign a graded relevance assessment by giving a rating for a specific content element. Many people use these ratings in order to convey their opinion to the other network users, or to organize their own content to gain easy access to their favorite files.

Collaborative filtering methods use the created rating profiles to establish a similarity between users or items. This similarity is often based on the *Pearson correlation* which has proven to be an effective measure to incorporate positive and negative feedback while compensating for differences in offset or rating variance between users [12; 53; 112; 133]. Because these similarity functions derive the similarity based on the overlapping part of the users' rating profiles these methods perform poorly in sparse data spaces [113; 59].

Graph-based methods have shown to effectively deal with extremely sparse data sets by using the entire network structure in the predicted ranking. Most of these methods have been developed to estimate a global popularity ranking in graphs with a single entity type, like websites [99; 70]. These methods are generally not adapted to negative relevance information and do not provide personalized rankings for each network user.

Using a personalized random walk over two graphs we separately compute a ranking based on positive and negative preference information. We combine these two rankings and compare the result on two real data sets. We discuss the positive and negative effects of the proposed method compared to recently proposed graph-based ranking models.

3.2 Graph Combination Model

We define two bipartite graphs $G^+ = \langle V, E^+ \rangle$ and $G^- = \langle V, E^- \rangle$ where the set of vertices consists of all users and items $V = U \cup I$ (U is the set of users $u_k \in U$ (with $k \in \{1, \ldots, K\}$) and I is the set of items $i_l \in I$ (with $l \in \{1, \ldots, L\}$)). The set of edges (E^+/E^-) consists of all user-item pairs $\{u_k, i_l\}$. The weight of the edges is determined by the value of the rating, which will be discussed in Section 3.3.

We propose to use a random walk model to obtain a ranking of the content in both graphs. A random walk can be described by a stochastic process in which the initial condition (S_n) is known and the next state (S_{n+1}) is given by a certain probability distribution. This distribution is represented by a *transition matrix* $\mathbf{A}$, where $a_{i,j}$ contains the probability of going from node i (at time n) to j (at time $n + 1$):

$$a_{i,j} = P(S_{n+1} = j | S_n = i) \tag{3.1}$$

The initial state of all network nodes can now be represented as a vector $\mathbf{v}_0$ (with $\sum_i \mathbf{v}_0(i) = 1$), in which the starting probabilities can be assigned. By multiplying the state vector with the transition matrix, we can find the state probabilities after one step in the graph ($\mathbf{v}_1$). Multi step probabilities can be found by repeating the multiplication $\mathbf{v}_{n+1} = \mathbf{v}_n\mathbf{A}$, or using the n-step transition matrix $\mathbf{v}_n = \mathbf{v}_0\mathbf{A}^n$. The number of steps taken in the random walk determines the influence of the initial state on the current state probabilities.

The random walk is a Markov model of order 1 (or *Markov chain*), because the next state of the walk only depends on the current state and not on any previous states, which is known as the *Markov property*:

$$\begin{aligned} &P(S_{n+1} = x | S_n = x_n, \ldots, S_1 = x_1) = \\ &P(S_{n+1} = x | S_n = x_n) \end{aligned} \tag{3.2}$$

If $\mathbf{A}$ is stochastic, irreducible and aperiodic, $\mathbf{v}$ will become stable, so that $\mathbf{v}_\infty = \mathbf{v}_\infty\mathbf{A}$ [142]. These limiting state probabilities represent the prior probability of all nodes in the network determined by the volume of connected paths [127]. In order to make these conditions true, we ensure that all rows of $\mathbf{A}$ add up to 1 by normalizing the rows, that $\mathbf{A}$ is fully connected, and that $\mathbf{A}$ is not bipartite.

We include self-transitions that allow the walk to stay in place, which increases the influence of the initial state. The self-transitions are represented by the identity matrix $\mathbf{S} = I$, so that the weight of the self-transitions is equal for all nodes.

We distinguish the transition matrix based on positive and negative ratings ($\mathbf{T}^+$ and $\mathbf{T}^-$). The random walk over the positive graph estimates the ranking of relevant documents, while the walk over the negative graph estimates the ranking of most irrelevant documents. We create the positive transition matrix as follows:

$$\mathbf{T}^+ = \begin{bmatrix} \alpha\mathbf{S}_K & (1-\alpha)\mathbf{R}^+ \\ (1-\alpha)\mathbf{R}^{+T} & \alpha\mathbf{S}_L \end{bmatrix}$$

where $\mathbf{R}^+$ contains the positive preference information (See Section 3.3). To make sure the graph is fully connected we add an edge with weight ϵ between all node pairs, which allows the walk to teleport to a random node at each step. The final transition matrix is now given by: $\mathbf{A}^+ = (1-\epsilon)\mathbf{T}^+ + \epsilon\frac{\mathbf{1}_{KL}}{K+L}$, where $\mathbf{1}_{KL}$ represents the ones matrix of size $K + L$. The teleport probability ϵ is set to 0.01 in all experiments. The negative transition matrix is constructed similarly.

We now create the initial state vector with a zero array of length $K + L$ and set the index corresponding to the target user to one $\mathbf{v}_0(u_k) = 1$. Multiplying the state vector with one of the transition matrices gives either the estimation of relevant ($\mathbf{v}_1^+ = \mathbf{v}_0\mathbf{A}^+$) or irrelevant ($\mathbf{v}_1^- = \mathbf{v}_0\mathbf{A}^-$) content for the target user. The first step gives the content annotated by the user himself, while subsequent steps ($\mathbf{v}_n^+$;$\mathbf{v}_m^-$) give an estimate of the most similar users and content.

Both random walks produce a state probability vector which indicates either the positive or negative information that we have about each node. To obtain a single content ranking, we combine the parts of the state vectors that correspond to item nodes ($\mathbf{v}^+(K+1, ..., K+L)$ and $\mathbf{v}^-(K+1, ..., K+L)$). The combined content ranking is obtained by simply subtracting the negative state probabilities from the positive state probabilities ($\mathbf{v}_n^+ - \mathbf{v}_m^-$). Intuitively, this subtraction ranks the content according to the difference in positive and negative information in the neighborhood of the target user.

3.2.1 Self-transition (α) and Walk Length (n)

The number of steps in the random walk (n) determines how strongly the final ranking depends on the starting point (target user u_k). The speed of convergence is determined by the self-transition probability α. Because all nodes have equal self-transition probability, the total number of non-self steps (Q) after n steps through the graph is a binomial random variable with probability mass function (PMF):

$$P_Q(q) = \begin{cases} \binom{n}{q}\alpha^q(1-\alpha)^{n-q} & q = 0, \ldots, n \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

Where $P_Q(q)$ is the probability of q non-self steps ($Q = q$).

If a large value is chosen for α, most of the probability mass will stay close to the starting point. A small value of α results in a walk that quickly converges to the stable state probability distribution. Based on earlier experiments we fix the self-transition probability (α) to 0.8 [23; 29].

3.3 Data

3.3.1 LibraryThing (LT)

LibraryThing[1] is a social online book catalog that allows its users to indicate their opinion about their books by giving a rating. Based on these preference indications LT gives suggestions about interesting books to read and about people with similar taste. The popularity of the system has resulted in a database that contains over 3 million unique works, collaboratively added by more than 400,000 users.

We have collected a part of the LibraryThing network, containing 25,295 active users[2]. After pruning this data set we retain 7279 users that have all supplied a rating to at least 20 books. We remove books that occur in fewer than 5 user profiles, resulting in 37,232 unique works.

The user interface of LibraryThing allows users to assign ratings on the scale from a half to five, the distribution of the ratings in our LT data sample is shown in Figure 3.1a.

3.3.2 MovieLens (ML)

To validate the reproducibility of our results, we also use the data set from MovieLens[3], which is a well known benchmark data set for collaborative filtering algorithms. ML consists of 100,000 ratings for 1682 movies given by 943 users. In this data, ratings have been given on a scale of 1 to 5, 1 being dreadful and 5 excellent. Figure 3.1b shows the rating distribution in the ML data.

[1] http://www.librarything.com

[2] Crawled in July 2007

[3] http://www.grouplens.org/

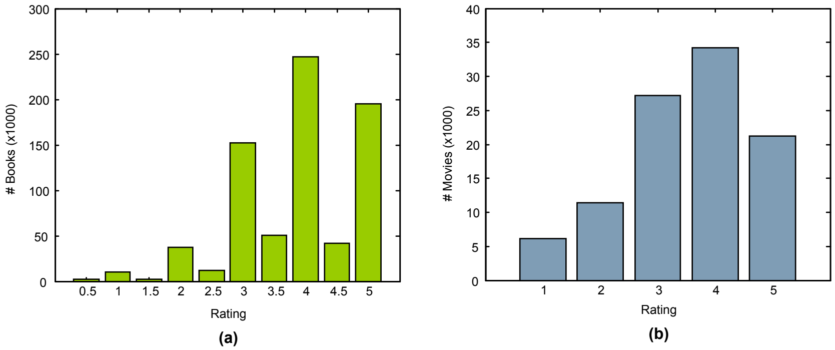


Figure 3.1: Rating distribution of **a)** LibraryThing and **b)** MovieLens.

3.3.3 Rating to Edge Weight

Figure 3.2 gives the edge weights we assign to various ratings. The positive rating matrix $\mathbf{R}^+$ integrates the ratings $3 - 5$ and the negative rating matrix $\mathbf{R}^-$ contains the ratings $\frac{1}{2} - 2\frac{1}{2}$ where the largest weight is assigned to the lowest rating. Although a 3 is the average rating that can be given in most user interfaces (by clicking a number of stars) it is generally regarded as *slightly positive*, because more than half of the stars are filled when a user has clicked the third star.

The LT data consists of a total of 749401 ratings, using the split between positive and negative ratings as indicated in Figure 3.2, $\mathbf{R}^+$ will have a density of $2.53 \cdot 10^{-3}$ and $\mathbf{R}^-$ has a density of $2.34 \cdot 10^{-4}$. The MovieLens data has a much lower positive bias and the indicated data split results in $\mathbf{R}^+$ with density of $5.20 \cdot 10^{-2}$ and $\mathbf{R}^-$ with density of $1.10 \cdot 10^{-2}$. The difference in positive bias can be explained because watching a movie is a social experience while reading is not. Users will therefore watch more movies they do not like (because your friends want to see it) than read books they do not like.

The resulting graphs (G^+, G^-) have a clear power-law structure, which is common to socially organized data [93]. However, the long tail is reduced due to the pruning step in which the users and items with few connections were removed.

LibraryThing MovieLens	½	1 1	1½	2 2	2½	3 3	3½	4 4	4½	5 5
Edge Weight	5	4	3	2	1	1	2	3	4	5
	$\mathbf{R}^-$					$\mathbf{R}^+$				

Figure 3.2: Conversion of ratings to edge weights.

3.4 Experimental Setup

3.4.1 Data Split

To obtain a fair comparison without overfitting the model to the data we split the data sets in two equal parts, see Figure 3.3. First we use the training users to estimate the optimal model parameters, by finding the optimal value for our evaluation criterion. We remove 1/5 of the ratings of 1/5 of the training users (validation set) and use the rest of the data to predict the missing values. We average the results over 5 different independent splits.

Using the optimal model parameters we evaluate the different models on the test set. We again remove 1/5 of the user profiles and use a 5-fold cross validation to obtain stable results.

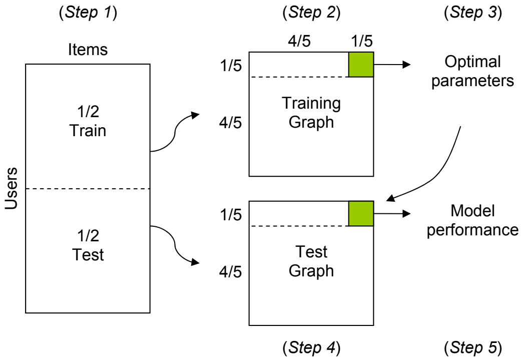


Figure 3.3: Splitting the data in a train and test set.

3.4.2 Evaluation

3.4.2.1 NDCG

To evaluate the predicted content ranking we use the Normalized Discounted Cumulative Gain (NDCG) proposed by Järvelin and Kekäläinen [62].

We first create a gain vector G with length L (all items) of zeros. In this gain vector the predicted rank positions of the held-out validation items are assigned a value equal to the edge weights in the training graph (See Figure 3.2), called the *gain*.

In order to progressively reduce the gain of lower ranked test items, each position in the gain vector is discounted by the $\log_2$ of its index i (where we first add 1 to the index, to ensure discounting for all rank positions > 0). The Discounted Cumulative Gain (DCG) now accumulates the values of the discounted gain vector:

$$\mathrm{DCG}[i] = \mathrm{DCG}[i-1] + G[i]/\log_2(i+1) \tag{3.4}$$

The DCG vector can now be normalized to the optimal DCG vector. This optimal DCG is computed using a gain vector where all test ratings are placed in the top of the vector in descending order. Component by component division now gives us the NDCG vector in which each position contains a value in the range $[0, 1]$ indicating the level of perfectness of the ranking so far. We use the area below the NDCG curve as score to evaluate our rank prediction.

We want to evaluate the prediction of relevant content with respect to the prediction of irrelevant content. We separately compute the predictive value for positive test items (NDCG$^+$) and negative test items (NDCG$^-$) and use the fraction of the two NDCG measures as evaluation method: NDCG$^+$/NDCG$^-$. This measure will be optimal if the predicted ranking contains the positive test items at the top (in descending rating order) and the negative test items at the bottom.

3.4.2.2 PPV@20

The *positive predictive value* indicates the fraction of recommended relevant documents (*true positives*, TP) with respect to incorrectly recommended irrelevant documents (*false positives*, FP):

$$PPV = \frac{TP}{TP + FP} \tag{3.5}$$

We assume that the system will give 20 recommendations and therefore compute the PPV on the top 20 of the ranked list (PPV@20).

Compared to *precision*, which is defined as TP/n (where n is the number of recommended documents), PPV does not regard the unassessed recommended items as incorrect recommendations. Because we are interested in evaluating the number of relevant compared to negatively assessed documents we use PPV as evaluation method.

3.4.2.3 Recall@20

Recall indicates the number of true positives with respect to all relevant documents in the database and is defined as:

$$Recall = \frac{TP}{TP + FN} \tag{3.6}$$

where FN indicates the number of unrecommended relevant documents (*false negatives*).

3.5 Experiments

3.5.1 Optimizing Relevance Ranking

We separately optimize the ranking of positive and negative content on the training set. We first look at the NDCG$^+$ for increasing walk length on the positive graph using $\mathbf{v}^+$ to obtain the content ranking. Figure 3.4a shows that for both LT and ML the optimal ranking is achieved after only 5 steps through the graph. The NDCG$^+$ quickly converges to a stable value when the state vector reaches the global content popularity.

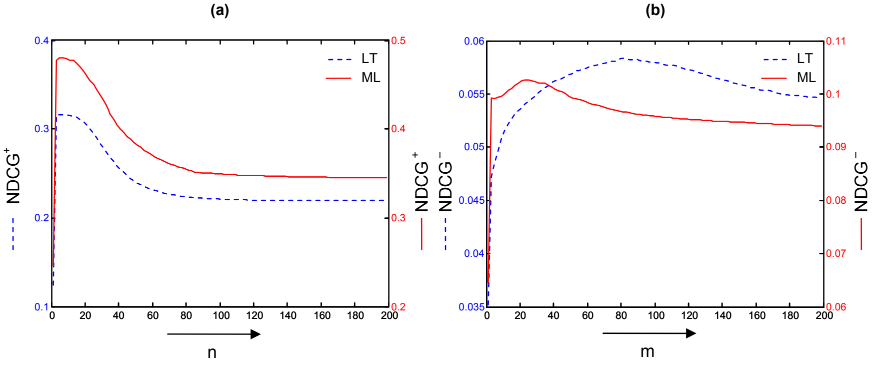


Figure 3.4: a) Optimization of the walk length over the positive rating graph (Max. at $n = 5$ for both data sets). **b)** Optimization of the prediction of the prediction of negative test items (Max. at $m = 81$ for LT and $m = 23$ for ML).

The absolute difference between the performance on the two data sets can be explained by two factors. The ML users on average rated a larger fraction of the available items. Therefore, the probability of finding a relevant test item at the top of the ranking is higher, independent from the used method. Also, the more extensive user profiles result in denser social graphs, allowing the model to make more accurate predictions.

3.5.2 Optimizing Irrelevance Ranking

Figure 3.4b shows the NDCG^{-} (prediction of irrelevant content) optimized on the negative rating graph (ranking based on $\mathbf{v}_m^-$). It is clear that a longer walk over the graph is needed to obtain an optimal prediction. Also, the optimal negative ranking is reached for a smaller number of steps on the ML data than on the LT data. This is expected because the ML data contains more negative ratings; in other words, the negative graph of ML is much more dense than the LT negative graph. On a dense graph the state probability vector of the random walk will converge more quickly to the stable distribution (i.e., the graph has a shorter *mixing time*). Because the random teleport probability is very low ($\epsilon = 0.01$) it will only slightly decrease the mixing time.

3.5.3 Test Results

Table 3.1 summarizes the evaluation results on the test set for recommendation using different model settings. We first compare our proposed model using the difference of state probabilities as ranking function ($\mathbf{v}_n^+ - \mathbf{v}_m^-$) to the ranking based on positive information alone ($\mathbf{v}_n^+$). We now use the optimal settings of the walk length parameters m and n derived from the individual optimizations on the training set. If we can correctly predict both positive and negative content, subtracting the probability of reaching a node in the negative graph from the state probability in the positive graph will give a ranking with good content at the top and bad content at the bottom.

Table 3.1: Test results for both datasets. NDCG is abbreviated with N.

Data	Method	Evaluation measure			
		N$^+$	N$^+$/N$^-$	PPV@20	Recall@20
LT	$\mathbf{v}_5^+$	0.310	5.279	0.973	0.474
	$\mathbf{v}_5^+ - \mathbf{v}_{81}^-$	0.195	7.466	0.976	0.362
	$\mathbf{v}_5$	0.318	4.909	0.967	0.496
ML	$\mathbf{v}_5^+$	0.491	3.538	0.944	0.596
	$\mathbf{v}_5^+ - \mathbf{v}_{23}^-$	0.167	6.165	0.971	0.278
	$\mathbf{v}_5$	0.508	3.246	0.925	0.627

Our proposed combination model outperforms the ranking based on positive information if we use the fraction NDCG$^+$/NDCG$^-$ or PPV@20 as evaluation measure. This means that the top of the ranking contains relatively more positive test items than negative test items. However, we also observe a large drop in recall (and NDCG$^+$), meaning that our method finds a lower absolute number of relevant test items. Apparently the use of the negative graph not only removes irrelevant content from the top of the ranking, but also penalizes some of the relevant content.

Alternative model $\mathbf{v}_n$ is obtained using all ratings as positive evidence in the transition matrix (Rating $\frac{1}{2} \ldots 5$ mapped to edge weights $1 \ldots 10$ in $\mathbf{R}$). The test results show that this method has a higher PPV@20 and NDCG$^+$ than the ranking based on $\mathbf{v_n^+}$. This shows that the negative training items even have a small predictive value for relevant content.

3.5.4 Understanding the Test Results

Figure 3.5 shows the position of the positive (Rating ≥ 3) and negative (Rating < 3) test items in the predicted content ranking, aggregated over all test users. The ranked list is split into bins of 100 items and the graphs plot the number of test items that fall into a certain bin. The gap between positive and negative ratings in the top part of the ranking is clearly larger in the combined model, based on $\mathbf{v}_n^+ - \mathbf{v}_m^-$ (Figure 3.5b,d) than in the purely positive model, based on $\mathbf{v}_n^+$ (Figure 3.5a,c). This finding corresponds to the increase in NDCG$^+$/NDCG$^-$.

As expected by previously discussed results, we observe a peak at the bottom of the ranking in Figure 3.5b and 3.5d, both in negative and positive test items. This confirms the effect that the negative graph also penalizes some of the relevant content, meaning that some of the relevant content has more connections to the target user in the negative graph than in the positive graph.

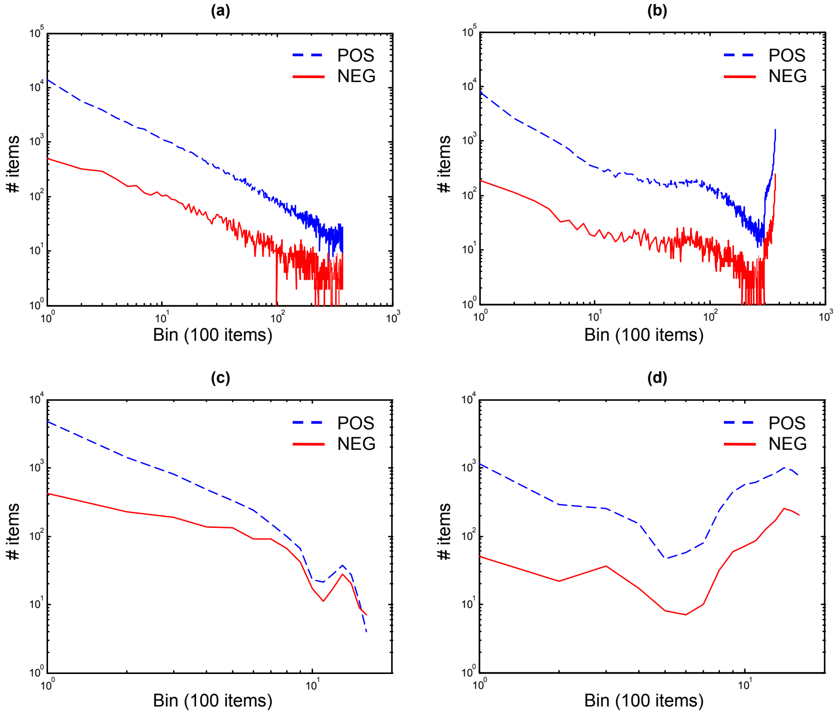


Figure 3.5: a) LT: Aggregated ranking using only the positive graph after 5 steps ($\mathbf{v}_5^+$). **b)** LT: Aggregated ranking for the combined method ($\mathbf{v}_5^+ - \mathbf{v}_{81}^-$). **c)** ML: Aggregated ranking using only the positive graph after 5 steps ($\mathbf{v}_5^+$). **d)** ML: Aggregated ranking for the combined method ($\mathbf{v}_5^+ - \mathbf{v}_{23}^-$).

3.6 Discussion

3.6.1 Graph-based Ranking Models

Graph-based algorithms have shown to be very effective to find a global ranking of hyperlinked documents in the web graph [99; 70]. Also in other domains have these methods shown to be useful ranking mechanisms.

Gyöngyi et al. adapted traditional PageRank in order to reduce the rank position of spam web-sites [46]. Analogous to our approach they try to find an optimal document ranking in a graph with many unjudged documents. The small subset of documents that has received a relevance judgement is used as seed of the random walk. Besides the difference in domain, this method mostly differs from our model in the fact that the authors assume a global binary opinion on the content quality, while our approach is based on the individual preference annotations.

Gori and Pucci used the graph of referenced scientific research papers to obtain a paper ranking, based on a user's history [45]. User-based recommendation algorithms were described as a graph-theoretic model by Mirza et al. [87]. In their model the graph is represented by *hammocks* (the set of 2-step connections between 2 nodes), based on rating commonality between users. By taking multiple steps over the user similarity graph their algorithm finds latent relations between users. These models are based on graphs with a single type node (items and users relatively), an extra step needs to be taken to relate the content to the target user.

Different methods have been proposed that represent both users and items into a single graph. Huang et al. applied spreading activation algorithms on the user-item graph to explore transitive associations among consumers through their past transactions and feedback [59]. Fouss et al. [37] removed the diffusion parameter and used the average commute time between nodes in the graph-based representation of the MovieLens database to derive similarities between the entities in the data. These algorithms showed to be very effective on binary relevance data. They ignored the numeric value of the rating provided by the user and only used the fact that a user did or did not see/buy the content (binary relevance assessments).

The random walk model with self-transitions has been applied on the graph constructed by graded relevance information based on queries and clicks on images [29]. In this work Craswell and Szummer explained the soft clustering effect that is obtained with a medium length random walk. This effect is clearly visible in our results on the negative rating graph, where an average length walk finds the cluster of irrelevant content and therefore outperforms direct relations or the popularity ranking.

We have to the best of our knowledge for the first time used a graph-based ranking model on the positive and negative user-item graph. Because of the different graph statistics we used the walk length parameter to individually optimize the prediction of relevant and irrelevant content. The combined model showed that the information in negative relevance assessments can be used to improve the positive predictive value of the content ranking, by pushing some documents to the bottom of the ranking.

3.6.2 Selective Assessment Explains Positive Predictive Value

We have shown that negative preference indications not only predict irrelevant content, but also have a predictive value for positively rated test items. This can be explained by the fact that users in a social content systems do not randomly select the content to assess. People carefully select the content to read/view based on prior knowledge about theme, author etc. Based on this prior knowledge the user assumes that he will like the content (otherwise he would not view it).

Although the user gives a low rating to the selected content, this content can still be related to other documents the user does like, because of features corresponding to prior knowledge. In those cases, negative items are connected to books that the user would give a high rating. In the negative graph, these items incorrectly drag some of the relevant content with them to the bottom of the ranking. Perhaps, modeling more aspects of the content will separate the relevant and irrelevant recommendations.

Part II

Tags

4

The Task Dependent Effect of Tags and Ratings on Social Media Access

Recently, online social networks have emerged that allow people to share their multimedia files, retrieve interesting content, and discover like-minded people. These systems often provide the possibility to annotate the content with tags and ratings.

Using a random walk through the social annotation graph, we have combined these annotations into a retrieval model that effectively balances the personal preferences and opinions of like-minded users into a single relevance ranking for either content, tags or people. We use this model to identify the influence of different annotation methods and system design aspects on common ranking tasks in social content systems.

Our results show that a combination of rating and tagging information can improve tasks like search and recommendation. The optimal influence of both sources on the ranking is highly dependent on the retrieval task and system design. Results on content search and tag suggestion indicate that the profile created by a user's annotations can be used effectively to adapt the ranking to personal preferences. The random walk reduces sparsity problems by smoothly integrating indirectly related concepts in the relevance ranking, which is especially valuable for cold-start users or individual tagging systems like YouTube and Flickr.

This chapter is published as: Maarten Clements, Arjen P. De Vries, and Marcel J. T. Reinders. The Task Dependent Effect of Tags and Ratings on Social Media Access. *ACM Transactions on Information Systems*, 28(4), October 2010.

4.1 Introduction

The most widely discussed change in Internet usage over the last few years is the increase in interactivity and user contribution. Dynamic websites are used to distribute videos or photos, share opinions about books and movies, find interesting people or just maintain contact information of relations. People actively use the provided communication tools to get in touch with other network users and discuss their opinion on the available content. In this way, people build up an on-line identity and make new friends within the network. The addition of these social aspects in on-line databases has strongly increased their popularity, which in turn has resulted in massive unstructured data collections, created by the collaborative effort of regular Internet users.

Many of these networks focus on the distribution of content that does not carry a clear contextual description by itself. In these *social content systems*, many people are willing to participate in the annotation of otherwise difficult to retrieve files. People actively tag the available content and enjoy giving their opinion by supplying a rating. Although most people use tagging to organize their own content collection, it has been shown that social tagging results in semantically descriptive annotations that can be used for content retrieval by the entire network [44; 83].

Most of the work on tagging systems or *folksonomies* has been focused on the understanding of the social phenomena underlying the use of these systems. Much less research has evaluated the actual retrieval performance of the entities in social content systems. In this work we discuss the design issues in social content systems with respect to the effectiveness of common retrieval tasks. More specifically, we evaluate how the users' tagging and rating rights and the representation in the interface affect the possibilities to adapt retrieval tasks to the personal user preference.

To describe the retrieval tasks in a social content system, we suggest a taxonomy that can always provide the user with relevant content, tags and people (see Figure 4.1). The top level gives the tasks that exist when no context (query) is specified by the user. Many research activities have focused on the recommendation of items (T_1), often by first finding a group of similar users (T_3), which is known as *collaborative filtering* [12; 106]. Because most of these algorithms were developed on rated databases without tags, not much attention has gone to the recommendation of tags (T_2), which could be useful to initiate a browsing session.

The second level in our taxonomy indicates the view on the network after the user has selected either an item, tag or another user. In total, this describes twelve tasks that apply for personalization in a collaboratively tagged database. Including common tasks like: suggesting tags when interesting content has been found (T_5), retrieving relevant content by using tags as queries (T_7), getting help from experts on a certain topic (T_6,T_9), making new friends (T_{12}) and using your friends to discover relevant content (T_{10},T_{11}).

We propose to use a single model that serves all these tasks. To this end, we adopt a previously used random walk variant, and show that it can be used for all these tasks by slightly modifying the parameters. Because the random walk integrates latent relations in the relevance ranking, this model is robust against sparsity problems that often hamper collaborative filtering based approaches [113; 59] and problems arising

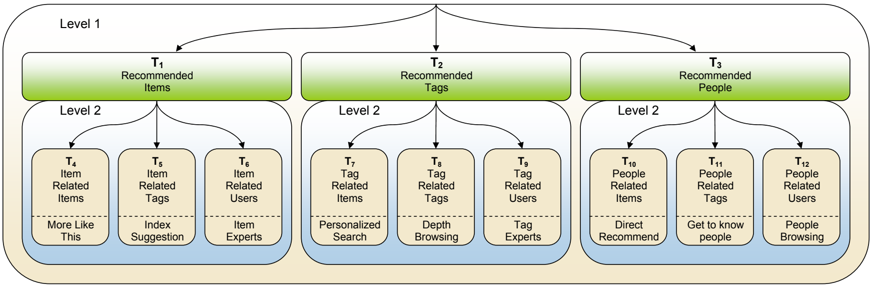


Figure 4.1: A taxonomy of the tasks in a social content system that apply for personalization. Level 1 shows the three tasks that apply to users that just enter the system (T_1-T_3). Level 2 indicates the tasks that arise after the user has selected either an item, tag or another user (T_4-T_{12}).

from synonymous terms in social tagging systems [8; 44]. We use the parameters in our ranking model to identify the effect of system design choices on the effectiveness of common retrieval tasks (Section 4.5). To show that this model is valid for our analysis we compare the ranking performance to recently proposed algorithms in Section 4.6 and discuss other related work in Section 4.7.

4.2 Personalization Model

For the relevance ranking of the tasks in our taxonomy we propose to use a random walk over the graph, created by all annotations. A random walk is a simple stochastic process in which the initial condition (S_n) is known and the next state (S_{n+1}) is given by a certain probability distribution. This distribution can be represented by a *transition matrix* $\mathbf{A}$, where $a_{i,j}$ contains the probability of going from node i (at time n) to node j (at time $n+1$):

$$a_{i,j} = P(S_{n+1} = j | S_n = i) \tag{4.1}$$

The state of the random walker is described by the probability distribution $\mathbf{v}_n$, where $\mathbf{v}_n(i) = P(S_n = i)$ and $\sum(\mathbf{v}_n) = 1$. The initial state probability of all network nodes can now be represented by $\mathbf{v}_0$, which indicates the starting points of the random walk. By multiplying the state vector with the transition matrix, we can find the state probabilities after one step in the graph ($\mathbf{v}_1 = \mathbf{v}_0\mathbf{A}$). Multistep probabilities can be found by repeating the multiplication $\mathbf{v}_{n+1} = \mathbf{v}_n\mathbf{A}$, or equivalently using the n-step transition matrix $\mathbf{v}_n = \mathbf{v}_0\mathbf{A}^n$. The number of steps taken in the random walk determines the influence of the initial state vector. If $\mathbf{A}$ is stochastic, irreducible and aperiodic, $\mathbf{v}$ will become stable (so that $\mathbf{v}_\infty = \mathbf{v}_\infty\mathbf{A}$) and it will contain the prior probability of all nodes in the network.

Figure 4.2 shows how we define the social graph and the transition matrix. We create a tripartite graph $G = \langle V, E \rangle$ where the set of vertices consists of all users, items and tags $V = U \cup I \cup T$ and the set of edges (E) is determined by the information derived from the social annotations. How we set the weight of the edges depends on the available information in the social content system and is discussed in Section 4.3.

Table 4.1: The most important symbols used in this chapter.

T_{1-12}	A retrieval task
U, u_k	The set of all users and user k
I, i_l	The set of all items and item l
T, t_m	The set of all tags, and tag m
$\mathbf{D}$	Three dimensional matrix containing tag assignments
$\mathbf{UI}, \mathbf{UT}, \mathbf{IT}$	Projections of $\mathbf{D}$
$\mathbf{R}$	Two dimensional matrix containing ratings
$\mathbf{A}$	The transition matrix
S	The state of the random walk
$\mathbf{v}$	The state probability distribution
n	The number of steps of the random walk
α	Self-transition probability
β	Probability of making a User-Tag step
γ	Probability of making an Item-Tag step
δ	Probability of making a Tag-Item step
θ	Initial state probability of the query element

We include self-transitions that allow the walk to stay in place, increasing the influence of the initial state. We set the weight of the self transitions equal for all nodes. This removes the tri-partite structure of the graph.

We normalize the weights of the outgoing edges so that the edges to each of the other node types sum up to one, and combine them in the transition matrix $\mathbf{A}$, using the parameters α, β, γ and δ as shown in Figure 4.2. In this model $\alpha \in [0, 1]$ is the weight of the self transitions and $\beta, \gamma, \delta \in [0, 1]$ determine the influence of the binary relations between the three different types of network elements (users, items and tags). Because of the normalization step the rows of $\mathbf{A}$ sum to 1 so they can be used as transition probabilities.

According to the desired task from our taxonomy (Figure 4.1), the random walk is initiated by assigning the starting positions in the initial state vector $\mathbf{v}_0$. For all level 1 tasks (T_{1-3}), the index corresponding to the target user $u_k \in U$ is set to one: $\mathbf{v}_0(u_k) = 1$. When only the target user is used as starting point, the state vector $\mathbf{v}_n$ estimates the probabilities that users, items and tags are relevant to that user. Depending on the number of steps in the random walk (n) the probability estimate is mostly influenced by the user's personal preferences or the global popularity ($\mathbf{v}_\infty$).

For level 2 tasks (T_{4-12}), the initial state vector is initialized with two values: $\mathbf{v}_0(u_k) = 1 - \theta$ and $\mathbf{v}_0(x) = \theta$, where x indicates the user, item or tag, selected at level 1. (E.g. $t_m \in T$, represented in Figure 4.2.) The parameter θ ($\theta \in [0, 1]$) determines the influence of the personal preferences. When θ is set to 0, the state probabilities only depend on the preferences of the target user, and will therefore result in the same prediction as level 1 tasks. When $\theta = 1$ the probabilities depend only on the selected query element, so the result will not be personalized for u_k. If $0 < \theta < 1$ the model derives the probabilities, based on both the target user and the query element.

The final ranking is obtained by ordering the elements according to their state probability in $\mathbf{v}_n$. Depending on the task, the part of $\mathbf{v}_n$ that corresponds to either user, item or tag nodes should be sorted. This ranking will also contain the training data (i.e. the annotations created by the target user himself). Depending on the task, these training nodes should be removed from the final ranking. A content rec-

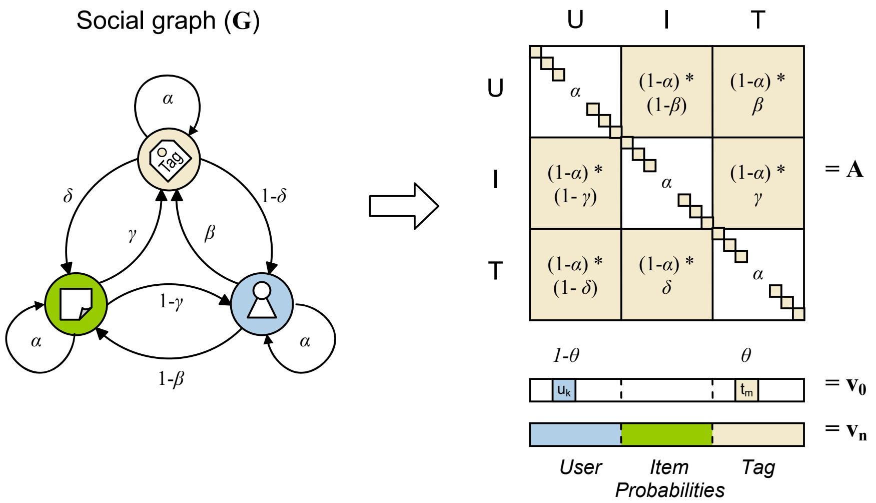


Figure 4.2: A social content system is represented as a tripartite graph, containing users, items and tags as nodes. The weight of the edges between these entities is determined by the annotations created by the network users. We add self transitions to allow the random walk to stay in the same node with a certain probability. Together, these edges constitute transition matrix $\mathbf{A}$. In the initial state vector $\mathbf{v}_0$, one or more starting nodes can be assigned according to the task from Figure 4.1. The result of the walk $\mathbf{v}_n$ contains the relevance probabilities of all three network elements. The model parameter α is used to tune the influence of self transitions. The weight of the edges between the three node types is controlled by β, γ and δ. For level 2 tasks, θ sets the personalization weight.

ommendation (T_1) for example, should only contain items that the target user has not seen before. However, when tags are suggested to annotate newly found content (T_5), previously used tags should also be recommended instead of only suggesting new tags.

The random walk directly models the social browsing behavior of network users. It estimates the probability that a user will reach a certain node after a few clicks in the network. The edge weight parameters model the probability of navigating from one entity type to another and the self transitions represent a user preferring the current results over another click. Therefore, the assumption is that by using this model, users will need fewer clicks to find the network elements that are relevant to their information needs.

4.2.1 Self Transition (α) and Walk Length (n)

Depending on the number of steps in the random walk (n), the final ranking is mostly influenced by the starting points (target user and query) or the background distribution ($\mathbf{v}_\infty$). The influence of the background after a certain number of steps is determined by the self-transition probability α. A large self transition probability allows the walk to stay in place (by taking many *self steps*), reinforcing the importance of the starting point, where a small value of α results in a walk that quickly converges

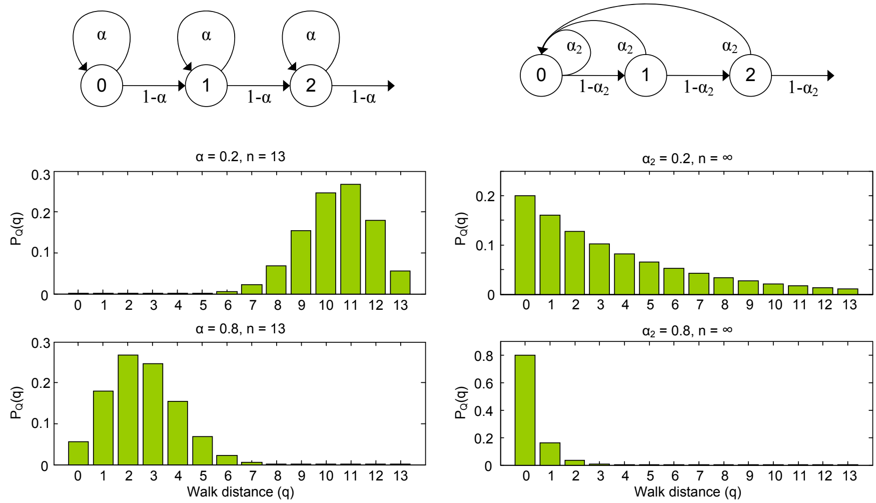


Figure 4.3: The PMF of the walk distance after a fixed number of steps through the social graph, for $\alpha = 0.2$ and $\alpha = 0.8$. The left distributions are based on a random walk with self transitions, where we set $n = 13$ as an example. The distributions on the right arise from a random walk with back teleport which is commonly applied with a fully converged state vector ($n = \infty$). Note that the walk distance does not equal the shortest path distance from the starting node.

to the stable background distribution.

Figure 4.3 shows the walk distance (number of non-self steps) for $\alpha = 0.2$ and $\alpha = 0.8$ at $n = 13$. Because all nodes have the same self transition probability, the walk distance (Q) after n steps through the social graph is a binomial random variable with the probability mass function (PMF):

$$P_Q(q) = \begin{cases} \binom{n}{q}\alpha^q(1-\alpha)^{n-q} & q = 0, \ldots, n, \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

where $P_Q(q)$ is the probability of being at distance q.

The PMF shows that if a large value is chosen for α, most of the probability mass will stay close to the starting point and a long tail is created toward more distant nodes. With a self transition probability < 0.5 most of the probability mass will move to the next state with each step, creating a long tail on the left side of the distribution.

We will compare this model to the more commonly used random walk with back teleportation, which allows the surfer to return to the starting node with a certain probability (Right model in Figure 4.3). The back teleport is used in personalized pagerank [99] and has shown to be a competitive model for network edge prediction (*rooted pagerank* in [76]) and can be used for recommendations and tag suggestions (part of the *FolkRank* method in [58; 63]). In this model, with teleport probability α_2, the probability distribution of the walk distance converges to $P_Q(q) = \alpha_2 * (1 - \alpha_2)^q$ when $n \to \infty$. This distribution always assigns the highest value to the nodes closest to the starting position, while the self transition model allows more distant nodes to be more relevant (for increasing n). Because the entities directly connected to the user (network distance equal to one) are already known to him we believe that the self-transition model is more appropriate to find *new* relevant content.

4.2.2 Edge Weights (β,γ,δ)

Three parameters (β,γ,δ) are used to tune the edge weights in the graph. For each node type the probability of making a transition to the other two node types is based on any of these parameters, e.g.: $P(S_{n+1} \in I|S_n \in U) = 1 - \beta$ and $P(S_{n+1} \in T|S_n \in U) = \beta$. We will not strive to optimize all these parameters for all tasks, but only show the effect of the relevant parameters given a certain task. With this approach, we evaluate the influence of the different types of annotations on the retrieval tasks.

4.2.3 Query Weight (θ)

Most tag-based retrieval systems use the selected tag as query term and rank the content according to popularity or freshness. Experience from the field of information retrieval has shown that a single term alone, absent other information like the user's interests or current context, is often not semantically expressive enough to clearly define the user's information need [120]. Also other retrieval tasks might benefit from extra information besides the query element.

In our model, we consider the selection of a tag, item or user as an indication of the user's *context*. In these level 2 tasks we will enrich the original ranking (based on the user) by integrating this context information. In the initial state vector, both the target user and the query element (either user, item or tag) are assigned a value according to θ, so the random walk will have 2 starting points. The weight of θ determines the amount of context adaptation.

This model can easily be extended by allowing more selected elements as input query, which corresponds to adding extra levels (below level 2) in our task taxonomy. When the user more clearly specifies his context, by selecting more entities, the model can incorporate this information in the initial state vector and derive a more context aware ranking. Many websites show the user's *breadcrumb trail*[1] at the top of the user interface, indicating the route a user has taken to the current view by clicking the links on the website. By adapting the initial state vector, the random walk model can use the entire breadcrumb trail created by the user while browsing through user, item and tag links. Previous work has shown that when the user selects enough query terms the information need will be sufficiently clear so that personalization will have no more influence on the ranking [25].

[1] http://en.wikipedia.org/wiki/Breadcrumb_navigation

4.3 Data

4.3.1 Data Characteristics

Tagging In currently popular social content systems, there is a clear distinction between *collaborative* tagging systems (e.g. CiteULike[2] and Del.icio.us[3]) and *individual* tagging systems (e.g. YouTube[4] and Flickr[5]). Many systems that allow user-generated content injection are individual tagging systems (IT) where only the injector of the content is able to assign the tags. In these systems, which are also known as *narrow folksonomies* [132] many people (who do not contribute but only consume content) will not build up a profile of the tags they prefer. In collaborative tagging (CT), every user can tag any piece of content [44]. In this way, users implicitly indicate which aspects of the content correspond to their personal interest. Also, in CT systems the aggregated tags of the network users create a relevance distribution for each content element, resulting in a *broad folksonomy* [132]. Furnas et al. already stated in 1987 that people often choose different terms to annotate content, resulting in low precision retrieval [40]. They argued that a theoretically optimal system would allow *unlimited aliasing* to describe the content (containing an infinite amount of annotations). We advocate that collaborative tagging approaches unlimited aliasing and is therefore a solution to enable effective personalized content retrieval.

Another option in tagging design is to remove the relation between users and their tags. In the well-known movie database IMDb[6], registered users can add or remove keywords for movies. They collaboratively build on a single collection of tags for each available movie. These keywords are however different from collaborative tags, because the relation between individual users and tags is not stored (or at least not visible from the outside). We will refer to this type of tagging system as *anonymous tagging* (AT). The study of anonymous tagging is interesting because it indicates how well the tag description created by the users of one system could be used by another community. Also, anonymous tagging could be compared to the use of key-words extracted from the content's metadata.

Marlow et al. defined two types of tag storage methods; A *set-model* maintains a single list of tags for each content element, while a *bag-model* stores the aggregated tags of all users [83]. AT$_{bag}$ can be derived from CT by storing only the item-tag relations and AT$_{set}$ can be derived from AT$_{bag}$ by binarizing the item-tag relations.

If users, items and tags are seen as separate entities, the act of tagging creates a ternary relation among them [86]. These relations can be stored in a 3D matrix $\mathbf{D}(u_k, i_l, t_m)$, where each position indicates if user $u_k \in U$ (with $k = \{1, \ldots, K\}$) tagged item $i_l \in I$ (with $l = \{1, \ldots, L\}$) with tag $t_m \in T$ (with $m = \{1, \ldots, M\}$).

Even collaborative tagging systems are usually very sparse. Different methods have been proposed to efficiently work with this data. Symeonidis et al. used a Higher Order Singular Value Decomposition (HOSVD) technique to reduce the dimensional-

[2] http://www.citeulike.org

[3] http://del.icio.us

[4] http://www.youtube.com

[5] http://www.flickr.com

[6] http://www.imdb.com/

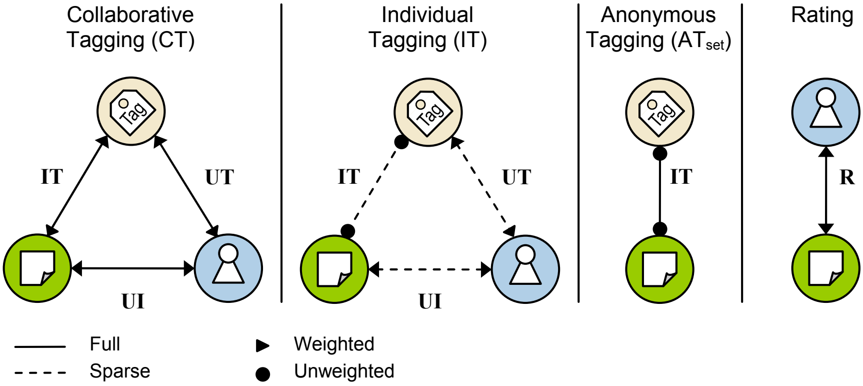


Figure 4.4: Common system designs with respect to content annotation. Collaborative tagging (CT) creates the richest annotation network, Individual tagging (IT) results in a very sparse graph, Anonymous tagging (AT$_{set}$) discards the user specific information and Rating (R) allows the user to give explicit preference indications. Anonymous tagging with bag storage (AT$_{bag}$) is similar to AT$_{set}$, but with weighted Item-Tag relations.

ity of the data [126]. Similar to Mika [86] and many other related works, we propose to sum over the 3 dimensions of $\mathbf{D}$ to obtain (see also Figure 4.4):

UT matrix: $\mathbf{UT}(u_k, t_m) = \sum_{l=1}^{l=L} \mathbf{D}(u_k, i_l, t_m)$, indicating how many items each user (u_k) tagged with which tag (t_m). In AT systems this matrix is not stored.

IT matrix: $\mathbf{IT}(i_l, t_m) = \sum_{k=1}^{k=K} \mathbf{D}(u_k, i_l, t_m)$, indicating how many users tagged each item (i_l) with which tag (t_m). In IT/AT$_{set}$ systems, this will be a binary matrix.

UI matrix: $\mathbf{UI}(u_k, i_l) = \sum_{m=1}^{m=M} \mathbf{D}(u_k, i_l, t_m)$, indicating how many tags each user (u_k) assigned to each item (i_l). In AT systems this matrix is not stored.

The impact of the independence assumptions made while flattening the $\mathbf{D}$ matrix is discussed in Section 4.8.

The three tagging matrices contain information about the relevance between pairs of nodes in the social graph. To reduce the impact of very popular elements, the matrices are normalized using TF-IDF weighting [110]. For example, the weighted User-Tag matrix is computed by:

$$\mathbf{UT}_{\text{TF-IDF}}(u_k, t_m) = \mathbf{UT}(u_k, t_m) * \log\left(\frac{K}{\sum_{k=1}^{k=K} \operatorname{sgn}(\mathbf{UT}(u_k, t_m))}\right) \tag{4.3}$$

where the sign function (sgn) sets all values > 0 to 1.

The weighted matrices together give the set of edges in the social graph: $E = \{\langle u_k, i_l\rangle | \mathbf{UI}(u_k, i_l) > 0\} \cup \{\langle i_l, t_m\rangle | \mathbf{IT}(i_l, t_m) > 0\} \cup \{\langle u_k, t_m\rangle | \mathbf{UT}(u_k, t_m) > 0\}$

Rating Besides tagging, the social aspects of networks stimulate people to share their opinion about the provided content. In many interfaces people can assess the quality of the content by giving a rating. Work on collaborative filtering systems has shown that these ratings can effectively be deployed to predict a user's interest and make predictions about his future behavior [[12]; [53]; Wang, J. et al. [133]].

Earlier work on collaborative tagging systems proposed to create the social graph from the three projections of the ternary user-item-tag relation [86; 58]. Although the $\mathbf{UI}$ matrix contains interesting information about the users' tagging behavior, the relation between the number of tags assigned to an item and the preference of the user toward that item is unclear. Therefore, when modeling the users' preference, we propose to replace the tag based User-Item matrix by the matrix based on the users' ratings. We will discuss the effect of this choice on the personalized search task in Section 4.5.2.

The rating matrix ($\mathbf{R}(u_k, i_l)$) contains the users' preference for the available content, often expressed on a five or ten point scale. Previous work has shown that even low ratings can be used to retrieve relevant content [24]. Based on the observations in this work we will directly use the value of the ratings as the weight of the User-Item edges $E = \{\langle u_k, i_l\rangle | \mathbf{R}(u_k, i_l) > 0\}$. The transition probability in the random walk is based on the relative differences between a user's ratings (and tag assignments), therefore there is no need to normalize the rating profiles before they can be compared (unlike the usual approach in collaborative filtering models [12]).

4.3.2 Datasets

LibraryThing (LT) LibraryThing[7] is an online web service that allows users to create a catalog of the books they own or have read. A user can tag and rate all the books he adds to his personal library. The social aspects of this network give the users the opportunity to meet like-minded people and find new books that match their preference. The popularity of the system has resulted in a database that contains over 3 million unique works, collaboratively added by more than 400,000 users. We are not aware of any other open network of this size where both collaborative tags (≈ 40 million) and ratings (≈ 5 million) are actively used.

We have collected a trace from the LibraryThing network, containing 25,295 actively tagging users[8]. The most popular tags represented in a large cloud (> 1000 tags) on LibraryThing[9] are used as seeds of the crawl. For each of these tags we get the list of people who have used the tag more than once and for all these users we download their entire book list. To get the users who have clearly expressed their preference we filter the data and retain 7279 users who have all supplied both ratings and tags to at least 20 books. We remove books and tags that occur in fewer than 5 user profiles, resulting in 37,232 unique works and 10,559 unique tags. This pruned dataset contains 2,056,487 UIT (User-Item-Tag) relations, resulting in a density of $7.2 * 10^{-7}$ (fraction of non empty cells in $\mathbf{D}$). The derived $\mathbf{R}$, $\mathbf{UT}$ and $\mathbf{IT}$ matrices have a density of respectively: $2.8 \cdot 10^{-3}$, $5.2 \cdot 10^{-3}$ and $2.0 \cdot 10^{-3}$. This pruning step is a fair choice, as it represents the typical recommender system deployment; these systems do not give recommendations to users who have provided insufficient preference information. Movielens, for example, asks new users to rate at least 15

[7] http://www.librarything.com

[8] Crawled in July 2007. Available from http://dmirlab.tudelft.nl/users/maarten-clements

[9] http://www.librarything.com/tagcloud.php

movies before they can start using the system[10]. Table 4.2 summarizes the statistics of the datasets. In this table 'Annotations' refers to the total number of applied tags or ratings, while 'Posts' indicates the number of User-Item relations. For tags there can be more annotations than posts as a user can choose to assign multiple tags when posting a single resource. The rows 'Users', 'Items' and 'Tags' give the total number of *unique* users, items and tags in the dataset.

As expected in data organized by human activity, we found that the number of books and annotations in the users' catalogs follow a power-law distribution [[7], [49]]. We expect that this data is comparable to collaboratively annotated movies, as books and movies comprise the same themes and storylines that can be categorized by tags.

The user interface of LibraryThing allows users to assign ratings on the scale from a half to five. Half ratings can be given by clicking a star twice. The distribution in Figure 4.5a shows that half ratings occur about 4 times less frequently than whole ratings. Figure 4.5b shows the relation between the rating and the number of tags given to an item. The upward trend shows that there is a slight correlation between these two variables. This graph also shows that books with half ratings tend to get more tags. This might indicate that the half ratings are used by people who put more effort into the categorization of their books. To substantiate this claim we define two user groups, the *active annotators* for whom at least 40% of the ratings are half ratings and the *lazy annotators* who have never used half ratings. If we plot the average number of tags per rating for these two groups we can clearly see that the people who use half ratings give more tags than the people who do not. There is even a slight bump at ratings 1/1.5 which indicates that the active annotators try to explain why they give a book a low rating.

LibraryThing also allows users to assign 'Friends' and connect to people with 'Interesting Libraries'. In our pruned dataset, 665 users have one or more 'Friends' and 532 users have indicated at least one 'Interesting Library'. Experiments in Section 4.5.4 validate relevant user prediction (Figure 4.1, T_3) using these relations.

Movielens (ML) Sen et al. discussed a tagging application that was implemented in the Movielens system and used for evaluation with a select user group [116]. In

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

Table 4.2: Data statistics

	LT		ML		MLR	IMDb	RSDC08
	Rating	CT	Rating	CT	Rating	AT$_{set}$	CT
Annotations	749401	2056487	15865	27887	100000	40150	709019
Posts	749401	749401	15865	15865	100000	-	214188
Users	7279	7279	443	443	943	-	2346
Items	37232	37232	2511	2511	1682	1682	186280
Tags	-	10559	-	2400	-	2479	56722
Density R	$2.8 \cdot 10^{-3}$	-	$1.4 \cdot 10^{-2}$	-	$6.3 \cdot 10^{-2}$	-	-
Density UI	-	$2.8 \cdot 10^{-3}$	-	$1.4 \cdot 10^{-2}$	-	-	$4.9 \cdot 10^{-4}$
Density UT	-	$5.2 \cdot 10^{-3}$	-	$1.1 \cdot 10^{-2}$	-	-	$1.1 \cdot 10^{-3}$
Density IT	-	$2.0 \cdot 10^{-3}$	-	$3.2 \cdot 10^{-3}$	-	$9.6 \cdot 10^{-3}$	$6.2 \cdot 10^{-5}$

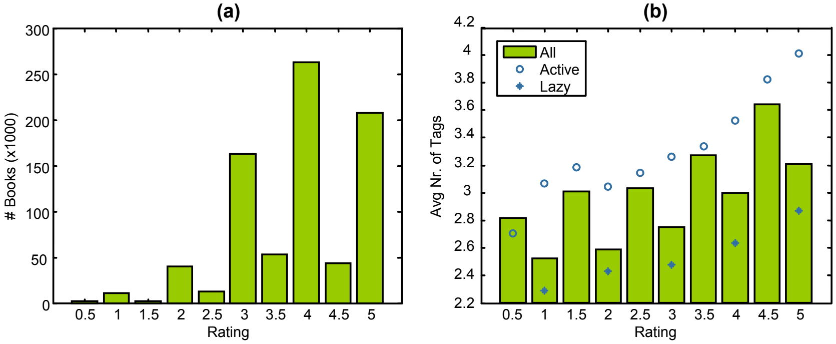


Figure 4.5: LibraryThing data statistics: a) The distribution of rating occurrences in the pruned dataset. b) The average number of tags assigned, given the rating for all users, only active annotators and only lazy annotators.

January 2009 the Grouplens research lab released this dataset with 10 million ratings and 100,000 tags for 10,681 movies by 71,567 users collected with the Movielens recommender system[11]. Many users have supplied a rating without giving tags. We only keep the annotations that contain both rating and tags and remove users with fewer than 5 items, and items/tags with fewer than 2 users. The resulting data statistics are summarized in Table 4.2.

Movielens Rating - IMDb Tagging (MLR-IMDb) Grouplens also provides a well known benchmark dataset for collaborative filtering algorithms, containing a collection of 100,000 user ratings. The data consists of 943 users who have all given at least 20 ratings to the collection of 1682 movies [52].

To study the effectiveness of anonymous tagging systems, we enrich the rating information with keywords extracted from the IMDb[12] database, using the urls included in the data descriptions of the Movielens set[13]. The combined **IT** matrix is pruned, so that all tags are used on at least 5 movies, resulting in a set of 2,479 unique tags. Because IMDb has an anonymous tagging system without tag aggregation (AT_{set}) we expect that the IMDb keywords will be less valuable for personalized retrieval tasks than the LT dataset.

Wang, X. et al. [135] showed that enriching the Movielens ratings with the movie titles can improve recommendation performance. A plausible explanation for this performance increase could be the presence of many sequels and series in the Movielens data (e.g. there are eight ‘Star Trek’ episodes in the Movielens data). In our experiments however (not shown), we found little evidence for the value of titles for recommendations when these series are ignored.

RSDC’08 To improve the state of the art of tag suggest systems, the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in

[11] http://www.grouplens.org/

[12] http://www.imdb.com/

[13] Crawled in December 2007. Available from http://dmirlab.tudelft.nl/users/maarten-clements

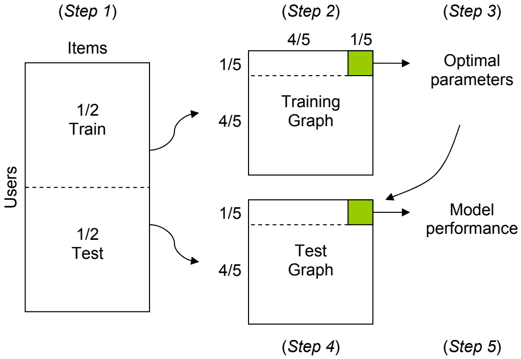


Figure 4.6: Splitting the data in a train and test set.

Databases (ECML PKDD) has organised the *RSDC'08 Discovery Challenge*[14]. For this challenge a dataset from the social bookmark and publication sharing system *BibSonomy*[15] was provided. BibSonomy allows users to annotate both bookmarks and publications with tags. The goal of the challenge was to learn a model that effectively predicts the tags a user will use to describe the content.

We have cleaned the data following the RSDC'08 Discovery Challenge's guidelines by removing system tags (like 'imported'), removing non-word characters and converting all words to lower-case. Because this data does not have any ratings we use the **UI** matrix to create the user-item edges. The resulting data statistics are included in Table 4.2. We will use this dataset to compare our findings on the tag suggestion task (T_5) with the results on LibraryThing data in Section 4.5.3.

4.4 Experimental Setup

4.4.1 Data Preparation

In order to estimate the performance of our model without overfitting to the data, we split the data in two equal parts (see Figure 4.6). Half of the users are put into the *training* set and the other half constitute the *test* set (*Step 1*), together with all the annotations they created (ratings and tags). We now use the training set to optimize the model parameters by holding out $1/5$ of the user's annotations of $1/5$ of the training users (the *validation* set, *Step 2*). We use our model to predict the held-out content and optimize the mean of the NDCG measure discussed in the next section (*Step 3*). For the smaller datasets (ML and MLR-IMDb) we use stratified cross validation, by repeating step 2 and 3 for five equally sized non-overlapping selections of validation items.

We then use the optimal model parameters to compute the performance on the test set, by holding out again $1/5$ of the user profiles of $1/5$ of the users, and computing the

[14] http://www.kde.cs.uni-kassel.de/ws/rsdc08/
[15] http://www.bibsonomy.org/

NDCG (*Step 4*). To evaluate the stability of the results we compute the performance on all 5 non-overlapping splits of validation users and show the variance in Section 4.6.

The validation items are selected randomly from the user profiles, as opposed to temporal selection. Non-temporal selection allows us to do cross-validation by selecting different sets of test items. The user-scenario matching this selection procedure is the recommendation of unseen relevant content instead of predicting future content.

In the rest of this chapter, all graphs will visualize parameter optimization on the training set, while tables give a comparison on the test set.

4.4.2 NDCG Evaluation

To evaluate the suitability of the predicted content ranking for any of the item ranking tasks (Figure 4.1, $T_{1,4,7,10}$) we use the Normalized Discounted Cumulative Gain (NDCG) proposed by Järvelin and Kekäläinen [62].

We first create a gain vector G with length L (all items) of zeros. In this gain vector, the predicted rank positions of the held-out validation items that correspond to a positive opinion $r \in \{3, 3.5, 4, 4.5, 5\}$ are assigned a value of respectively $G \in \{1, 2, 3, 4, 5\}$. This mapping is done because we do not want to predict content that has received a low rating and we want to reduce the impact of slightly relevant content on the evaluation. We do not normalize the rating profiles before assigning the gain, because we expect that the high offset in the ratings (See Figure 4.5a) is due to the fact that people tend to carefully select the content to view or read. As a result, people have read many more books they like than books they do not like, while normalization by mean rating would assume that people only like about half of the books they read.

In order to progressively reduce the gain of lower ranked test items, each value in the gain vector is discounted by the $\log_2$ of its index i (where we first add 1 to the index, to ensure discounting for all rank positions > 0). The Discounted Cumulative Gain (DCG) now accumulates the values of the discounted gain vector:

$$\text{DCG}[i] = \text{DCG}[i-1] + G[i]/\log_2(i+1) \tag{4.4}$$

The DCG vector is normalized by comparing it to the optimal DCG vector. This optimal DCG is computed using a gain vector where all test ratings are placed in the top of the vector in descending gain order. Component by component division now gives us the NDCG vector in which each position contains a value in the range $[0, 1]$ indicating the level of perfection of the ranking so far. We use the area below the NDCG curve as score to evaluate our rank prediction. In the experiment section we show the mean of the NDCG over all validation users.

4.5 Experiments

4.5.1 Content Recommendation

Task description We first look at the task of content recommendation (Figure 4.1, T_1), which is a well known task in the field of collaborative filtering. For this task we will use the LT data (CT+Ratings), the ML data (CT+Ratings) and the MLR-IMDb data (AT$_{set}$+Ratings). In the random walk model we implement this task by setting

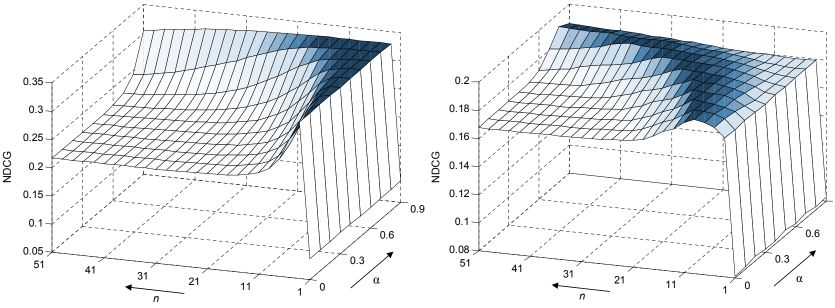


Figure 4.7: The NDCG for increasing self-transition probability and walk length for LT (left) and ML (right). The increase in self-transition probability procures a slower diffusion of the walk. On the Movielens data, a longer walk is needed to reach the optimal performance. Note that no even steps are shown as they would not produce a content ranking if $\alpha = 0$.

the initial state vector according to the target user, $\mathbf{v}_0(u_k) = 1$. After each number of steps n we rank the part of the state vector $\mathbf{v}_n$ that corresponds to the items, according to the state probabilities.

We will first describe the parameter optimization of the random walk model and then compare the test results for some selected settings. In Section 4.6 we will compare our results to the M-LSA model proposed by Wang, X. et al. [135] and the gradient descend SVD method from Funk [39].

Self transitions increase model robustness To set a baseline method and observe the effect of the self transitions, we walk randomly over the user-item graph of LT and ML. By setting $\gamma = 0$ (item-tag step) and $\beta = 0$ (user-tag step), no edges are created to the tag nodes, so the walk will only spread over the user and item nodes based on rating information. Figure 4.7 shows the mean NDCG for increasing walk length (n) and self transition probability (α). On the LT data the performance is optimal for 3 steps ($n = 3$), when the nearest unseen objects are found (see also Figure 4.8). The performance slowly drops when the ranking is more popularity based ($n \rightarrow \infty$). The results at $n = 3$ resemble a simplified version of traditional collaborative filtering where the content ranking is based on the ratings of the users with most similar content (Figure 4.8, path U-I-U-I). We use this method as a baseline to compare with the tag-included model.

On the ML data, the optimal performance is reached after a longer walk through the graph (Figure 4.7, right). A random walk has been shown to have a soft clustering effect that relates similar concepts before converging to the background probability [127; 29]. A longer walk therefore allows users with small profiles (few items and tags) to find the cluster of content that matches their preference, while for users with many items and tags in their preference profile, all relevant content can be found with very few steps. In ML the average user has a much shorter preference profile than in LT (as we use a less strict pruning policy, requiring only 5 items per profile for ML as opposed to 20 for LT). Therefore, the need for clustering is more prominent on

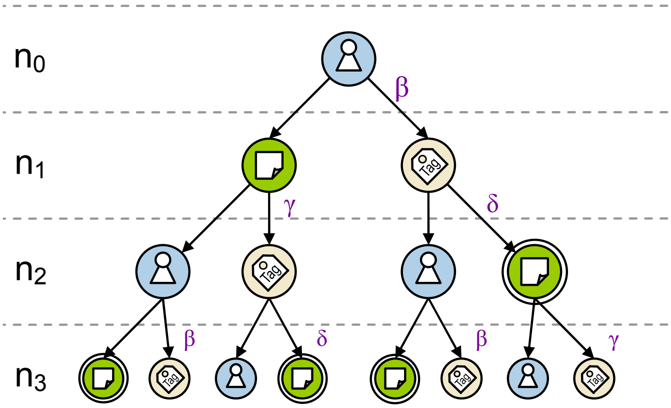


Figure 4.8: The first steps in the social graph starting from a user node. The first candidate items for recommendation are found 2 or 3 steps away from the user: Path U-I-U-I: Items of users who have rated the same content as me; Path U-I-T-I: Items that match the tags assigned to my items by the community; Path U-T-U-I: Items of users who have used the same tags as me; Path U-T-I: Items that match the tags I have previously used.

the ML dataset. When we repeat the parameter optimization in LT for only the users with a small profile (fewer than 50 annotations), we observe similar results and see that a longer walk is needed to reach the optimal prediction for these users (Data not shown).

We can see that the self transition probability does not influence the value of the optimal NDCG on the user-item graph. Only on the ML data the optimal NDCG slightly drops for very small values of α. A high self transition is however useful to increase the robustness of the model, because it makes the model less dependent on the walk length parameter. The slower diffusion of the walk assures that all nodes are reached (and therefore a complete ranking of all network elements can be made), while most of the probability mass remains close to the starting nodes (see Figure 4.3). In the following experiments we fix α at 0.8 for both datasets to create a slow diffusion of the random walk and reinforce the importance of the initial state.

Personal tags are only useful in coherent folksonomies Figure 4.8 shows the first three steps in the graph when only the target user is used as the starting point (level 1 tasks). The optimization of β (user-tag step) on the LT dataset (see Figure 4.9) indicates that a user's personally created tags do not contribute to the content ranking. This can be explained because the tags the target user would prefer for the validation items are not necessarily assigned by other people, so the path U-T-I might not exist yet. The tags that other people assigned to your training items appear to be the more predictive tags to retrieve the validation items (path U-I-T-I).

Running the same experiment on the ML data shows that the personal tags are much more useful in this dataset. The optimal NDCG is found at $\beta = 0.2$, see Figure 4.10. This can be explained by the fact that ML operates a tag suggestion system, which makes users select tags that are also in use by other users. Consequently, these

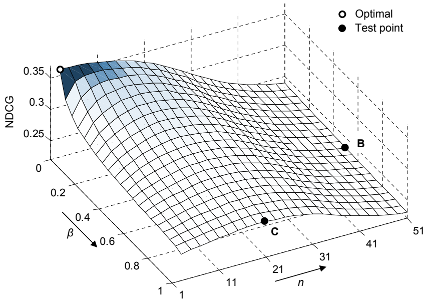


Figure 4.9: Optimizing β on the training set of LT ($\gamma = 0.4$, $\delta = 0.5$). We show the mean NDCG for odd steps starting from $n = 3$.

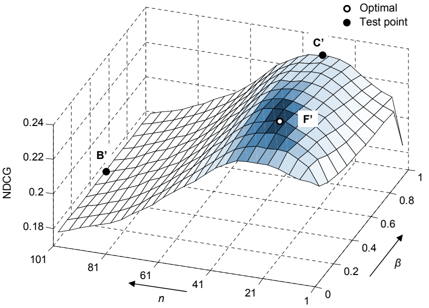


Figure 4.10: Optimizing β on the training set of ML ($\gamma = 0.8$, $\delta = 0.7$). Optimum at $\beta = 0.3, n = 25$.

tags also appear useful to find new content annotated by these other users. We conclude that the observed difference between LT and ML demonstrates the benefit of tag suggestion, as it can produce a more coherent folksonomy, where people use the same terms to annotate the same content.

Ratings and tags both improve ranking To compare the performance of a ranking based only on ratings to a tag based ranking, we fix β at zero and look at variations in γ (item-tag step). Figures 4.11 and 4.12 show the NDCG results on the training set of LT and ML, respectively. The γ parameter balances the influence of users who rated similar content (path U-I-U) versus tags that describe the target user's content (path U-I-T). The results clearly show an optimum when both aspects are taken into account (point F).

To get a valid comparison of the different parameter settings, the performance of the optimal and test points in the graphs is computed on the test set. Table 4.3 gives the mean NDCG for the different parameter settings on both datasets. For completeness it also contains the NDCG for a randomized item list (A) and the global popularity (B, where we assume full convergence at $n = 51$ for LT and $n = 101$ for ML). The significance of the performance difference with standard collaborative filtering ($\gamma = \beta = 0$, point D) is presented by the p-value computed with a Wilcoxon signed rank test [139]. Hereby, we test the hypothesis that the differences between the paired NDCG scores of both models come from a distribution with a median of zero. Small p-values indicate that the underlying distributions are significantly different. To show the actual improvement we have also included the percentage of change with respect to the reference model. Notice in particular that the combined model (point F) is significantly better than standard collaborative filtering (point D).

On the LT dataset we have also computed the optimal result when no TF-IDF weighting is applied to the input matrices (Table 4.3). The performance improvement over our reference model based on ratings drops from $+8.8\%$ with TF-IDF weighting to only $+0.3\%$ without. This shows that the downweighting of highly connected nodes in the graph allows for the discovery of more query specific content.

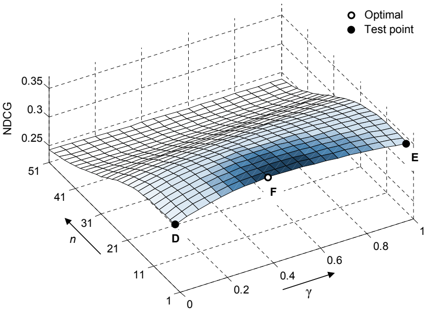


Figure 4.11: Optimizing γ on the training set of LT ($\beta = 0$, $\delta = 0.5$). Optimum at $\gamma = 0.4, n = 3$.

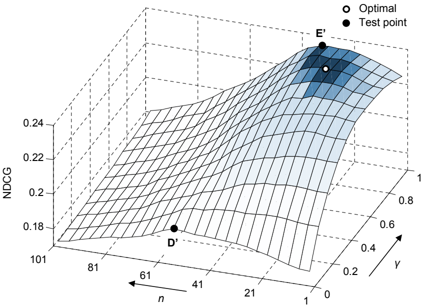


Figure 4.12: Optimizing γ on the training set of ML ($\beta = 0$, $\delta = 0.7$). Optimum at $\gamma = 0.8, n = 25$.

Tag-based ranking outperforms user-based ranking Table 4.3 shows that the ranking based on the tags applied by all users (point E, Figure 4.8 path U-I-T) outperforms the ranking based on similar users (point D, Figure 4.8 path U-I-U). On the ML dataset this difference is even more significant (Figure 4.12), which can again be explained by the more coherent folksonomy in Movielens.

Ramakrishnan et al. showed that recommender systems based on users' rating similarity can be abused by content advertisers and can reveal personal details for users with rare content interests [103]. Recommendations based on path U-I-T-I do not make use of other users' ratings and are therefore more robust against privacy attacks. The good results at parameter setting E show that a system that needs to guarantee the privacy of its users, can effectively use the aggregated tags (in CT or AT$_{bag}$ systems) to predict content recommendations without exposing other users' preferences.

Table 4.3: Content recommendation in collaborative tagging: test results

	Data	Model	β	γ	δ	n	NDCG	Diff.	p-value
A	LT	Random ranking	0.5	0.5	0.5	1	0.086	-73%	$< 1 * 10^{-15}$
B	LT	Global popularity	0.5	0.5	0.5	51	0.233	-27%	$< 1 * 10^{-15}$
C	LT	Personal tags (U-T)	1	0.4	0.5	21	0.271	-15%	$< 1 * 10^{-15}$
D	LT	Coll. filtering (U-I-U)	0	0	0.5	3	0.318	Ref	Ref
E	LT	Network tags (U-I-T)	0	1	0.5	3	0.326	+2.5%	$1.4 * 10^{-4}$
F	LT	Combined	0	0.4	0.5	3	0.346	+8.8%	$< 1 * 10^{-15}$
	LT	No TF*IDF	0	0.8	0.5	3	0.319	+0.3%	$3.8 * 10^{-2}$
A’	ML	Random ranking	0.5	0.5	0.5	1	0.082	-57%	$< 1 * 10^{-15}$
B’	ML	Global popularity	0.5	0.5	0.5	101	0.177	-7.1%	$4.4 * 10^{-6}$
C’	ML	Personal tags (U-T)	1	0.8	0.7	33	0.232	+21%	$2.4 * 10^{-10}$
D’	ML	Coll. filtering (U-I-U)	0	0	0.7	55	0.191	Ref	Ref
E’	ML	Network tags (U-I-T)	0	1	0.7	33	0.237	+24%	$1.0 * 10^{-12}$
F’	ML	Combined	0.3	0.8	0.7	25	0.243	+27%	$5.3 * 10^{-13}$

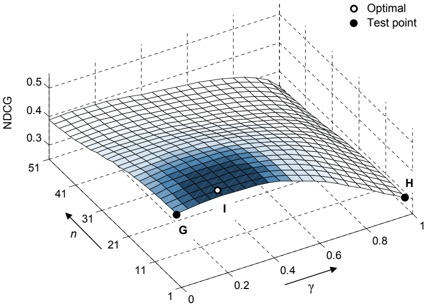


Figure 4.13: Optimizing γ for anonymous tagging (AT$_{set}$) in MLR-IMDb. The optimum on the training set is found at $\gamma = 0.2, n = 5$ (I).

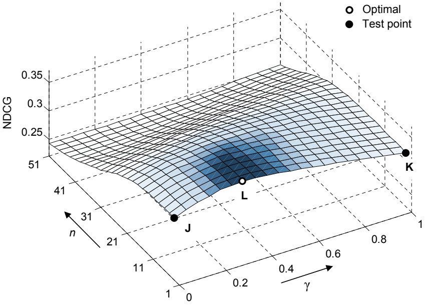


Figure 4.14: Optimizing γ for anonymous tagging (AT$_{set}$) in LT. The optimum on the training set is found at $\gamma = 0.3, n = 3$ (L).

Tag aggregation is essential To evaluate the effectiveness of an anonymous tagging system, we repeated the experiments on the combined MLR-IMDb data. Because this dataset does not contain the information about User-Tag relations ($\beta = 0$, $\delta = 1$), the first step through the graph will always be U-I. Therefore, we start with the optimization of γ, which determines the second step in the graph (See Figure 4.8). Figure 4.13 gives the NDCG for variations of n and γ. The shape of the plane is almost similar to the previous results on LT (Figure 4.11), with the exception that CF (Point G) now outperforms tag based ranking (Point H). To verify this result, we convert the LT data into anonymous tagging with *set-storage* by setting the **UT** matrix to zero and binarizing the values in the **IT** matrix, and we obtain Figure 4.14. We also observe the performance drop in the LT data when γ goes to 1. Results on the test set are shown in Table 4.4. The performance difference between test point K and E solely results from the binarization of the I-T edge. We conclude that the relevance distribution that arises from the aggregation of collaboratively contributed tags (in CT or AT$_{bag}$ systems) is essential for a retrieval system based on tagging information.

Conclusions The results on this task have shown that the exploitation of the graphical structure of the data can improve recommendations, especially for users with few annotations. The clustering effect of a medium length random walk allows for discovery of relevant content that is not directly linked to the current preferences of the user.

Table 4.4: Content recommendation in anonymous tagging: test results

	Data	Model	γ	n	NDCG	Diff.	p-value
G	MLR-IMDb	Coll. filtering	0	3	0.513	Ref	Ref
H	MLR-IMDb	Tag based	1	3	0.293	-43%	$< 1 * 10^{-15}$
I	MLR-IMDb	Combined	0.2	5	0.521	+1.6%	0.50
J	LT-AT$_{set}$	Coll. filtering	0	3	0.318	Ref	Ref
K	LT-AT$_{set}$	Tag based	1	3	0.302	-5.0%	0.05
L	LT-AT$_{set}$	Combined	0.3	3	0.345	+8.5%	$3.2 * 10^{-3}$

Deployment of tag suggestion in a tagging interface results in more useful tags with respect to content recommendation. The comparison between the LT and ML datasets shows that when users select tags that are also used by other people, the resulting preference profile can be used to find more related content.

Content ranking based on the aggregated tags of the items in a user's catalog ($\gamma = 1$, path U-I-T) can outperform a ranking based on similar users, which corresponds to (a simplified version of) collaborative filtering ($\gamma = 0$, path U-I-U). Ranking on aggregated tags is more robust to security issues like content promotion or discovery of private user information. To obtain the aggregated item-tag relation, a system needs to be designed as either CT or AT$_{bag}$. All datasets showed that the optimal recommendation should be based on a combination of rating and tagging information.

4.5.2 Personalized Search

Task description To initiate content retrieval, social tags are often shown in a *tag-cloud*, a visual depiction of tags in which the more popular tags are typeset in a larger font or more prominent color. Although many different methods exist to draw these clouds [67], the relevance of a tag is often based on the global popularity of the tags in the entire network (e.g. popular tags in Last.fm[16]). In this way of navigation only a single popular word is used as a query, resulting in many retrieved documents. In traditional information retrieval as well as web-search, people often use multiple word queries in order to disambiguate their information needs.

We see the selection of a tag as an indication of the user's current context. Because the user selects only a single term as query, the content ranking cannot be reliably based on this context alone. We need to find an optimal balance between the ranking based on the user's personal preference and the selected tag (Figure 4.1, T_7).

In the previous section we optimized the completely personal content ranking. Now we will first find the optimal edge weight settings for completely unpersonalized search and then optimize θ and n to combine the user and tag into a single query. We slightly alter the evaluation process for this task. For each user we separately compute a content ranking for each of the tags appearing in the validation set. We now try to optimize the ranking of the validation items that were originally annotated by *this* user with *this* tag. Here we make the assumption that the user would use the same terms for the annotation of his content as he would use to retrieve this content. For this task we will use the graph created by the ratings and collaborative tagging data of LT.

Tag based queries need multistep walks We will first optimize the edge weight parameters for unpersonalized content retrieval (by setting $\theta = 1$, and thus $\mathbf{v}_0(t_m) = 1$). Starting from a tag, the first step through the social graph depends on δ, where $\delta = 0$ implies a T-U step (and $\delta = 1$ a T-I step). Figure 4.15 shows the optimization of δ and n on the training set. We see that, similar to the results on T_1 (content recommendation), the connection between user and tag does not improve content ranking on this dataset (the optimum has $\delta = 1$).

[16] http://www.last.fm/tags

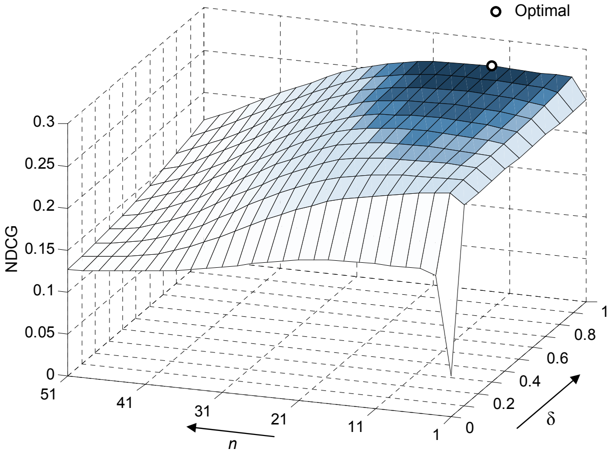


Figure 4.15: Optimization of the first step and walk length for tag-based search ($\theta = 1$), with fixed $\beta = 0.5$ and $\gamma = 0.5$. The optimal result on the training set of LT is found at $\delta = 1$ and $n = 13$.

We also find that the optimal number of steps is larger than one ($n = 13$), which means that the random walk improves effectiveness over a content ranking based on direct relations only. Since there are no spelling or language rules, tagging systems usually contain many synonymous terms, therefore content that has not been tagged extensively will often miss the terms used as a query by other people. The random walk can find these latent relations that are not explicitly present in the data.

Combining user and tag outperforms frequency ranking Both user based content ranking and tag based content ranking have been shown to perform optimally when the user-tag relation is not taken into account. We now combine the user and tag into one query balanced by θ. We set the edge weight parameters β (user-tag step) and δ (tag-item step) to the optimal values derived from previous results ($\beta = 0$ and $\delta = 1$, Figures 4.9 and 4.15). Surprisingly, this setting corresponds to using an aggregated anonymous tagging system (AT$_{bag}$). We fix γ (item-tag step) at 0.5, and focus on the optimization of the personalization strength θ. The results of Figure 4.16 then show that personalized retrieval gives a more accurate prediction than both completely personal and completely tag based queries (point R).

We define two baseline methods: *Random* (M): the mean NDCG for a random ranking, and *Global Popularity* (N): the NDCG at $n = 51$, with $\theta = 0.5$ (We assume that the state vector is fully converged, so that $\mathbf{v}_{51} \approx \mathbf{v}_\infty$ and it is independent of θ). We now compare four different model settings, derived from Figure 4.16 (see Table 4.5). *Frequency Search* (O): taking one step in our random walk model ($n = 1$) with $\theta = 1$ gives the ranking according to the number of times the tag was applied to the data. Because we have applied TF-IDF weighting on the **TI** matrix this setting corresponds to using a simple vector space model with TF-IDF weights. We see this parameter setting as the most standard implementation of tag based search in social content systems and therefore use it as the reference method. *Random Walk Search* (P): using

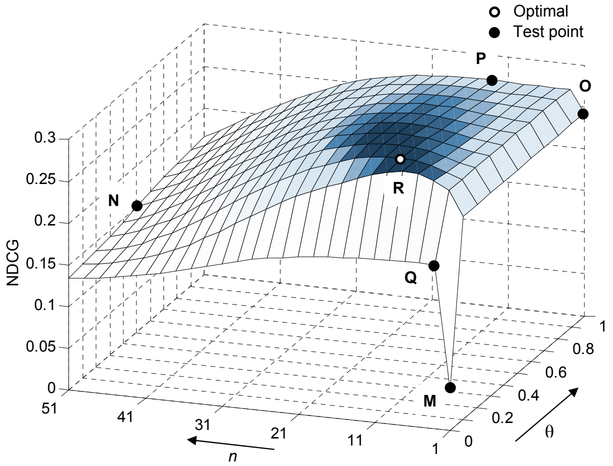


Figure 4.16: Optimization of the personalization influence θ and the walk length n on the LT training data ($\beta = 0$, $\gamma = 0.5$, $\delta = 1$). We show the mean NDCG for an odd number of steps from 1 to 51. The optimum is reached at $\theta = 0.2$ and $n = 11$ (R). Note that the NDCG at $n = 1$ and $\theta = 0$ (M) is equal to a random ranking, because users have no direct links to potentially interesting unseen content.

the optimal number of steps at $\theta = 1$ ($n = 13$) represents the optimal performance with our model without personalization. Compared to frequency search, this method integrates more indirectly related concepts. *Recommendation* (Q): when the model is completely personal ($\theta = 0$) the ranking will not depend on the tag. Obviously this model setting gives lower performance, we see however that the performance is higher than the popularity ranking, indicating a strong coherence within the users' libraries. *Personalized* (R): the optimal parameter setting of our model ($\theta = 0.2, n = 11$).

The results show that the combination of personalization and smoothing with indirectly related concepts (by increasing n) improves significantly over traditional frequency based retrieval. Our *personalized search* model outperforms *frequency search* by 19% and the random walk model without personalization (*Random walk search*)

Table 4.5: Personalized search: Results on the test set of LT. Popularity search is taken as reference. Using a Wilcoxon signed rank test, all results are significantly different from the reference point with a p-value $< 1 * 10^{-15}$.

	Data	Model	θ	n	NDCG	Diff.
M	LT	Random	0	1	0.048	-80%
N	LT	Global Popularity	0.5	51	0.148	-39%
O	LT	Frequency Search	1	1	0.241	Ref
P	LT	RW Search	1	13	0.268	+11%
Q	LT	Recommendation	0	3	0.188	-22%
R	LT	Personalized	0.2	11	0.286	+19%

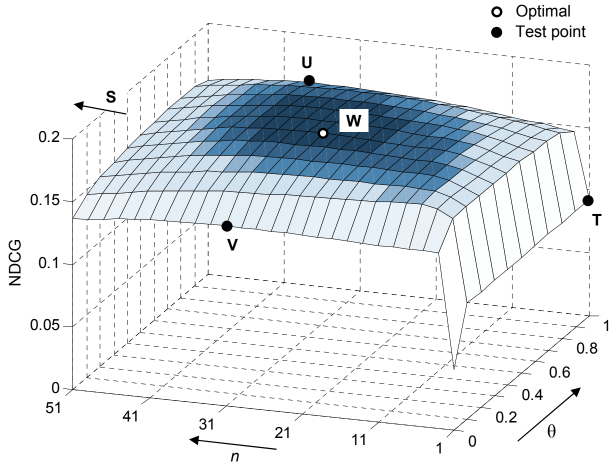


Figure 4.17: Optimization of the personalization influence θ in an individual tagging system created from the training set of LT. Because the random walk slowly converges in this sparse graph, we take the point of full convergence (point S) at $n = 101$. The optimum is reached at $\theta = 0.5$ and $n = 27$ (W)

gives a gain of 11%. If we compare the personalized model (R) to the random walk search (P), the integration of the user's history gives an additional improvement of 7%.

Individual tagging We adapt our data by removing all collaborative tags to evaluate the benefit of a collaboratively annotated collection over individual tagging. We construct the individual tagging graph by randomly selecting a single user per book and using only this user's annotations as edges (this user is assumed the uploader). We use the tags that would be assigned by the other readers as their queries to retrieve the held-out content. We are aware that we probably amplify the sparseness in individual tagging systems, because a user who is aware that he is the only annotator of the content might put more effort in his annotation. Because most existing IT systems allow all users to supply a rating to the data, we still construct the User-Item relation based on $\mathbf{R}$.

Sparse graph needs longer walk The results on the training set are shown in Figure 4.17. The most important observation is that a longer walk is needed to reach the optimal performance. This can be explained by the fact that the reduced number of edges makes it harder to reach a large amount of relevant content in a small number of steps.

IT needs integration of latent tag relations We show the results on the test set in Table 4.6. We observe that the absolute performance is almost twice as low as the results on the CT system, indicating that individual tagging systems provide less effective access to the provided content.

The NDCG gain of the personalized model (W) over the non-personal model (U:

Random walk search) is similar to the previously discussed results on collaboratively tagged data ($0.159 \rightarrow 0.171 = +7.3\%$). Compared to results on collaborative tagging, the random walk on the individually tagged graph shows much more improvement over *frequency search* (73% relative improvement). Even the global popularity (S) gives a more relevant ranking than frequency search (T). This can be explained by the experimental setup, taking other people's tags for the same content as queries; in practice, people choose different terms to describe the same content. Also, individual users consequently select their own favorite terms, especially when language differences are concerned [22]. Many synonyms occur in the content annotations but each user is only connected to few of these terms. The random walk smoothly integrates related concepts when a user uses his favorite term as query.

To get an impression of the difference in annotations between users, we have computed the average overlap of the tags that users have assigned to the same item. We find that the users' annotations on average have only 0.59 tags in common, which is only 0.23% of the tags they assign. This corresponds precisely to the variability in word choice observed by Furnas et al. [40]. Because users associate different terms with specific content, the retrieval model should take latent semantic relations into account, especially in individual tagging systems.

Preference indications are essential in individual tagging systems Because much related work has used the social graph based on tagging information only [58; 72; 86], we have also optimized our model on the graph created with the **UI** matrix instead of the **R** matrix. The **UI** matrix contains the number of tags a user has assigned to a certain item, which is a less explicit preference indication compared to ratings.

We find that the performance in the collaborative tagging graph without ratings is only slightly lower than the results obtained with rating based edges. With optimal parameter settings the NDCG drops from 0.286 (point P, Figure 4.16) to 0.285 (data not shown). This was already indicated by the correlation we found between the rating and the number of assigned tags (Figure 4.5b). We do however expect that in a dataset with more negative opinions, the integration of the explicit preference information might give larger performance gain over tag-based user-item relations, because it is impossible to assign a negative amount of tags.

If we remove the rating information from the individual tagging experiments and create the social graph with the tag-based **UI** matrix, we observe a much more significant performance drop from 0.171 (Point W, Figure 4.17) to 0.109. This is obvious

Table 4.6: Individual tagging: Results on the test set of LT. Using a Wilcoxon signed rank test, all results are significantly different from the reference point with a p-value $< 1 * 10^{-15}$.

	Data	Model	θ	n	NDCG	Diff.
S	LT-IT	Global Popularity	0.5	101	0.115	+25%
T	LT-IT	Frequency Search	1	1	0.092	Ref
U	LT-IT	RW Search	1	37	0.159	+73%
V	LT-IT	Recommendation	0	31	0.140	+52%
W	LT-IT	Personalized	0.5	27	0.171	+85%

because many User-Item relations are absent from the **UI** matrix. In individual tagging systems, people can only tag the content contributed by themselves, therefore the tagging data will not contain any preference indications about the other content in the network. The rating possibility allows people to create direct links with all content, instead of just the injected content.

Conclusions In this section we have found that single term queries can greatly benefit from personalization, because the user's annotations give an indication of his interests. The integration of the user in the query can disambiguate queries with multiple denotations or specialize queries with a broad semantic meaning.

For *personalized search* it is essential that a system allows users to create direct links to their preferred content, either by tagging or rating it. If all users can tag the content they like (as in CT systems) a weighted preference indication in the form of a rating does not contribute much to the retrieval performance. Individual tagging systems are overall less effectively accessible for retrieval tasks as the content is poorly annotated. To enable personalized retrieval in individual tagging systems ratings should be used to create user-item relations.

Compared to the user-based content ranking (T_1), tag queries need a longer walk to reach the optimal performance. By increasing the walk length, the content ranking will not only depend on the query tag, but also integrate latently related concepts and synonymous terms. Especially in individual tagging systems we have shown that multiple steps through the network are needed to improve the content ranking, because of terminology differences between users.

4.5.3 Tag Suggestion

Task description Most users are insufficiently aware of tags in use by others and they do not want to spend much time on content annotation. Ideally, the social content system suggests tags from the common vocabulary that fit the users' intention or interest, while remaining consistent with other users (Figure 4.1, T_5). It has been claimed that the suggestion of tags when a user is asked to label certain content would lead to a more coherent *folksonomy* [44], which was confirmed by our results in Section 4.5.1.

In this section, we compare the parameter optimization on LibraryThing data to that using the dataset released with the *RSDC'08 Discovery Challenge 2008*. We see tag suggestions as a (personalized) recommendation of tags where the user has indicated his context by clicking on an item. Our model implements tag suggestion as a random walk with two starting points, where one walk starts at the target user and the other at the selected item. We create the initial state vector as explained in Section 4.2. The target user gets a weight of $\mathbf{v}_0(u_k) = 1 - \theta$ and the selected item $\mathbf{v}_0(i_l) = \theta$.

Tag suggestions should be based on user and context For this task we will only discuss the influence of the parameters on the predicted ranking; we will not compare the actual effectiveness with related methods. Therefore, we do not keep a separate test set and use the entire LibraryThing dataset (CT + Rating) for the parameter optimization (We skip step 1,4 and 5 in Figure 4.6). We remove 1/5 of the profiles of 1000 users as a validation set. In this validation set, the average user has 13 items

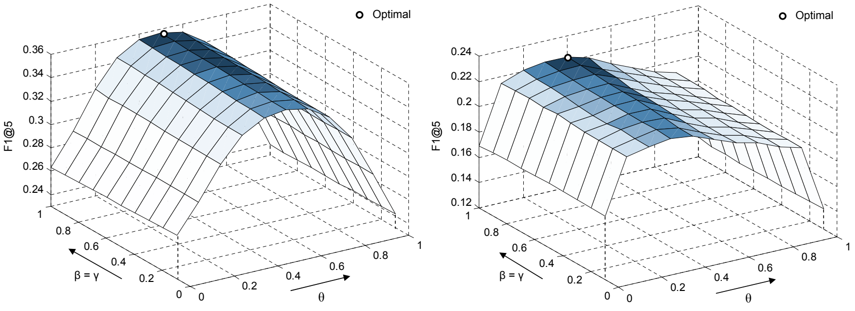


Figure 4.18: F1-measure at 5 of recommended tags on Librarything data (left) and RSDC08 data (right), both with $\alpha = 0.8$ and $n = 3$. The parameter $\theta \in [0, 1]$ determines the influence of the selected item. The weight of β and γ sets the use of tag vs. rating information. We find the optimum at $\beta = \gamma = 1$, and $\theta = 0.5$ for LT and $\theta = 0.4$ for RSDC08. The result at $\beta = \gamma = 0$ is not shown because this setting produces a random tag ranking.

(median), collaboratively annotated 54,665 times.

We now vary the contribution of the user walk vs. the item walk, by varying θ. Because tag assignments are binary relevance judgements, we report the F_1 measure at 5 instead of the NDCG used previously. The F_1 measure is defined as the harmonic mean of precision and recall:

$$\mathrm{F1} = \frac{2 * \mathrm{Recall} * \mathrm{Precision}}{\mathrm{Recall} + \mathrm{Precision}} \tag{4.5}$$

Figure 4.18 shows that the optimal tag suggestion combines the other users' description of the content with the target user's previous interests, with equal contribution ($\theta = 0.5$). Although we have shown that a user's personal tags are irrelevant for content retrieval, this indicates that the user-tag relations that exist in CT systems are essential for good tag suggestions. This corresponds to the user tests performed by Sen et al. who found that 51% of the tag applications are tags that the user has previously used [116].

We repeat the experiment in the same way on the RSDC08 data and see that the behaviour of the F_1 measure for the parameter variation is very similar (Figure 4.18 right). The main difference is a more abrupt performance drop when the ranking is based on the user or item alone ($\theta = 0/1$). The reason for this is that many users and items occur only once in the dataset, which produces a random ranking when that user/item is used as starting point of the random walk.

Rating information does not improve tag suggestions To evaluate the effect of rating vs. tagging information we set $\beta = \gamma$ (user-tag and item-tag step) and we optimize both parameters at the same time (Figure 4.18). We found the optimal results if only tagging relations are used to create the social graph ($\beta = \gamma = 1$), providing evidence that high ratings do not correspond to more descriptive tags. Users who like or dislike certain content are equally capable of describing the topic, therefore the paths I-U-T and U-I-T do not contribute anything to the tag ranking.

The optimal value of the walk length is found at $n = 3$ (data not shown), which means that the directly connected tags (at $n = 1$) can be slightly improved with indirectly related tags found at distance 2 and 3.

When we use the model with optimal parameters on the *test set* provided with the RSDC challenge, the tag prediction results drastically drop to 0.0274, using the evaluation script that was used for the challenge (which still corresponds to a fourth place in the 2008 challenge). Because the train and test set are split temporally, most of the users and items in the test set do not occur in the training set (and the task represents the situation of cold-start recommendation). One of the leading contributions to the challenge has argued that this does not produce a realistic scenario [80]. The results presented by the two best papers in the challenge show that a significant amount of data processing on the content titles and descriptions is necessary to account for these unseen users and items [80; 130].

Related work on tag suggestions Tag suggestions were extensively studied by Xu et al. who optimized the diversity of the recommendations and integrated the information about the tags already assigned by the user to the content [141]. This knowledge about already assigned tags would be implemented in our model by setting these tags as extra starting points of the walk.

Recent work from Song et al. shows that the textual information in documents can be used in a real-time tag suggest method when a spectral clustering method is used to limit the size of the item space [123]. Symeonidis et al. showed that tag suggestion can be improved with Higher Order Singular Value Decomposition (HOSVD), which enables the exploitation of latent semantic relations in multi dimensional sparse data. They evaluated this method on Bibsonomy and Last.fm data and showed that it is competitive with random walk based approaches [126].

Conclusions Adequate tag suggestions are a combination of the user's previously used tags and the tags assigned by other people. If both the user and the item have already built up a profile of previous annotations, both sources contribute almost equally in the prediction. Therefore the system needs to store both personal tags and the aggregated tags from all users, which is only done if the system is designed as *collaborative tagging*.

4.5.4 User Recommendation

Task description In our taxonomy we indicated that our model can also be used for the task of user recommendation (Figure 4.1, T_3). The LibraryThing website allows users to indicate which people are *friends* and who they consider to have an *interesting library*. We can use the prediction of *interesting libraries* as a means to evaluate our user prediction task. We assume that a user who qualifies someone else's library as 'interesting' will have a similar taste. It was shown by Schenkel et al. that search in social networks can be improved by exploiting friends with a similar tagging behaviour [114]. Here we evaluate if a user's taste is more clearly represented by his given ratings or tag assignments. We set the weight of β (user-tag step) and γ (item-tag step) equal to each other. These two parameters control the contribution of rating- versus tag-based edges to the random walk over the graph. We will keep δ

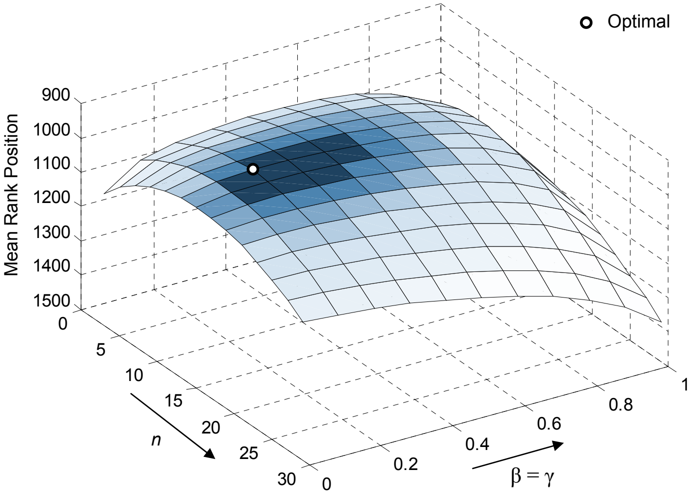


Figure 4.19: The mean rank position (MRP) of the people with 'interesting libraries' after n steps in our random walk model (with $\alpha = 0.8$, $\delta = 0.5$ and $\gamma = \beta$). Optimal at $\beta = \gamma = 0.2$ and $n = 13$.

(tag-item step) fixed at 0.5.

Using the random walk model we compute a personal user ranking for all people who assigned any *interesting libraries*. Because these interesting library assignments are independent from the tag and rating assignments, we use the entire UIT graph (7279 users) as training data. We initiate the walk with the target user as starting point ($\mathbf{v}_0(u_k) = 1$), and rank the user part of the resulting state probabilities ($\mathbf{v}_n(u_1, \ldots, u_K)$) in descending order. Because of the limited number of the social annotations (median of 2 *interesting libraries* out of 7278 users), the dataset is not suited to study actual effectiveness. Instead, we look at the mean rank position (MRP) of the validation set (people labeled as *interesting library* by u_k) in the predicted ranked list (see Figure 4.19). Although the performance is not good enough to actually suggest people with high precision, the parameter optimization gives us an interesting insight in the factors that determine a user's identity. The results on the training data are summarized in Table 4.7.

Ratings and tags both contribute We can see that an average length walk (around $n = 13$) gives the optimal results when β and γ are set to 0.2 (MRP = 954). This means that user prediction tasks benefit from both rating and tagging information. Looking at the extreme values of β, a ranking based on rating information alone ($\beta = \gamma = 0$, optimal at $n = 15$ with MRP = 973) outperforms the ranking based on tagging information alone ($\beta = 1$, optimal at $n = 13$ with MRP = 1,190). This again shows that the users in LibraryThing are more clearly represented by their ratings, but tagging information can give improvement.

Random walk improves user prediction The optimal MRP also outperforms a ranking based on directly connected users (MRP = 1,026 at $n = 2$) and a ranking

Table 4.7: User recommendation: Training results with independent validation data.

Model	β	γ	δ	n	MRP
Optimal	0.2	0.2	0.5	13	954
Rating based	0	0	0.5	15	973
Tag based	1	1	0.5	13	1190
Global popularity	0.2	0.2	0.5	51	1618
Direct connections	0.2	0.2	0.5	2	1026
Profile overlap	-	-	-	-	1485

based on the global network activity of the user (MRP = 1,618 when $n \to \infty$, for $\beta = \gamma = 0.2$). To find like-minded users, the LibraryThing website shows the people with a big catalog overlap, regardless of the rating people have given; this method results in an MRP of 1,485. In our random walk, the path between two users is weighted with the ratings the users have given. Besides, tags can find latent user relations because they can relate different books. Our method generalizes the user ranking as it predicts a relevance probability for all users, even if they have not read any identical books.

Conclusions We have seen that a combination of a walk over the user's content (via ratings) and his personal tags gives the best user recommendation. This indicates that besides similar content preference, people are attracted by other people with a common vocabulary.

Increasing the walk length can improve the ranking over a ranking based on directly related users. A longer walk does not only take the direct overlap between the users' catalogs into account, but also integrates the similarity between the books that are not shared by both users.

4.6 Model Performance Validity

We have used the random walk model to evaluate the differences between various ranking tasks in differently designed social content systems. Conclusions from these comparisons are only valid if the random walk model has any utility on this task. Therefore, we now compare the random walk results to those of three other recently proposed ranking methods. We will compare our results on item recommendation (T_1) to the M-LSA method proposed by Wang, X. et al. [135] and the SVD method described by Simon Funk [39], which has shown to be effective in the Netflix competition[17]. We use the personalized search task (T_7) to compare the random walk model with self-transitions to personalized PageRank which uses a back teleport probability [99].

SVD model description It has long been known that Singular-Value Decompositions (SVD) can effectively be used to overcome the sparsity problems in collaborative filtering [42]. The reasoning behind the use of an SVD can be explained as follows. If

[17] http://www.netflixprize.com/

the rating matrix is factorized in the form:

$$\mathbf{R} = U\Sigma V^T \tag{4.6}$$

where the columns of V form a set of orthonormal input basis vector directions for $\mathbf{R}$, the columns of U form a set of orthonormal output basis vector directions for $\mathbf{R}$ and Σ contains the singular values. The most prominent aspects of the data are represented by the basis vectors corresponding to the largest singular values in Σ. If we now only use a subset k of these vectors to reproduce the rating matrix ($\widehat{\mathbf{R}} = U_k\Sigma_k V_k^T$), the empty values are filled in with the most likely values based on the most prominent genres in the data.

Simon Funk proposed to estimate the SVD using a gradient descent algorithm which optimizes the prediction of the ratings by following the derivative of the prediction error [39]. The parameters in this model are: k, the final number of vectors in U and V; L, the learning rate of the gradient descend method and E, the number of training iterations over the entire dataset.

M-LSA model description One of the recently proposed methods for relevance ranking of multiple-type interrelated data objects is the M-LSA method of Wang, X. et al. [135]. Their model extends Latent Semantic Analysis (LSA) by integrating all pairwise relations between multiple types of objects. We compare our random walk model to the M-LSA approach on the content recommendation task, because their solution to a collaborative filtering problem uses an input matrix that is similar to our transition matrix ($\mathbf{A}$). Their matrix combines the submatrices ($\mathbf{R}$, $\mathbf{UT}$ and $\mathbf{IT}$) in the same way, but without the self transitions. Their model parameters are slightly different:

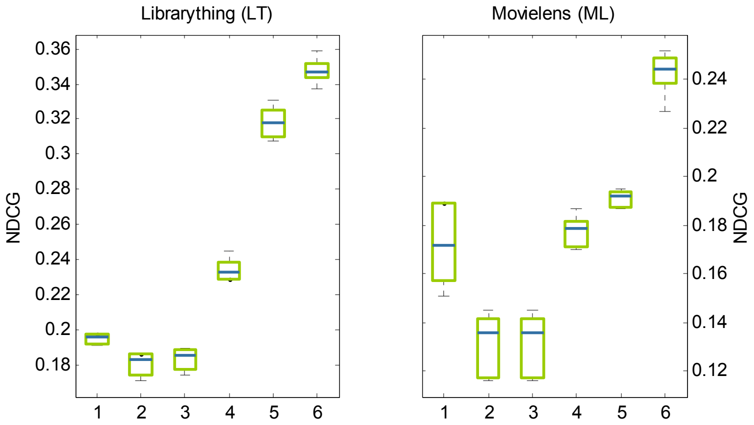
$$\mathbf{A}_{M-LSA} = \begin{bmatrix} 0 & \alpha\mathbf{R} & \beta\mathbf{UT} \\ \alpha\mathbf{R}^T & 0 & \gamma\mathbf{IT} \\ \beta\mathbf{UT}^T & \gamma\mathbf{IT}^T & 0 \end{bmatrix}$$

where $\alpha+\beta+\gamma = 1$. From this matrix, M-LSA computes the SVD and uses the k most prominent vectors to find latent semantic relations between users, items and tags. Using this low dimensional representation of the network, the top-N most similar users (N_u) to the target user (u) are retrieved (using Pearson correlation on the top part of the eigenvectors weighted by their eigenvalues). This subset of users is then used to predict the rating for the target user's validation items ($\hat{r}_{u,i}$) by computing a weighted average (just like standard memory based collaborative filtering [12]):

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v\in N_u}(r_{v,i} - \bar{r}_v)\cdot w_{u,v}}{\sum_{v\in N_u} w_{u,v}} \tag{4.7}$$

where $w_{v,u}$ is the Pearson correlation between the profiles of the target user and a user from his top-N, $r_{u,i}$ is a user's rating for item i and $\bar{r}_u$ is a user's average rating. The content ranking is derived by ranking the items according to the predicted ratings.

Random walk outperforms SVD-based methods Using the optimized parameters of all models (k,L,E for SVD; N,k,α,β,γ for M-LSA), we compute the NDCG on 5 folds of the test data and show the resulting box plots for both datasets in Figure 4.20. Because it is not trivial to include the tagging information in the SVD method we



1. SVD	: k = 7, L = 0.0005, E = 10
2. M-LSA - UI	: N=250, *k*=200
3. M-LSA - UIT	: N=300, *k*=200 *α*=0.6, *β*=0.1, *γ*=0.3
4. RW - GP	: Point B, Figure 9
5. RW - UI	: Point D, Figure 11
6. RW - UIT	: Point F, Figure 11

1. SVD	: k = 70, L = 0.0005, E = 10
2. M-LSA - UI	: N = 221
3. M-LSA - UIT	: N = 221
4. RW - GP	: Point B’, Figure 10
5. RW - UI	: Point D’, Figure 12
6. RW - UIT	: Point F’, Figure 10

Figure 4.20: Test results comparing three settings of the random walk (RW) to SVD and M-LSA with and without tag information. The parameters are set to the optimal values derived from the training data.

only report results on the rating matrix. We see that the random walk model clearly outperforms the other methods on both datasets. If we evaluate the rating prediction performance by computing the Mean Absolute Error (MAE), both SVD and M-LSA reach a performance around MAE = 0.7, which corresponds to previously published rating prediction results [52]. This confirms the observations in related work that rating prediction methods perform badly on ranking tasks [63; 81].

The optimal parameters of the SVD method indicate that the best ranking results are obtained when the vectors are not completely trained (both the learning rule L and the number of iterations E are much smaller than the values reported by Funk [39]). This means that the features corresponding to the users and items with most ratings in the training set are closer to their optimal value with respect to rating prediction. In this set-up the ranking resembles the global popularity ranking of the random walk model.

The M-LSA model could not find an optimal number of users in the ML dataset. The available number of similar users is simply too low to give reliable rating predictions. The presented ranking is therefore based on the mean rating of all users in the test set and does not differ for the model with or without tags. In addition, the M-LSA model only extends traditional user-based collaborative filtering by defining a different similarity function (with integrated tags). It is not trivial to apply this method to the more complicated level 2 tasks, making the random walk model more generally applicable.

Table 4.8: Comparison self transition vs. back teleport

Data	Model	α/α_2	θ	n	NDCG
LT-CT	Self transition	0.8	0.2	11	0.286
LT-CT	Back teleport	0.4	0.1	∞	0.282
LT-IT	Self transition	0.8	0.5	27	0.171
LT-IT	Back teleport	0.6	0.3	∞	0.164

Back teleportation In Section 4.2.1 we already explained the traditional random walk with back teleport based on PageRank (Figure 4.3) [99]. We have also optimized this model for the *personalized search* task (T_7), on the collaborative and individual tagging graphs. The back teleport model ranks the content according to the stationary state vector. We found that the state vector does not change significantly after 9 steps of the random walk and therefore use $\mathbf{v}_\infty \approx \mathbf{v}_9$.

The results for both models with optimized parameters are summarized in Table 4.8. Our experiments show that the back teleport model performs slightly (but insignificantly) less on the search task. We have chosen to use the self-transition model because it allows us to use the number of steps as a parameter to identify the differences between various retrieval tasks and tagging systems. In this way we can use one model to compare the frequency ranking with the smoothed ranking.

Conclusions Where SVD and neighbourhood methods work well on rating prediction, methods that exploit the graph structure (the random walk family of algorithms) do better on ranking tasks. The main difference between these two tasks is the fact that the evaluation of rating prediction does not take unrated items into account. The ranking evaluation explicitly judges the unrated items as less relevant than the rated items.

4.7 Related Work

4.7.1 Random Walk

The most widely known application of random walk models for information retrieval is the PageRank algorithm, first used in the Google search engine [99]. The main difference with our model is that the standard PageRank algorithm computes a fully converged state vector ($\mathbf{v}_\infty$), the principle eigenvector of $\mathbf{A}$, while our model uses the number of steps as a smoothing parameter. Our experimental results show that the relevance prediction using $\mathbf{v}_\infty$ is significantly less effective than when a short or average length walk is used (Section 4.5). Also, the difference between a ranking based on rating or tagging information seems to be absent when $n \to \infty$. This can be explained, because the general concepts (prevalent in the eigenvectors) are represented in both information sources.

In the first paper on PageRank, Page et al. also describe *personalized PageRank*, in which the random surfer jumps back to his starting page with a certain probability greater than zero [99]. The PageRank of pages close to a user's start site (or entire preference profile in the form of bookmarks) will then be higher. We have shown that the performance of the self-transition model is comparable to personalized PageRank on search in social content systems. The self-transitions however allow the walk to stay close to the query without the need to adapt the transition matrix for each retrieval task. Only the initial state vector has to be adapted to the target user and query terms.

The random walk with self transitions has first been used in the work of Craswell and Szummer [29]. They used a random walk on a query-image graph to retrieve more relevant images for each textual query. We have applied the model on the tripartite graph in which users, items and tags constitute the nodes. Because we directly integrate the network user in the model, the tasks that we describe are more focused on social interactions, which meets the desires of many current Internet users. Furthermore, in our model we always start the random walk from the target user, which makes all retrieval tasks personalized to each user's individual preferences.

Szummer and Jaakkola studied the soft clustering properties of a medium length random walk through a graph [127]. It was shown by Xi et al. that this clustering effect can be used to improve the similarity estimation between different typed entities that are connected by certain relations [140]. We have seen that depending on the data and task, increasing the walk length can improve the ranking in the social graph. The clustering effect allows the walk to find a group of semantically related entities before converging to the finite state probabilities.

Recommendation algorithms were described as a graph problem by Mirza et al. [87]. By allowing multiple steps over the user-item graph their algorithm finds latent relations between users and items. Fouss et al. used the Laplacian of the social user-item graph to compute the average commute time (ACT) of a random walk in order to provide collaborative recommendations [37]. The ACT replaces the walk-length parameter with a single distance value between all pairs of nodes. We chose not to use the average commute time in order to study the effect of the walk length, which gives an interesting insight in the distance from the query node to the most relevant elements and therefore the optimal amount of smoothing from the background probability.

4.7.2 Tagging Graph

A large part of the research on tagging systems has focused on the analysis of statistical patterns arising by the collaborative effort of network users. Golder and Huberman analyzed the structure of social bookmarking in Del.icio.us. They discovered recurring patterns of growth dynamics and identified various user tasks that result in different tagging behavior [44]. Halpin et al. extended this work by investigating the evolution of collaborative tagging patterns into stable distributions by computing the Kullback-Leibler divergence between different time points in Del.icio.us [49]. Marlow et al. used data from the popular photo catalog Flickr to show that individual tagging systems evolve differently over time [83]. Our results demonstrate that individual tagging also drastically reduces retrieval performance, which concurs with the *vocabulary problem* defined by Furnas et al.: people tend to use different terms to describe content [40].

Mika extended the common bipartite ontology model by directly integrating the network user in the graph [86]. The resulting tripartite graph gives more insight in the dynamics of social networks. Lambiotte and Ausloos used the projected matrices (**UT**, **IT** and **UI**) of the same graph to visualize the network structure of Audioscrobbler[18] and CiteULike [72].

Bao et al. used the finite state vector of the tripartite tagging graph for webpage ranking. They showed that this ranking outperforms normal PageRank, because it is based on the opinion of web annotators instead of web-creators; and, as web-annotators are the same people as web-consumers, they conclude that the ranking function should be based on their opinions [5]. Compared to our work, they try to predict a globally optimal ranking independent of the query, where we try to optimize the ranking with respect to a specific user and his or her query.

The model that we have used is strongly related to the *FolkRank* method by Hotho et al., which also computes a random walk over the tagging graph [58]. Their random walk model contains a self-transition probability as well as a back teleport. They use the difference between the personalized random walk and an unpersonalized random walk as ranking criterion, resulting in more *serendipitous* recommendations. They only performed an empirical evaluation of their model, without clearly explaining the effect of the model parameters. Jäschke et al. evaluated FolkRank on the tag suggestion task using a snapshot of Bibsonomy and Last.fm data. This evaluation showed that random walk based models strongly outperform the classic collaborative filtering approaches [63], confirmed by our comparisons.

All these methods use the tag-based **UI** matrix, which does not precisely define the user-item relation. We showed that there is a correlation between preference (ratings) and the number of tags assigned in LibraryThing (Figure 4.5b). Also, the performance between both information sources does not deviate significantly. However, in individual tagging systems, where most users are not able to apply tags to the content, the ratings provide essential information that can drastically improve content retrieval. We therefore argue that when explicit user preference data is available, this information should be integrated in the social graph, especially in data with few tags or many negative ratings.

4.8 Discussion

4.8.1 Generalizability

In the generation of our datasets we have applied a pruning policy to reduce the sparseness of the graph. Based on our work we can therefore not make any assumptions about the ranking performance for users with a limited number of items. In Section 4.3.2 we state that because many deployed recommender systems do not give recommendations to users who have provided insufficient preference information we see the proposed pruning step as a fair choice. Based on the differences between the results on T_1 for both the ML and LT dataset we predict that the optimal walk length will be higher for users with very few annotations.

[18] http://www.audioscrobbler.net/

Another issue is whether including users with small profiles in the graph would influence the predicted rankings for users with big profiles. We are confident that these small users will have little influence on the rankings as the random walk model has a strong bias towards strongly connected nodes. Therefore, the users and items that pass our pruning policy will be most influential even when the full data collections would be considered. Based on these observations we conclude that the optimizations in this work are accurate for users with a minimum number of items.

4.8.2 Train-Test Split

In the creation of the training and test set we have chosen to remove a set of randomly selected test items, as opposed to a temporal data split (Section 4.4.1). Primarily, this experimental design choice was forced because the Movielens data does not contain time stamps and the 'reading date' field in LibraryThing is often empty, making temporal splits impossible. Also, splitting temporally would not allow us to perform cross validation on the data, reducing the accuracy of finding the optimal model parameters.

The user-scenario matching our selection procedure is the recommendation of unseen relevant content instead of predicting future content. This is a common scenario in our datasets as people often watch movies or read books that are not recently released.

A completely valid temporal split would take all previous posts into account when making the current prediction. Therefore, a different data split is needed for each predicted ranking. In Section 4.5.3 we have seen that if a single fixed temporal split is applied to bookmarking data, most of the users and items in the test set do not occur in the training set. Based on related work in the RSDC challenge we conclude that only with a significant amount of data processing on the content titles and descriptions predictions can be made for unseen users and items [80; 130].

4.8.3 Only Positive Relations

One of the main limitations of the currently used random walk model is that only positive information can be used in the graph. In rating systems, people however have the opportunity to assign a low rating, which corresponds to a negative opinion on the content. We previously studied the effect of explicitly pulling out the lower ratings and combining the information from the positive and negative graph in a single ranking [24]. This work showed that a user's low ratings can actually improve the ranking of preferred content, because they are usually given to content that has low quality but is still representative for the user's preferred genre. Based on these results we have included the low ratings in the graph as positively weighted edges.

4.8.4 Rank Sinks

In the proposal of the PageRank algorithm, Page et al. discussed the phenomenon where nodes behave as rank sinks [99]. To overcome this problem the original PageRank description contains a small random jump probability, which allows the walk to get out of a sink. Because the social annotation graph is undirected, part of the weight

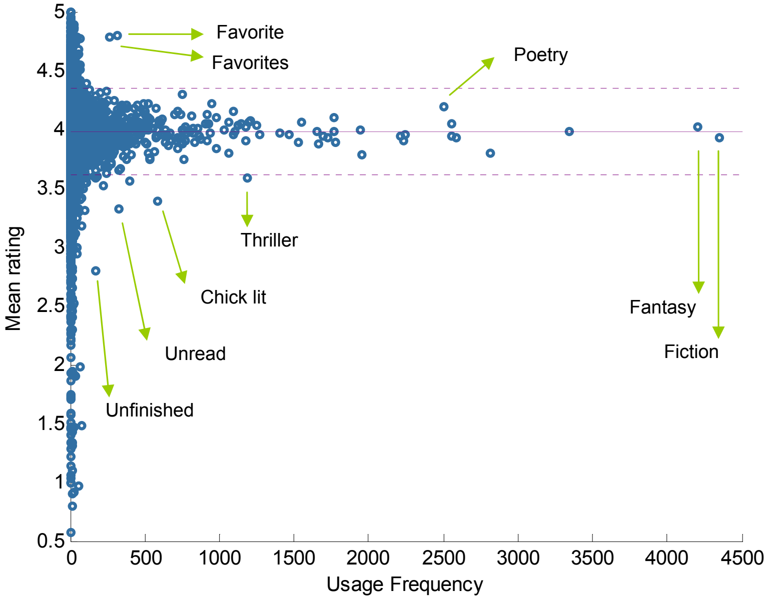


Figure 4.21: The mean rating given to a book with a certain tag versus the number of users of that tag. Each circle represents one tag. The solid line indicates the mean of all ratings and the dotted lines are the mean +1 and -1 standard deviation.

that went through an edge at step n will flow back at step $n+1$. The undirected edges therefore remove the sink problem that occurs in directed graphs, and we have not used this random jump probability in our evaluations.

4.8.5 Tag Relation Flattening

Section 4.3 describes how we flatten the ternary UIT relation, and build our model from the resulting binary relations. This conversion detaches the context from the actual usage of the tag. This loss of information could be problematic if a tag is not used as a content description, but as an opinion about the content. If a user for example uses a descriptive tag like 'poetry', it does not matter on which book he used this tag, as long as we maintain the relations user-poetry and item-poetry. However, if someone uses the tag 'awful' it is important to know to which book this user has assigned that tag, because the tag is a description of the user's opinion about that book.

Figure 4.21 shows the relation between tags and the average rating assigned to the books it was used on in LibraryThing. We see that frequently used tags converge to the mean rating, which means that they do not contain an opinion (e.g. *fantasy* and *fiction*). Only scarcely used tags are sometimes highly correlated to a certain opinion. The bottom rated tags contain words like: *worst book ever*, *horrible*, *stupid*, *awful*, etc. Top rated tags contain: *best book ever*, *incredible*, *very funny*. This graph also demonstrates genre differences like the fact that poetry contains more works of high quality than genres like *thriller* and *chick-lit*.

The small standard deviation ($r \approx 0.4$) indicates that most tags do not deviate very far from the mean rating. The tags ‘favorite’ and ‘favorites’ are the most widely used tags that describe a positive opinion. Below the mean rating ‘unread’ and ‘unfinished’ are the most prominent non-descriptive tags. We have decided not to actively remove these opinionated tags, in order to obtain an honest evaluation of the effect of user generated tags on the retrieval performance. The relatively small amount of opinionated tags gives us the indication that it will not harm our personalization model.

4.9 Conclusions

Amer-Yahia et al. discussed that all the information contributed by the collaborative effort of social network users should be combined in order to enable effective content retrieval in social content systems [2]. By combining both descriptive annotations (social tags) and preference indications (ratings) in a single personalization model, we take a step towards the integration of a user’s social network into the existing categorization and retrieval tasks.

We have shown that a random walk with self transitions is a versatile model to observe the influence of the design choices in social annotation systems on common ranking tasks. This model has also proven to be a comparative ranking model for networked entities. We have used a new dataset collected from LibraryThing, that contains both collaborative tagging information and preference annotations in the form of ratings. As far as we know, this dataset contains the largest number of annotations that contain both tags and ratings ever reported.

The random walk can be used to integrate indirectly related entities in the ranking. We have shown that various tasks benefit from the integration of these latent relations. Especially in sparse graphs, that emerge for example in individual tagging systems, increasing the number of steps through the graph can greatly improve the ranking accuracy.

Storing a user’s personally contributed tags is essential to give annotation suggestions for newly found content or to find like minded users. The influence of a users personal tags on the predicted ranking in content retrieval tasks is however strongly dependent on the system design. We hypothesize that the implementation of tag suggestions in a system creates a more coherent folksonomy and therefore more useful personal tags with respect to search.

The relevance distribution of tags that arises in collaborative tagging systems has shown to be beneficial for effective content retrieval. A social content system should therefore allow all users to tag all available content.

A weighted combination of rating and tag information from the social network can improve both recommendation and search tasks. We promote that these different tasks should therefore not be treated as separate problems but integrated into a single framework.

5

The Influence of Personalization on Tag Query Length in Social Media Search

Social content systems contain enormous collections of unstructured user-generated content, annotated by the collaborative effort of regular Internet users. Tag-clouds have become popular interfaces that allow users to query the database of these systems by clicking relevant terms. However, these single click queries are often not expressive enough to effectively retrieve the desired content. Users have to use multiple clicks or type longer queries to satisfy their information need.

To enhance the predicted content ranking we use a random walk model that effectively integrates the user's preference and semantically related query terms. We use the collaborative annotations from a popular on-line book catalog to create a social annotation graph and study the effect of personalization and smoothing for increasing query lengths.

We show that personalization and smoothing allow the user to find equally relevant content with fewer query terms compared to a frequency based content ranking with TF-IDF weighing. As expected, we see that the influence of the random walk model disappears if users type more detailed queries. Finally, we discuss the observations with respect to synonyms and homographs which are well known to hamper the performance of information retrieval systems.

This chapter is published as: Maarten Clements, Arjen P. de Vries, and Marcel J. T. Reinders. The Influence of Personalization on Tag Query Length in Social Media Search. *Information Processing & Management*, 46:403-412, May 2010.

5.1 Introduction

In the last decade, the explosive use of budget digital cameras and integrated multimedia devices has resulted in an enormous increase in user-generated multimedia content like movieclips and pictures. On-line databases are actively used to store and share this content. Recently, the addition of social aspects in these databases has resulted in a large popularity increase. Millions of people use these *social content systems* to publish their creations or to be entertained by other people's contributions. Since the contributed data often does not carry a clear contextual description and there is no librarian to categorize the content, this has resulted in huge collections of unstructured data.

For future retrieval, many network users actively annotate the content using tags. Although most people use tagging to organize their own content collection, it has been shown that social tagging results in semantically descriptive annotations that can be used for content retrieval by the entire network [44; 83]. To initiate content retrieval, social tags are often shown in a *tag-cloud*, a visual depiction of tags in which the more popular tags are typeset in a larger font or more prominent color. Although there exist many different methods to draw these clouds [67], the relevance of a tag is often based on the global popularity of the tags in the entire network (e.g. popular tags in Last.fm[1]). In this way of navigation only a single popular word is used as a query, resulting in many retrieved documents. In traditional information retrieval as well as web-search engines, people often use multiple word queries in order to disambiguate their information need. To enable effective content ranking, users should therefore click multiple times on the suggested tags to make their query more specific. In this work we show that a personalized system that takes latent semantic relations into account can aid the user in his search for the desired content.

We focus on social content systems that enable *collaborative tagging* to annotate the available content. In collaborative tagging systems (like CiteULike[2] and Del.icio.us[3]), every user can tag any piece of content. In this way, users indicate which aspects of the content correspond to their personal interest. Also, the aggregated tags of the network users create a relevance distribution for each content element. Furnas et al. already stated in 1987 that people often choose different terms to annotate content, resulting in low precision retrieval [40]. They argued that a theoretically optimal system would allow *unlimited aliasing* (assigning an infinite number of annotations) to describe the content. We advocate that collaborative tagging approaches unlimited aliasing and is therefore required to enable effective personalized content retrieval.

Besides tagging, the social aspects of networks stimulate people to share their opinion about the provided content. In many interfaces people can assess the quality of the content by giving a rating. With the introduction of ratings and tags in on-line databases, content annotation has shifted to subjective categorization. The combination of these two information sources creates a non-hierarchical database categorization based on both content quality and topic. Using ratings and tags, we

[1] http://www.last.fm/tags

[2] http://www.citeulike.org

[3] http://del.icio.us

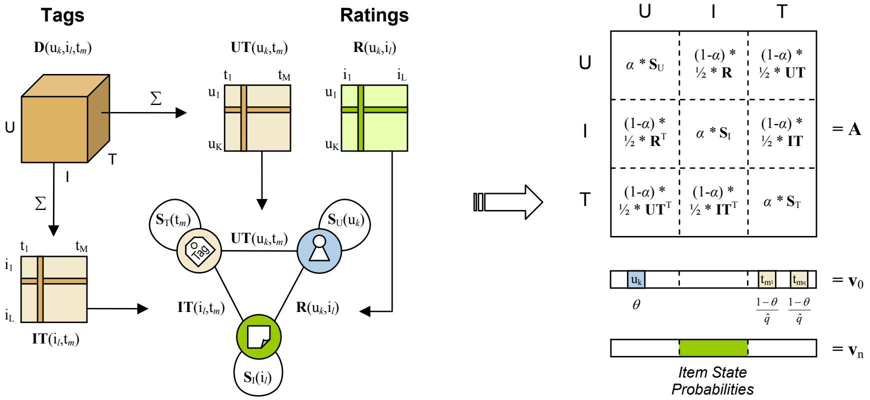


Figure 5.1: We create the graph based on rating ($\mathbf{R}$) and tag count ($\mathbf{UT}$, $\mathbf{IT}$). Self transitions ($\mathbf{S}$) allow the random walk to stay in the same node with probability α. Together, these edges constitute transition matrix $\mathbf{A}$. In the initial state vector $\mathbf{v}_0$, the indexes corresponding to the target user and the selected query tags are assigned with weights θ and $1-\theta$. If multiple ($\hat{q}$) tags are selected as query, the weight of $(1-\theta)$ is divided amongst these tags ($\mathbf{v}_0(t_{m_1,\ldots,m_q}) = (1-\theta)/\hat{q}$). The result of the walk $\mathbf{v}_n$ contains the relevance probabilities of all three network elements.

create a graph of the network, resembling the actual relations in social content systems. We use a personalized random walk over this graph to evaluate the retrieval performance of queries with increasing number of terms.

5.2 Personalization Model

For the relevance ranking of the content based on the selected query tags we propose to use a random walk over the graph, created by all rating and tagging actions. The random walk has proven to be an effective ranking method for networked entities [99; 76; 58; 29]. For a more elaborate evaluation of this model on the effect of different annotation methods on ranking tasks in social media we refer to our previous work [26]. For clarity we now briefly discuss the model.

A random walk is a stochastic process in which the initial condition is known and the next state is given by a certain probability distribution. This distribution can be represented by the *transition matrix* $\mathbf{A}$, where $a_{i,j}$ contains the probability of going from node i (at time n) to j (at time $n+1$):

$$a_{i,j} = P(S_{n+1} = j | S_n = i) \tag{5.1}$$

The initial state can now be represented as a vector $\mathbf{v}_0$ (with $\sum(\mathbf{v}_0) = 1$), in which the query elements can be assigned. By multiplying the state vector with the transition matrix, we can find the state probabilities after one step in the graph ($\mathbf{v}_1$). Multi step probabilities can be found by repeating the multiplication $\mathbf{v}_{n+1} = \mathbf{v}_n\mathbf{A}$. The number of steps taken in the random walk determines the influence of the initial state vector versus the background distribution. Under certain graph conditions, $\mathbf{v}$ will become stable (so that $\mathbf{v}_\infty = \mathbf{v}_\infty \mathbf{A}$) and in a completely connected graph it will contain the background probability of all nodes in the network.

5.2.1 Transition Matrix ($\mathbf{A}$)

Figure 5.1 shows how we create the transition matrix by combining rating and tagging information. If users, items and tags are seen as separate entities, the act of tagging creates a ternary relation between them. These relations can be visualized in a 3D matrix $\mathbf{D}(u_k, i_l, t_m)$, where each position indicates if user u_k (with $k = \{1, \dots, K\}$) tagged item i_l (with $l = \{1, \dots, L\}$) with tag t_m (with $m = \{1, \dots, M\}$).

Even collaborative tagging systems are usually very sparse, therefore we propose not to use the ternary relations directly, but sum over the 3 dimensions of $\mathbf{D}$ to obtain:

UT matrix: $\mathbf{UT}(u_k, t_m) = \sum_{l=1}^{l=L} \mathbf{D}(u_k, i_l, t_m)$, indicating how many items each user tagged with which tag.

IT matrix: $\mathbf{IT}(i_l, t_m) = \sum_{k=1}^{k=K} \mathbf{D}(u_k, i_l, t_m)$, indicating how many users tagged each item with which tag.

UI matrix: $\mathbf{UI}(u_k, i_l) = \sum_{m=1}^{m=M} \mathbf{D}(u_k, i_l, t_m)$, indicating how many tags each user assigned to each item.

In this representation the $\mathbf{UI}$ matrix does not contain a clear indication of the users' preference towards the available content. Therefore, we replace the tag based User-Item matrix by the matrix based on the users' ratings. The rating matrix ($\mathbf{R}(u_k, i_l)$) contains the explicit users' preference for the available content, often expressed on a five or ten point scale.

Using the nonzero matrix values as edges, these three matrices ($\mathbf{UT}$, $\mathbf{IT}$ and $\mathbf{R}$) constitute a tripartite graph with users, items and tags as nodes. We include self-transitions that allow the walk to stay in place, which increases the influence of the initial state. The self transitions are represented by an identity matrix $\mathbf{S}_U = \mathbf{I}_K$, so that the weight of the self transitions is equal for all nodes.

To reduce the influence of frequently occurring elements, we use TF-IDF weighing on the input matrices [110]. For example, the weighted User-Tag matrix is computed by:

$$\mathbf{UT}_{\text{TF-IDF}}(u_k, t_m) = \mathbf{UT}(u_k, t_m) * \log\left(\frac{K}{\sum_{k=1}^{k=K} \operatorname{sgn}(\mathbf{UT}(u_k, t_m))}\right) \tag{5.2}$$

where the sign function (sgn) sets all values > 0 to 1. Before combining the matrices we normalize them so that all rows sum to one.

We combine $\mathbf{UT}$, $\mathbf{IT}$ and $\mathbf{R}$ in the transition matrix $\mathbf{A}$, as shown in Figure 5.1. For each relation we create an undirected edge by putting $\mathbf{UT}$, $\mathbf{IT}$ and $\mathbf{R}$ in the upper diagonal of $\mathbf{A}$ and the transposed matrices in the lower diagonal. Due to the normalization of the submatrices, the rows of $\mathbf{A}$ now sum to 1, so they can be used as transition probabilities.

In this model $\alpha \in [0, 1]$ is the weight of the self transitions. This parameter is strongly related to the optimal length of the random walk, we can therefore fix α and optimize only the walk length. Earlier work has shown that the optimal retrieval performance is not strongly dependent on the choice of self transition probability [26; 29]. A slower walk does give a higher resolution to pick the optimal walk length, we therefore fix the self-transition probability (α) to a relatively high value of 0.8.

In the initial state vector ($\mathbf{v}_0$), the starting points of the walk are assigned according to the target user and the query terms. The state probabilities after n steps are computed by repeating the multiplication of the state vector and the transition matrix $\mathbf{A}$. After n steps, the content ranking is obtained by ordering the part of $\mathbf{v}_n$ that corresponds to content ($\mathbf{v}_n(K+1,\ldots,K+L)$) according to the state probabilities. This ranking will also contain the training data (i.e. the items already rated by the target user). We assume that a different user interface is used to browse previously seen content (the user's library), therefore we remove the training examples from the final ranking.

5.2.2 Query Weight (θ)

Most tag-based retrieval systems use the selected tag as query term and rank the content according to popularity or freshness. In the proposed model the query tag can be enriched by integrating the users history in the search. In the initial state vector, both the query tag and the target user are assigned a value according to θ ($\theta \in [0,1]$): $\mathbf{v}_0(u_k) = \theta$ and $\mathbf{v}_0(t_m) = 1-\theta$ (where u_k is the target user and t_m indicates the set of selected query tags: $m = \{m_1,\ldots,m_{\hat{q}}\}$). The weight of $1-\theta$ will be divided over the number of tags that are selected as query ($\hat{q}$). When θ is set to 0, the state probabilities only depend on the selected query tags, so the result will not be personalized for u_k. When $\theta = 1$ the state probabilities depend only on the profile of the target user, so the predicted content ranking will not be relevant to the query, which closely resembles collaborative filtering [106].

5.3 Data

5.3.1 LibraryThing

LibraryThing[4] is an on-line web service that allows users to create a catalog of the books they own or have read. A user can tag and rate all the books he adds to his personal library. The social aspects of this network give the user the opportunity to meet like-minded people and find new books that match his preference. The popularity of the system has resulted in a database that contains almost 4 million unique works, collaboratively added by more than 500,000 users. We are not aware of any other open network where both collaborative tagging ($\approx$ 40 million) and rating ($\approx$ 5 million) are actively used by the community.

We have collected a trace from the LibraryThing network, containing 25,295 actively tagging users[5]. After pruning this data set we retain 7279 users that have all supplied both ratings and tags to at least 20 books. We remove books and tags that occur in fewer than 5 user profiles, resulting in 37,232 unique works and 10,559 unique tags. This pruned data set contains 2,056,487 UIT relations, resulting in a

[4] http://www.librarything.com

[5] Crawled in July 2007

density of $7.2 * 10^{-7}$ (fraction of non empty cells in $\mathbf{D}$). The derived $\mathbf{R}$, $\mathbf{UT}$ and $\mathbf{IT}$ matrices have a density of respectively: $2.8 \cdot 10^{-3}$, $5.2 \cdot 10^{-3}$ and $2.0 \cdot 10^{-3}$.

Figure 5.2 shows all annotations sorted by the number of assigned tags (q). In most related literature this relation is modeled by a power-law distribution [49; 119]. Although word frequencies in full documents indeed follow a power-law [93], the distribution of tag assignments seems to deviate from a true power-law for the lower values ($q < 5$). It was proposed by Arampatzis and Kamps to model the distribution of query length in web search by a combination of a Poisson and power-law model [3]. We find that their findings strongly correspond to the annotation length distribution in LibraryThing if we use the Poisson distribution for $q \leq 5$ and a power-law fit for $q > 5$:

$$P_Q(q) = \begin{cases} \frac{\lambda^q e^{-\lambda}}{q!} & \text{if } q \leq 5 \\ Cq^{-s} & \text{if } q > 5 \end{cases} \tag{5.3}$$

We find the optimal fit with parameters $\lambda = 2.3$ for the Poisson distribution and $C = 83, s = 4.4$ for the Power-law.

Arampatzis and Kamps compared various TREC and AOL data sets and found that the average exponent is close to 5 for web query length [3]. In full textual documents the exponent of word frequencies is generally known to be lower, for example Newman found a value of 2.2 for English text in the book *Moby Dick* [93]. The slope of the Power-law fit on the LibraryThing tagging data (Exponent: 4.4) lies between the distributions observed in web queries and in full English documents. This shows that the annotations that people make in social tagging systems are more exhaustive than queries but more focused than full documents.

In our experiments we will assume that people would use a subset of the terms they have used to annotate their content if they needed to retrieve this content. Here we make the common assumption that the query and document are derived from the same language model. We obtain the query tags by randomly selecting $\hat{q}$ terms from the set of tags that were originally assigned by the user ($\hat{q} \leq q$). When increasing the number of query tags we keep the previously selected set and enrich the query with an extra randomly selected tag. As in any text based retrieval system, we assume that we can only answer queries with terms that occur in the corpus.

To study the effect of personalization on the retrieval performance for increasing number of query tags ($\hat{q} \in 0, \ldots, 4$), we want to use the same validation set for each value of $\hat{q}$. Therefore only the annotations (tags assigned by a single user on a single item) that consist of 4 or more tags can be used for evaluation. Figure 5.3 shows the number of annotations that qualify for our analysis per user, sorted in descending order. Based on the extensive study of Mislove et al. on the effect of crawling methods on the collected data, we expect that this distribution lacks a power-law tail because our data crawl was biased towards users with many annotations [88].

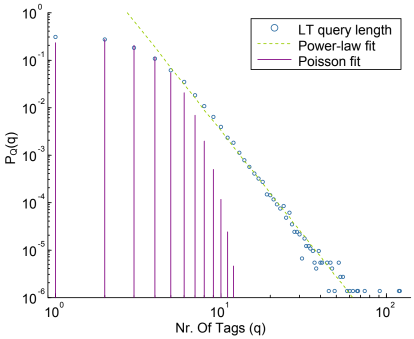


Figure 5.2: The probability of observing q tags in an annotation, with fitted Poisson ($q \leq 5$) and power-law ($q > 5$) distributions.

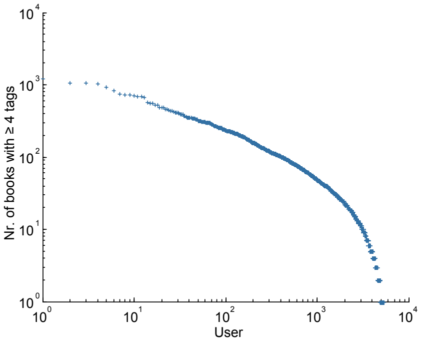


Figure 5.3: Users sorted by number of books they annotated with $q \geq 4$. 1844 users never applied 4 or more tags on a single book.

5.4 Experimental Setup

5.4.1 Data Preparation

In order to estimate the performance of our model without overfitting to the data, we split the data in two equal parts (see Figure 5.4). Together with all the created annotations (ratings and tags), half of the users (3640 profiles) are put into the *training* set and the other half constitute the *test* set (*Step 1*). We now use the training set to optimize the model parameters by holding out $1/5$ of the items of $1/5$ of the training users (the validation set).

We use our model to predict the held-out content (*Step 2*) using the tags assigned by the target user as query for the content he applied the tags on. From the user's validation set we select 4 random tags that were used together on at least one item. We use $\hat{q}$ of these tags as a query and compute the NDCG measure (discussed in the next section) over the items that were tagged by this user with these tags. We repeat this selection until every item that was given at least 4 tags by this user has been evaluated at least once.

We compute the mean score over all validation users in the training set and to obtain stable results we repeat the optimization for all 5 independent user splits. We compare the obtained mean performance for different settings of n and θ to find the optimal model for personalized search (*Step 3*, Figure 5.5). There are 1844 users who have never used 4 or more tags to annotate a book (Not visible in Figure 5.3 because of the logaritmic scale). These users will therefore not be part of our evaluation set, but will be present in the graph.

The optimal model parameters derived from the training set are used to compute the performance on the test set, by holding again $1/5$ of the user profiles of $1/5$ of the users out, and computing the NDCG (*Step 4*). Finally we compare the results of our optimal model to the results achieved with conventional methods (*Step 5*, Figure 5.7).

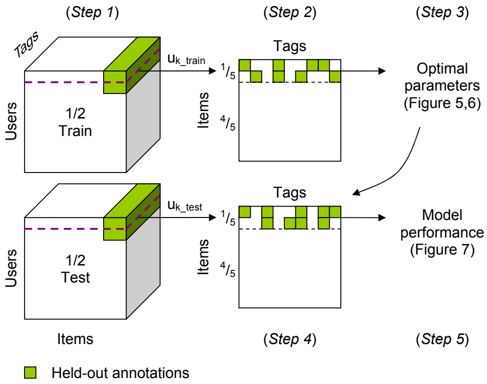


Figure 5.4: *Step 1*: Splitting of the $\mathbf{D}$ matrix, the $\mathbf{R}$ matrix is split accordingly. *Step 2,4*: A slice of the matrix contains a single user's items and tags. *Step 3,5*: The tags used by that user are used to predict the held-out content.

5.4.2 NDCG Evaluation

To evaluate the predicted content ranking, we use the Normalized Discounted Cumulative Gain (NDCG) proposed by Järvelin and Kekäläinen [62].

In the predicted content ranking, the rank positions of the held-out validation ratings that correspond to a positive opinion $r \in \{3, 3.5, 4, 4.5, 5\}$ are assigned a value of respectively $G \in \{1, 2, 3, 4, 5\}$, called the *gain*.

In order to progressively reduce the gain of lower ranked test items, each position in the gain vector is discounted by the $\log_2$ of its index (where we first add 1 to the index, to ensure discounting for all rank positions > 0). The Discounted Cumulative Gain (DCG) now accumulates the values of the discounted gain vector:

$$\mathrm{DCG}[i] = \mathrm{DCG}[i-1] + G[i]/\log_2(i+1) \tag{5.4}$$

The DCG vector is normalized to the optimal DCG vector. This optimal DCG is computed using a gain vector where all test ratings are placed in the top of the ranking in descending order. Component by component division now gives us the NDCG vector in which each position contains a value in the range $[0, 1]$ indicating the level of perfectness of the ranking so far. We use the area below the NDCG curve as score to evaluate our rank prediction.

5.5 Experiments

We will use the proposed random walk model to discuss the retrieval performance for increasing query length. For each number of tags ($\hat{q} \in 0, \ldots, 4$) we will compare the optimal ranking of our random walk model to a frequency based ranking. This frequency ranking is obtained by taking one step through the graph with $\theta = 0$. In this case the ranking will only be based on the number of people who have assigned the query tag(s) to the content. Due to the TF-IDF weighting on the $\mathbf{IT}^T$ matrix this ranking corresponds to using a simple vector space model with TF-IDF weighting.

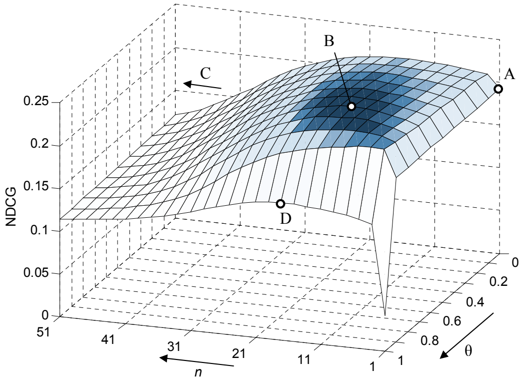


Figure 5.5: Optimization of the personalization influence θ and the walk length n for $\hat{q} = 1$. The test points are indicated with small circles. A: The frequency ranking for $\hat{q} = 1$, B: The optimized ranking for $\hat{q} = 1$, C: The query independent popularity ranking ($\hat{q} = 0$), D: The query independent optimal ranking. Note that the NDCG at $n = 1$ and $\theta = 0$ is equal to a random ranking, because a user has no direct link to potentially interesting content.

5.5.1 Smoothing and Personalization

To find the optimal model parameters and evaluate the sensitivity of the model we use the random walk to predict the left-out content of the training part of the LibraryThing data. We will first use only one tag ($\hat{q} = 1$) and the target user as query. Figure 5.5 shows the effect of the personalization (θ) at different walk lengths. The optimal NDCG is found at $\theta = 0.6$, which means that personalized retrieval gives a more accurate prediction than both completely personal and completely tag based queries (Figure 5.5, point B).

We also find that the optimal number of steps is larger than one ($n = 13$), which means that the random walk improves a content ranking based on direct relations. Content that has not been tagged extensively will often miss the terms used as a query by other people. The random walk smoothly integrates these latent relations that are not explicitly present in the data. When the number of steps further increases ($n \rightarrow \infty$) the state vector will converge to the stable distribution based on the global network popularity of the content. As expected the mean NDCG drops to a lower value if the content ranking *forgets* the information about the initial query.

If $\theta = 1$ the content ranking will be completely based on the user's preference. This setting therefore represents the optimized model for $\hat{q} = 0$ (Figure 5.5, point D). When $n \rightarrow \infty$ the state vector will converge and the content ranking will be independent on the initial query (and θ). We therefore use the popularity ranking based on the finite state vector as frequency ranking with $\hat{q} = 0$ (Figure 5.5, point C). The difference in NDCG between $n = 99$ and $n = 101$ is $5.1 \cdot 10^{-5}$ which is smaller than the reported significance, therefore we use the result at $n = 101$ as finite state vector.

We repeat the optimization for increasing number of query tags and show the optimal model parameters in Figure 5.6. If we add more tags to the query we see that

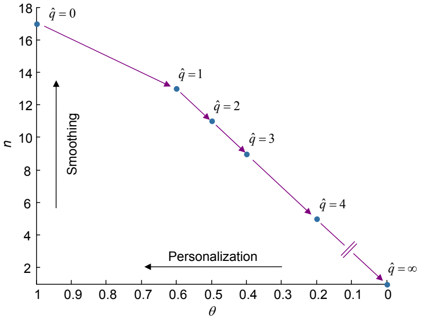


Figure 5.6: The optimal parameter settings for increasing number of query terms. As expected the optimal parameters converge to the frequency ranking ($\theta = 0$, $n = 1$) when a user types a very detailed query.

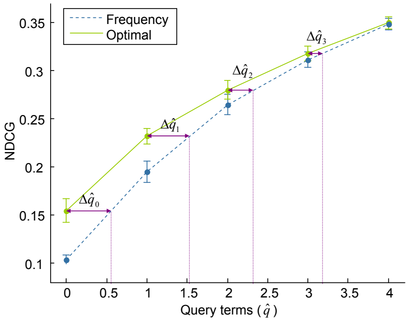


Figure 5.7: The mean NDCG for the optimal model and frequency based ranking for 0 to 4 query terms. For each setting we indicate the gain in number of query terms ($\Delta\hat{q}$). The errorbars indicate the standard deviation (σ) over the 5 folds of the cross validation.

the optimal parameter settings converge to the frequency based ranking ($\theta = 0$ and $n = 1$). If a user puts more effort in indicating his information need by giving more query terms to the system the influence of personalization and smoothing diminishes.

5.5.2 The Influence of Personalization and Smoothing

To evaluate our model performance without overfitting to the data, we use a separate test set as discussed in Section 5.4.1. For each number of query terms we will compare the NDCG with optimal model parameters to the NDCG obtained with frequency ranking. The mean NDCG and standard deviation (σ) are shown in Table 5.1 and visually depicted in Figure 5.7.

These results show that the optimal model indeed converges to the frequency based model when the user issues longer queries. When the query consists of 4 tags there is an insignificant performance difference between both models. When the query consists of 4 tags only the user itself can make his information need more specific by selecting more tags.

We map the result of the optimal model to the interpolated NDCG obtained with frequency ranking. In this way we can express the performance difference in a number of hypothetical query terms ($\Delta\hat{q}$). We find that optimized content recommendation ($\hat{q} = 0$) gives the same performance as a frequency ranking based on 0.56 tags. This indicates that the accuracy of recommendations is closer to a ranking based on a user selected query term than to the popularity ranking.

If a user selects a tag from a tag-cloud ($\hat{q} = 1$), a system that uses both personalization and smoothing can give the same performance as a frequency based system with 1.53 query terms. If the user decides to put more effort in his search by clicking more terms or typing a longer query the gain of the personalized model disappears.

Table 5.1: Test results for optimal and frequency ranking. For comparison, a random ranking gives an NDCG of 0.048

	Frequency				Optimal				
$\hat{q}$	θ	n	NDCG	σ	θ	n	NDCG	σ	$\Delta\hat{q}$
0	0	101	0.104	$4.5 \cdot 10^{-3}$	1	17	0.155	$1.2 \cdot 10^{-2}$	0.56
1	0	1	0.195	$7.8 \cdot 10^{-3}$	0.6	13	0.232	$1.1 \cdot 10^{-2}$	0.53
2	0	1	0.265	$9.7 \cdot 10^{-3}$	0.5	11	0.280	$1.1 \cdot 10^{-2}$	0.32
3	0	1	0.311	$7.6 \cdot 10^{-3}$	0.4	9	0.318	$7.9 \cdot 10^{-3}$	0.18
4	0	1	0.348	$6.0 \cdot 10^{-3}$	0.2	5	0.350	$6.0 \cdot 10^{-3}$	n/a

5.6 Discussion

5.6.1 Related Work

Research on query length in web search has shown that users generally do not like to type exhaustive queries. The reported average query lengths range from 2 to 4 terms, while less than 4% of the queries consist of 6 terms or more [61; 3]. Belkin et al. showed that longer queries however result in higher user satisfaction and compared various interfaces to elicitate users to type longer sentences [9]. This work advocates that personalization provides another way to disambiguate the multiple meanings encoded by (too) short queries.

Personalized web-retrieval systems are usually divided into three categories *re-ranking*, *filtering* and *query expansion* [68]. Re-ranking and filtering methods first need to get a list of documents that contains at least the relevant set. Most of the proposed methods take the results from existing search engines and adapt the obtained document set to the user's preference [94; 120]. Collaboratively annotated social media are however characterized by sparse data descriptions. The initially retrieved list with documents that match the query terms might therefore be too short to allow re-ranking or filtering. The graph ranking method we have used exploits the network structure which is inherently present in social media. Therefore it allows content to be ranked, even if it does not contain the initial query terms.

Personalized retrieval systems need information about the user's preference. This user profile can either be *explicitly* or *implicitly* created. In web-retrieval, explicit profile creation is unpopular because it usually means that users have to spend extra time to fill in long forms with personal information [68]. In social media, tagging and rating profiles are explicitly created but not only serve the purpose of personalization. People naturally see the importance of tagging and often enjoy giving their opinion by supplying a rating. In this work we have represented the preference of the user by his explicitly created tags and ratings. We have shown that personalization based on query expansion with a users previous annotations can on average improve short queries by half a query term.

Furnas et al. observed that people use a surprisingly great variety of words when they refer to the same thing. In a large user study they found that the probability

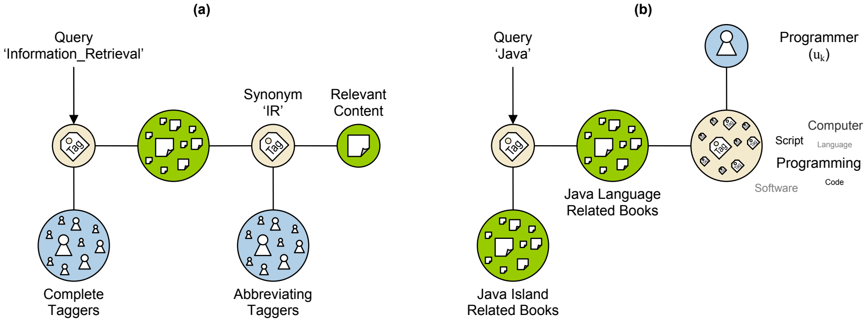


Figure 5.8: Our model is robust against both synonym and homograph problems: a) Synonyms or spelling differences (like 'Information_Retrieval' and the abbreviation 'IR') reinforce the content ranking because of the soft clustering created by the random walk model. b) Homographs like *Java* can be disambiguated using the target user's history.

that two users favored the same term was < 0.20, resulting in big failure rates of IR systems. Begelman et al. acknowledged this vocabulary problem in the social tagging context and used clustering methods to group tags with strong semantic relations to enable effective content access [8]. In previous work we have used the random walk model to evaluate various retrieval tasks in differently designed tagging systems. We have shown that the positive effect of the random walk on the reduction of the vocabulary problem becomes larger in sparser graphs [26]. Here we have shown that the effect of the random walk model is strongly dependent on the length of the query. Especially the commonly short queries can benefit from the integration of latent relations.

5.6.2 Synonym and Homograph Robustness

Well known problems in tagging systems are synonyms and homographs. Synonyms are different words that share the same or closely related meaning. The problem in tagging systems arises, because there is no clear regulation on which words to use. If a piece of content has been tagged with a certain word and someone with a different background uses its synonym as a query, the content might not be found. The same problems arise when people use abbreviations, singular or plural words, word combinations and different languages. If a tag cloud is used to query a database, only a single word is used as initial query resulting in sub-optimal retrieval performance. A user has to click multiple tags if he needs to disambiguate his initial query.

Clustering methods have been proposed to group tags with strong lexical relations [8]. Clustering algorithms create binary relations between concepts although the natural similarity between words is a continuous relation. A random walk has shown to have a soft clustering effect that smoothly relates similar concepts before converging to the background probability [127]. Figure 5.8a shows that if enough users have tagged certain content, a large number of paths will exist between synonyms or otherwise semantically related terms. The soft clustering effect will group these strongly connected entities which makes the random walk robust against synonymity problems.

Homographs are words that do not necessarily have the same pronunciation, but are written in exactly the same way. If a browsing user selects a homograph as query, the system will not know which denotation the user aimed for. In order to disambiguate the terms a user is looking for, our personalized random walk model integrates the information about the past behavior of that user. By starting the random walk at both the query tag and the target user, the content that matches the target user's preference is more likely to be found first (see Figure 5.8b.). Integrating the user in the query can therefore reduce the number of clicks a user has to do to reach the desired content.

5.7 Conclusions

The number of tags per annotation in social media closely resembles the number of terms people use in web-queries. We expect that the same generative process forms the basis of both events. We have therefore used the tags assigned by the users as their hypothetical queries to retrieve this content.

Retrieval models in social content systems can greatly benefit from personalization and smoothing. Due to the vocabulary differences between the network users a simple vector space model will not find all the content related to a short tag-based query. In social media, people explicitly create their preference profiles by annotating their content. The graphical structure that arises in these systems can effectively be used to give recommendations or rank the content according to the users' queries.

We have shown that there is a clear relation between the length of the query and the quality of the predicted content ranking. For queries shorter than 4 terms the proposed model can significantly improve the content ranking. When a user decides to put more effort in his query the positive effect of personalization and smoothing diminishes.

Part III

Geotags

6
Finding Wormholes with Flickr Geotags

We propose a kernel convolution method to predict similar locations (wormholes) based on human travel behaviour. A scaling parameter can be used to define a set of relevant users to the target location and we show how the geotags of these users can effectively be aggregated to predict a ranking of similar locations. We evaluate results on world and city level using independent test sets collected from Wikipedia and GeoNames.

This chapter is an extended version of: Maarten Clements, Pavel Serdyukov, Arjen P. de Vries, and Marcel J. T. Reinders. Finding Wormholes with Flickr Geotags. In Cathal Gurrin et al., editors, *ECIR 2010, LNCS 5993*, pages 658-661. Springer-Verlag, Berlin, Heidelberg, 2010.

6.1 Introduction

Many visitors of travel websites like www.expedia.com have not decided on a location for their holiday at the moment they enter the website. To actively assist indecisive users, these websites show cheap travel deals. Knowledge about travel locations similar to the queries of the user can boost the sales by showing targeted advertisements.

We define *wormholes* as similar, but not necessarily spatially close locations on the planet. We hypothesize that users have a specific travel preference and therefore visit locations that are to some extend similar. Furthermore, making a photo at a visited location is an indication that the user likes that location. Based on these hypotheses, the aggregated travel data of many users should be able to reveal which locations are most similar to a given query location. In photo sharing websites like Flickr[1], users can indicate the geographical location of their pictures by placing them on a world map. We propose a method, similar to neighborhood based collaborative filtering, to combine the users' *geotags* in a prediction for similar locations.

The exploitation of geotags has shown to be effective with for various tasks. A method for global event detection has been proposed by Rattenbury et al. [104]. Ahern et al. made a mapping of popular tags to geographical locations [1]. This work was extended by Kennedy et al., who select relevant pictures for the predicted clusters [69]. Crandall et al. suggested not to use a fixed number of clusters and proposed a mean shift algorithm to find the most prominent landmarks and representative photos [28].

Another application of Flickr's geotags was proposed by Lee et al. who used the geographic clusters related to a tag to improve the prediction of similar tags [74]. Furthermore, several methods have been proposed to predict the geotags of a photo, based on its textual tags [117], visual information [28] and individual user travel patterns [66].

As far as we know we propose a first attempt to predict similar locations based on geotags alone. Textual tagging will always require manual effort and cameras with GPS functionality will become mainstream in the coming years. Therefore we believe that future data collections will contain more geotag data than manual annotations, and data analysis will rely more strongly on geotags. The contributions of this work can be summarized as follows:

1. We propose a weighing scheme to estimate the relevancy of a user to a given location at various scales.
2. We compare several methods to aggregate user information in a way that accurately predicts similar locations.
3. We propose an evaluation technique for similar location prediction, based on an independent test set.

[1] http://www.flickr.com

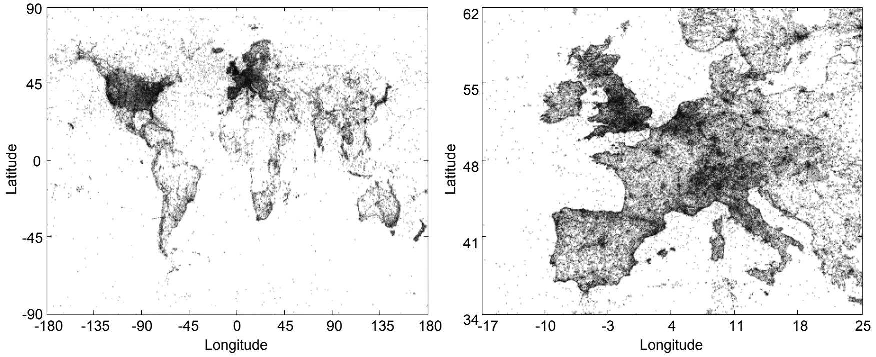


Figure 6.1: A 2D histogram of the data clearly shows the most developed areas in the world. The zoom on Europe shows that the geotags of all users make an accurate map which clearly shows the big cities and coast lines.

6.2 Flickr Geotag Data

Using the public API of Flickr we have collected the top-100 most popular localities (cities, parks, etc.) for each day in 2008. The aggregated data contains 8,643 places including the number of photos geotagged in 2008 by all Flickr users. To retrieve the geotagged data, we now repeatedly follow the following procedure:

1. Select a location l from the full distribution, with the probability relative to the global popularity in 2008.
2. Get a photo i_l from this location.
3. Get all the photos from the user who made i_l.

Following this strategy, we have collected the geotags of 36,264 users. Together these users have uploaded 52,425,279 photos of which 22,710,496 have been geotagged. This set contains over 20% of all public geotags available in Flickr in October 2009[2]. The histogram of all users' geotags gives an accurate representation of the most developed areas in the world (Figure 6.1). Looking at the distibution of the number of photos and annotations per user (Figure 6.2) we see that many users have a large number of geotags. Based on related work on social annotation data we expect that the true data distribution will follow a power-law [93; 119]. The proposed crawling strategy is clearly biased towards active users. For the experiments proposed in this work the long tale of the geotag data can be ignored, since we need users that have shown interest in several distinct locations.

6.3 Wormhole Detection

From a given target location L we want to find the most similar locations around the world. For each user u, a weight $W_{L,u}$ is computed based on the distance of

[2] According to: http://www.flickr.com/map

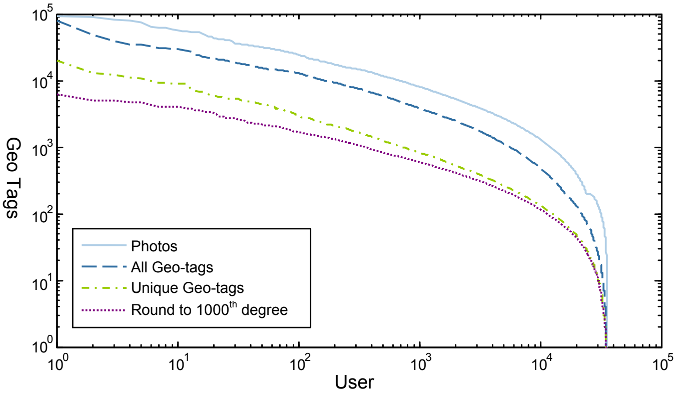


Figure 6.2: The distribution of number of photos, geotags, unique geotags and unique geotags when rounded to a 1000th degree (max. 111 meter) per user. The large gap between these distributions shows that the data is strongly clustered. This is expected as people make many pictures around famous landmarks. The relatively small difference between the number of photos and geotags shows that the proposed crawling strategy has a strong bias for users who geotag their photos.

the nearest geotagged photo of the user to the target location, weighted by a normal distribution (to decrease the influence users that have never been close to the target):

$$W_{L,u} = \exp\left(-\frac{\min_i(d(L, G_{u,i}))^2}{\sigma^2}\right), \tag{6.1}$$

where standard deviation σ is used as a scaling parameter and $d(L, G_{u,i})$ computes the euclidean distance between the i^{th} geotag of a user $G_{u,i}$ and L. The scale parameter σ can be compared to the number of users selected in neighbourhood based collaborative filtering algorithms [12], as it determines which users are similar enough to contribute to the prediction. The estimation of the relevancy of a user to a given location might be improved by using the time that a user spent at a certain location [101].

The wormholes from L are now derived by creating a 2000x4000 histogram H_L of all users' geotags, using $W_{L,u}$ as weight per user. For each geotag in the collection we add the weight of the corresponding user to the respective bin. The choice of grid size results in cells that are at most 10x10km (around the equator). The longitudinal size of the cells reduces to 5km around mid Canada and south of Chile.

Using a grid is prone to errors when locations close to one of the cell boundaries are considered. To transfer the weight of a single grid cell to the neighbour cells, we perform a kernel convolution of the histogram with a Gaussian kernel (with the same σ as used in Equation 6.1). The difference between the resulting profile and a distribution based on all users ($W_{L,u} = 1$ for all u and convolution with same σ) gives a score that indicates the relevance of each position on earth with respect to the target location L.

We evaluate this method for increasing values of σ and 3 different aggregation methods:

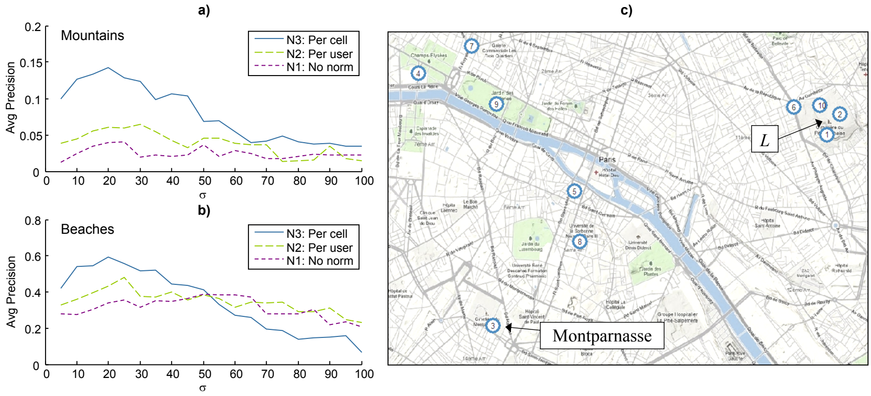


Figure 6.3: **a)** Prediction of mountains based on the 5 major summits (σ in km). **b)** Prediction of beaches based on Platja de Lloret de Mar (σ in km). **c)** Wormholes in Paris from Père Lachaise ($\sigma = 60m$, Method: N3).

N1. Add all the user's geotags with $W_{L,u}$ to the histogram:

$$H_L(bin) = \sum_{\forall u} \sum_{\forall G_{u,i} \in bin} W_{L,u}. \tag{6.2}$$

N2. Normalize per user by dividing the computed weight by the number of geotagged photos of each user ($|I_u|$):

$$H_L(bin) = \sum_{\forall u} \sum_{\forall G_{u,i} \in bin} W_{L,u}/|I_u|. \tag{6.3}$$

N3. Limit the contribution of each user to only one photo per histogram bin:

$$H_L(bin) = \sum_{\forall u} \sum_{\forall G_{u,i} \in bin} W_{L,u}/|G_{u,n} \in bin|. \tag{6.4}$$

6.4 Results for some Mountains, a Beach and a Cemetery

To evaluate our wormhole prediction method, we collect 156 test mountains from Wikipedia[3].We use the list of 5 summits (highest mountains per continent, excluding Antarctica and Oceania) as starting locations[4]. We now evaluate the result by taking the top ranked grid cell and count it as positive if one of the test mountains is found within a radius of 3 cells around that cell. For mountain prediction this can be motivated because many people do not visit the actual summit, but hike around the slope, which can extend for several tens of kilometers. Figure 6.3a shows the mean average precision over the top-50 predicted peaks based on the five summits for increasing σ.

[3] http://en.wikipedia.org/wiki/List_of_peaks_by_prominence

[4] http://en.wikipedia.org/wiki/List_of_mountains

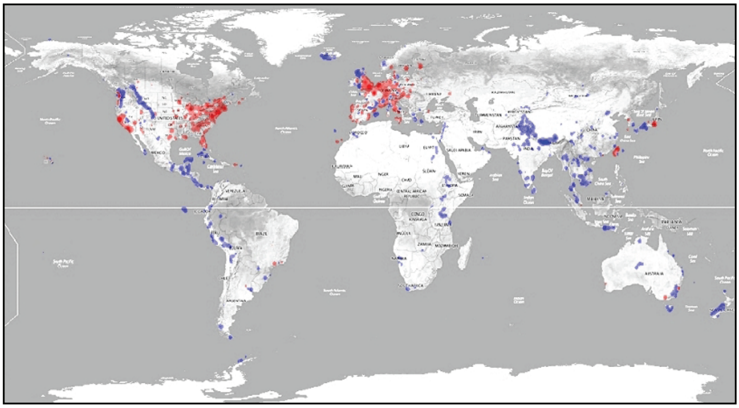


Figure 6.4: Predicted locations when Mnt. Everest is used as query. Blue: Positive predictions, Red: Negative predictions.

The optimal performance is reached with normalization per grid cell and a kernel σ of 20km.

As an example of the method performance, Figure 6.4 shows the recommended locations when Mnt. Everest is used as starting point. For visualisation purposes σ is set to 50km, a larger value than the optimum found in Figure 6.3a. Clearly the most populated areas like Europe and both coastlines of North America are unrecommended for mountaineers. Some mountain ranges are however clearly visible as positive recommendation: *Rocky Mountains*, *The Andes*, *Scottish Highlands*, *Mount Kilimanjaro* and of course the region around the query, the *Himalaya range*.

Next, we evaluate the prediction of beaches by finding the wormholes from *Platja de Lloret de Mar*, just north of Barcelona. We collect a list of 7216 beaches from Geonames[5] to evaluate the predicted locations. The results on beach prediction (Figure 6.3b) correspond to the results found on mountain prediction, N3 gives the optimal normalization and the optimal kernel width is around 20km.

Finally, we show that this method can be applied at multiple scales by predicting the wormholes from the famous cemetery *Père Lachaise* in Paris. Figure 6.3c shows the top-10 predicted locations. The highest ranked location not located close to L is found at *Cimetière du Montparnasse*, another big cemetery in Paris, demonstrating that this method can find similar locations also at city scale.

6.5 Conclusions

We have shown that geotags can effectively be used to predict similar locations with high precision. To limit the influence of individuals on the prediction only one geotag per grid cell should be considered per user. The kernel convolution method allows for detection of similar places at different scales, and can therefore be used for recommendations at global or city level.

[5] http://www.geonames.org/

7

Using Flickr Geotags to Predict User Travel Behaviour

We propose a method to predict a user's favourite locations in a city, based on his Flickr geotags in other cities. We define a similarity between the geotag distributions of two users based on a Gaussian kernel convolution. The geotags of the most similar users are then combined to rerank the popular locations in the target city personalised for this user.
We show that this method can give personalised travel recommendations for users with a clear preference for a specific type of landmark.

This chapter is published as: Maarten Clements, Pavel Serdyukov, Arjen P. de Vries, and Marcel J. T. Reinders. Using Flickr Geotags to Predict User Travel Behaviour. In *SIGIR10: Proceedings of the 33th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 851-852, New York, NY, USA, 2010. ACM.

7.1 Personalised Travel Guides

Before visiting a city, many people consult a travel guide or website that lists the most interesting locations. These travel guides are commonly based on the opinions of all other users. However, people have different preferences and therefore are not equally satisfied by these popularity rankings.

We propose to predict a user's favourite locations in a city based on his travel behaviour in previously visited cities. On social photo sharing websites like Flickr[1] people can annotate their photos, including the geographical location where the photo was made. Also, increasingly more cameras and smartphones are automatically storing the GPS coordinates when a photo is made. These *geotags* give an accurate indication of the user's preferred landmarks. Based on a set of collected geotags, we define a measure to identify similar users in previously visited cities. Then we aggregate these users' opinions in a different city to obtain a personalized travel recommendation for the target user.

The exploitation of geotags has shown to be effective for various tasks, like global event detection [104] and mapping textual tags to geographical locations [28]. Based on users' GPS tracks, location recommenders have been proposed that attempt to predict popular places and activities near the current location of the user [144; 128].

In this work we predict relevant locations based on users' geotags in a geographically remote location. We show statistical improvements over all users that visited the 10 largest cities and give an effective recommendation example based on an artificial user profile.

7.2 Flickr Geotags

Using the public Flickr API we have collected the geotags of 36,264 users, who actively use the geotag functionality. Together these users have uploaded 52,425,279 photos of which 22,710,496 have been geotagged.

We keep the data points that lie within the bounding boxes of the ten most visited cities. Based on our data, these cities in order of visitors are: *London*, *New York*, *Paris*, *San Fransisco*, *Los Angeles*, *Rome*, *Chicago*, *Washington*, *Barcelona*, *Berlin*. We only keep users who have made at least 5 photos in at least two cities. After this constraint the number of geotags of a single user in a city ranges from 5 to 5073 photos, and in total the 4750 remaining users have made 12,669 city visits. Together the users made 526,827 photos on the qualifying trips.

7.3 Methodology & Results

7.3.1 Baseline Ranking

As a baseline prediction we create a scale space representation of all the geotags in each city using a mean shift algorithm, similar to Crandall et al. [28], but using a Gaussian kernel instead of a uniform disc: $K(z) = e^{-z^2/2\sigma^2}$, where the standard

[1] http://www.flickr.com

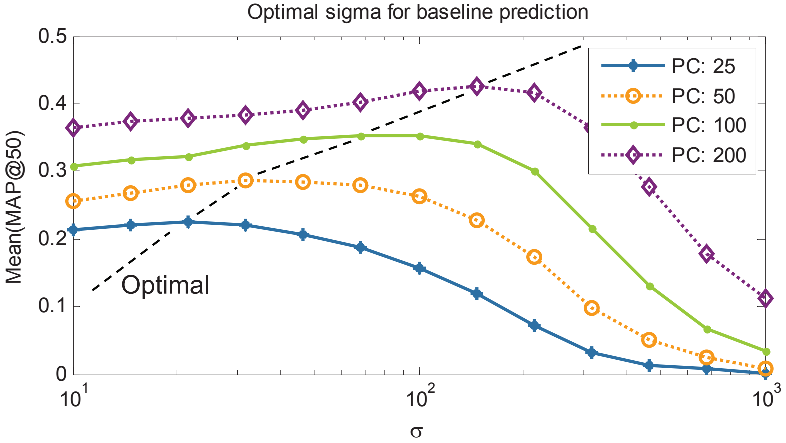


Figure 7.1: Mean MAP@50 of the baseline prediction in the top-10 cities for several positive cutoff values (PC) and increasing values of the kernel size (σ).

deviation σ is used as a scaling parameter. This method finds the maximum values of a kernel convolution of the distribution of all users' geotags with a Gaussian kernel (Φ_{All}). To ensure we reach all local maxima, we initiate the mean shift algorithm with all individual geotags. For each subsequent scale we use the peaks found in the previous scale to initiate the optimalisation procedure. The ranking based on the resulting peak weights gives us the top landmarks for each city, based on the general popularity.

To evaluate the ranking we judge a recommended location l_j as correct if the target user u_t has a geotag i within the positive cutoff value (PC) of that location, $\exists i : |l_j - u_t(i)| < PC$. Figure 7.1 gives the mean MAP@50 over the 10 cities, which computes the mean over the precision after each correct prediction in the top-50.

Figure 7.1 shows that the optimal σ is strongly dependent on the choice of PC. The predictions in this chapter will be evaluated at $PC = 100$ meter, which is roughly the radius of a landmark (e.g. the Colosseum is $189\,$m long). Based on the baseline results at $PC = 100$ we select $\sigma = 68\,$m for all further experiments.

7.3.2 Personalised Reranking

To personalise the landmark ranking for u_t in the target city (C_t), we compute the similarity between u_t and all other users u_c in the similarity city (C_s), where C_t and C_s are any two cities from the top-10, both visited by u_t. Using the mean shift algorithm we compute the peaks of u_t at $\sigma = 68m$ in C_s. For each peak k of u_t we now compute the value of the kernel convolution ($\Phi_{u_c}(k)$) on the geotags of u_c in C_s. The similarity between the two users is now derived by computing the sum over the minimum value in the two resulting profiles $S(u_t, u_c) = \sum_k min(\Phi_{u_t}(k), \Phi_{u_c}(k))$. As both profiles are normalised, this will give a similarity score in the range 0-1.

Based on all similar users we now rerank the top-50 popular locations l_j, predicted by the baseline method in C_t. This is done by recomputing the kernel con-

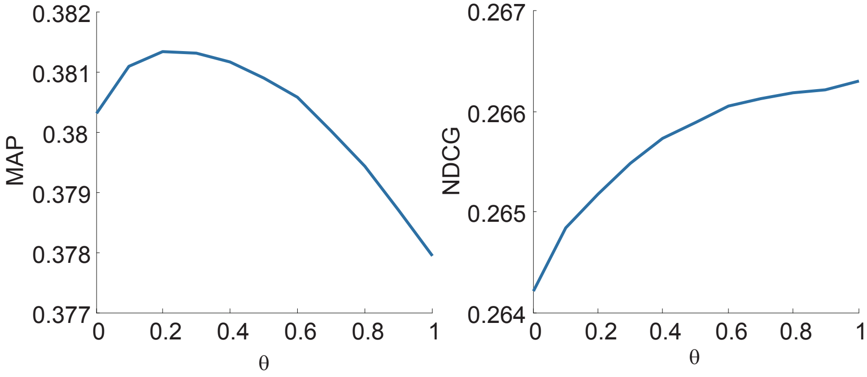


Figure 7.2: MAP@50 and NDCG@50 for increasing personalisation weight θ.

volution at these locations while weighing each user's geotags with his similarity to the target user: $\Phi_{Sim}(l_j) = \sum_{u_c} S(u_t, u_c)\Phi_{u_c}(l_j)$. The top-50 locations are now reranked by a linear combination of the baseline and the personalised score: $R(l_j) = (1-\theta)\Phi_{All}(l_j) + \theta\Phi_{Sim}(l_j)$.

Figure 7.2 gives the mean results over all users in the 90 possible combinations of two cities. The baseline is represented by the score at $\theta = 0$, where all user similarities are set to 1. Compared to the baseline, the optimal result on MAP improves 0.3%. At $\theta = 0.2$ there are 10,081 trips where we present an improved ranking to the user, against 8,440 trips where the baseline ranking would have been better. We also show the NDCG (refer to [62] for details) where the gain of each correct prediction is assigned as the inverse popularity of that location $(1/\Phi_{All}(l_j))$. The increase in NDCG shows that our recommender suggests less popular and therefore more serendipitous locations.

The improvement on MAP@50 is statistically significant in 22 out of 90 city pairs (based on a paired t-test with $p < 0.05$). For most users the improvement will however not make a big practical difference in the recommended locations. Compared to traditional collaborative filtering data sets, we find that many more people conform to the global popularity ranking if landmarks are concerned. For example, almost all people who visit Paris will make a photo of the Eiffel tower, while people who do not like Sci-Fi movies will never watch *Star Wars* even though it is one of the most highly ranked movies all times. This makes improving over the baseline a challenging task. Also, we observe many mixed preferences in user profiles (e.g. there are no users who *only* make photos at zoos), this makes it hard to match similar users.

As an example of the potential benefit of personalized travel recommendations, we created an artificial user profile with 10 geotags scattered around two modern/contemporary art landmarks in Barcelona (MACBA and Miro foundation). Table 7.1 shows a completely personalised ranking (with $\theta = 1$) and the rank difference between the baseline and the personalised ranking for modern art museums in other cities. It is clear that in all other cities where a modern art museum was in the top-50 we obtain a big rank improvement between the baseline and the predicted ranking.

Table 7.1: Query: *MACBA + Miro*

City	Rank	ΔRank	Landmark name
London	1	+3	Tate Modern
NY	1	+10	Guggenheim Museum
NY	3	+5	Museum of modern art
Paris	3	+4	Centre Pompidou
SF	3	+7	SF Museum of Modern Art
Chicago	2	+29	Museum of Contemporary Art
Washington	1	+20	Hirshhorn Museum
Berlin	7	+40	Hamburger Bahnhof Museum
Berlin	14	+16	Neue Nationalgalerie

7.4 Conclusions

A user's favourite landmarks in a previously unvisited city can be predicted by reranking the most popular locations based on users with similar travel preference. Our results indicate that statistical improvement over all users is hard to achieve, but for users with a clear travel preference very accurate predictions can be made.

8

Personalised Travel Recommendation based on Location Co-occurrence

We propose a new task of recommending touristic locations based on a user's visiting history in a geographically remote region. This can be used to plan a touristic visit to a new city or country, or by travel agencies to provide personalised travel deals to its customers.

A set of geotags is used to compute a location similarity model between two different regions. The similarity between two landmarks is derived from the number of users that have visited both places, using a Gaussian density estimation of the co-occurrence space of location visits to cluster related geotags. The standard deviation of the kernel can be used as a scale parameter that determines the size of the recommended landmarks.

A personalised recommendation based on the location similarity model is evaluated on city and country scale and is able to outperform a location ranking based on popularity. Especially when a tourist filter based on visit duration is enforced, the prediction can be accurately adapted to the preference of the user. An extensive evaluation based on manual annotations shows that more strict ranking methods like cosine similarity and a proposed RankDiff algorithm provide more serendipitous recommendations and are able to link similar locations on opposite sides of the world.

This chapter has been submitted as: Maarten Clements, Pavel Serdyukov, Arjen P. de Vries, and Marcel J. T. Reinders. Personalised Travel Recommendation based on Location Co-occurrence *ACM Transactions on Information Systems*. 2010.

8.1 Travel Recommendation

Location based services are quickly gaining popularity due to affordable mobile devices and ubiquitous Internet access. Websites like Foursquare[1], Gowalla[2], Google Latitude[3] and Facebook[4] show that people want to share their location information and get accurate location recommendations at any time and and place. In return for sharing their location data, users can now be matched to products, venues, events or local social relations and groups.

Accurate predictions of the user's preferred locations can simultaneously aid the user itself, advertisers of products specific to the recommended place and service providers (e.g. transportation to the recommended location). To provide these recommendations, the system needs to have an accurate way to find similarities between locations or people. We propose to exploit the past visiting behaviour of people to build a location similarity model that can be used for personalised location predictions.

In this work we will exploit a set of *geotags* collected from Flickr[5] to make a recommender that can predict relevant locations for individual users. In Flickr, geotags are tuples of latitude and longitude that represent the exact location where a user made a photo. Registration of geotags can be done manually by placing the photo on a map, or automatically by the device if it is equipped with a GPS module. Here we show that the collective knowledge represented in these geotags can be used to estimate similarities between locations and that personalised location recommendations can be derived from this similarity model.

Given a user's preference in one predefined area, we predict his activity in a another disjoint area. The proposed method will be evaluated on both city and country scale and will show that places on opposite sides of the world can be related based on user location histories.

8.2 Related Work

Since GPS equipped mobile phones have become mainstream, the amount of available geotags has grown to a number that allows for intensive data analysis. In this work, geotags are used to predict interesting locations for individual users, but the exploitation of geotags has shown to be effective for various other tasks. A method for global event detection has been proposed by Rattenbury et al., who searched for the occurrence of textual tags in spatial and temporal bursts [104]. Ahern et al. made a mapping of popular tags to geographical locations, resulting in a scale dependent map overlay with semantic information on the underlying data [1]. This work was extended by Kennedy et al. who selected relevant pictures for the predicted clusters [69]. Crandall et al. suggested not to use a fixed number of clusters and proposed

[1] http://foursquare.com/

[2] http://gowalla.com/

[3] http://www.google.com/latitude

[4] http://www.facebook.com/

[5] http://flickr.com/

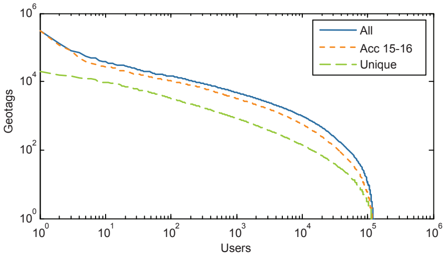
a mean shift algorithm to find the most prominent landmarks and representative photos [28].

Another application of Flickr's geotags was proposed by Lee et al. who used the geographical clusters related to a tag to improve the prediction of similar tags [74]. Furthermore, several methods have been proposed to predict the geotags of a photo, based on its textual tags [117], visual information [28] and individual user travel patterns [66].

As geotags relate to a location where the user made a photo, they inherently contain a touristic preference indication. Full GPS tracks are useful to study daily mobility patterns but extra effort is needed to extract touristically interesting spots. Based on users' GPS tracks, location recommender systems have been proposed that attempt to predict popular places and activities near the current location of the user. Some work has focused on the recommendation of specific types of locations. An item-based collaborative filtering method was used to recommend shops, similar to a user's previously visited shops [128] and a user-based collaborative filtering was proposed to generate restaurant recommendations through users with similar taste [57]. Zheng et al. extensively studied GPS tracks in Beijing, defined a method to extract interesting locations from this data (*Stay regions*) and proposed a matrix factorization method to suggest locations and activities based on the current state of the user [144]. They also showed that the HITS model can effectively be used to create a ranking of popular locations and experienced people [145].

Compared to most of the previously proposed methods, our system gives recommendations in a geographically remote location, so people can use it when they are planning a trip to another country or city. We have previously showed that geotags can be used to construct a measure of similarity between locations [27]. Here, we present a thorough extension of the previous work, using a similarity model based on a scale-space of location co-occurrence data. We evaluate the potential of this similarity model for personalised recommendations. The proposed model contains a scale parameter that allows the prediction of differently sized regions. So, when a user decides to visit a certain country the recommender can be used to find the most interesting cities and when a user gets to that city the same method can be used to find the most interesting landmarks, restaurants or other venues.

Many recommendation algorithms have been proposed based on similarities between objects in a discrete item-space [112; 133], which has proven to be effective in E-commerce applications [79]. Compared to these systems, a location recommender does not have a limited number of objects to recommend. Any point consisting of two continuous values of latitude and longitude can be recommended. On a more fundamental level, we introduce a model that includes the pairwise distances between points in order to reason in this continuous space. We will demonstrate the effectiveness of this model on geographical data, but it could easily be extended to include other continuous dimensions like temporal information.



	Users	Geotags	Mean	Med.
All	126123	42.9M	340	64
Acc 15-16	124860	26.4M	211	33
Unique	124860	7.2M	57	13

Figure 8.1: The distribution of the number of geotagged photos per user in descending order. The accuracy filter reduces the data set from 43M to 26M geotags. By selecting only unique geotags we maintain 7M points. The table also indicates the mean and median number of geotags per user.

8.3 Data

8.3.1 Data Collection

Using the public API of Flickr we have collected a large set of geotagged photos in a period of several months at the end of 2009 and early 2010. Figure 8.1 gives the distribution of the number of geotags per user (*All*). The distribution clearly shows that our crawl has a bias to people with many geotags, as the expected long tail of the distribution is missing. However, as we will only evaluate recommendations for users who have provided a sufficient amount of data, this bias in the crawl does not interfere with the objectives of this work. The total set corresponds to roughly 46% of the 93 million publicly available geotags in Flickr at the end of 2009[6].

Each geotag has an associated level of accuracy in the range of 1-16, 16 being the most accurate. This accuracy roughly relates to the zoom level of the map interface in Flickr. Because we want to make accurate predictions at the scale of individual landmarks, we keep only geotags at accuracy 15 or 16 (street level). The remaining data is represented by *Acc 15-16* in Figure 8.1. The possibility to integrate the accuracy value in the recommender system will be discussed in Section 8.8.

Flickr allows users to upload and annotate photos in *batches*. When someone uses this function it can either mean that he made many photos at that location, or that he did not take the effort to give the exact coordinates for each individual photo. Because of the uncertainty about the user's intent when uploading a batch to a single location, we choose to ignore the possible relation between user preference and batch size and store only one geotag per batch. After these filtering steps, we retain 7.2 million geotags contributed by 125 thousand users (*Unique* in Figure 8.1).

8.3.2 Data Statistics

The collected data set gives an interesting insight in the common behaviour of Flickr users. Besides the location of photos, Flickr also stores the date and time a photo was taken (according to the internal camera clock). Figure 8.2 shows the number of

[6] According to: http://www.flickr.com/map

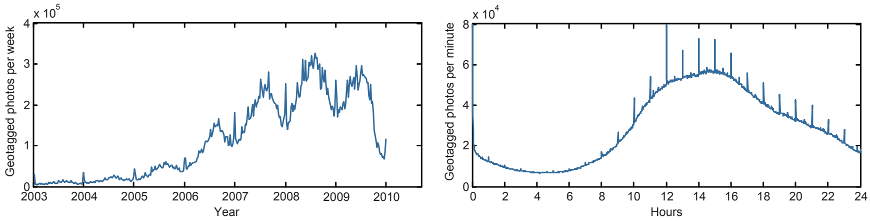


Figure 8.2: When Flickr users make photos. Left: Photo count per week from 2003 to 2010. Right: Photo count per minute of the day, aggregated over all days.

photos taken in a certain week between 2003 and 2010. Apart from the clear increase in popularity over the last 5 years it is interesting to see that most of the photos are taken during the northern hemisphere summer.

When we aggregate over all days and count the number of photos for each minute, we clearly see the bulk of photos is made late in the morning or early afternoon. In the evening the number of photos slowly decays until the minimum is reached around 4:30. The spikes at full hours and at January 1st in the weekly histogram are caused by default values of empty fields in Flickr's database.

Figure 8.3 gives the geographical distribution of the data. This 2000x4000 histogram of the geotags clearly shows the most popular travel areas in the Flickr community. Europe and North America have the largest density of data points, but the rest of the world is also recognisable. Figure 8.4 gives a closer view of North America, which shows that coastlines, cities and even highways are clearly represented in the data.

Based on this data, we select the 10 most popular countries and 10 most popular cities to evaluate the feasibility of personalised travel recommendation. We rank the

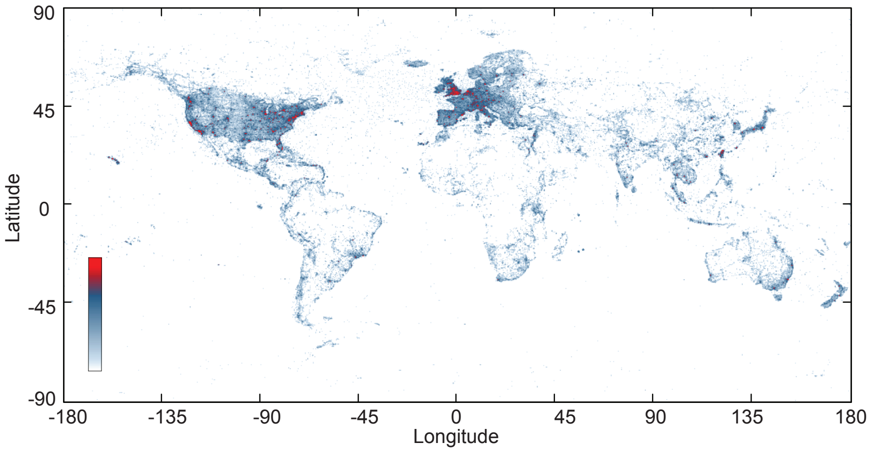


Figure 8.3: Where Flickr users make photos: World distribution.

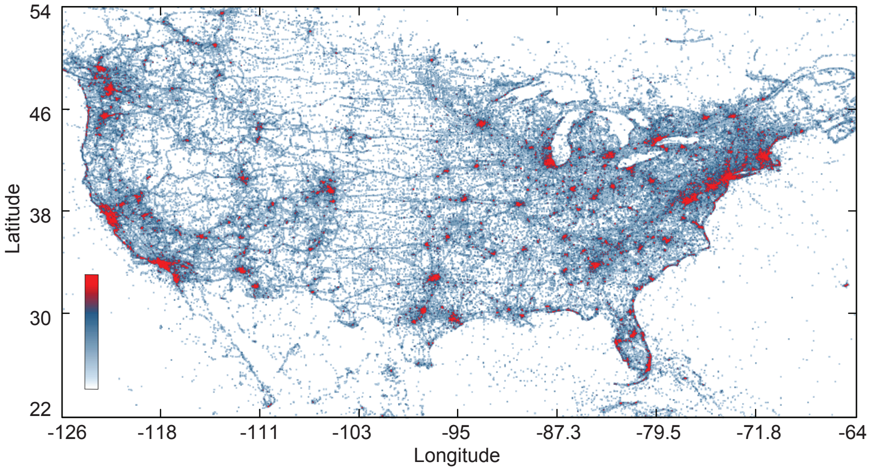


Figure 8.4: Where Flickr users make photos: USA distribution

Table 8.1: Number of users in top-10 cities and countries

Users	City	Users	Country
19802	London, England, United Kingdom	45738	United States EAST
18291	New York, NY, United States	32904	United States WEST
13786	Paris, Ile-de-France, France	25934	United Kingdom
12470	San Francisco, California, United States	18247	France
7893	Rome, Lazio, Italy	16995	Italy
7627	Los Angeles, California, United States	15414	Spain
7208	Washington, District of Columbia, United States	13381	Germany
7158	Chicago, Illinois, United States	11024	Canada
7069	Barcelona, Catalonia, Spain	6503	Netherlands
6569	Berlin, BE, Germany	5067	Australia

cities and countries by the number of users that have been there (Table 8.1), based on their geotags located within city bounding boxes[7] and country polygons[8]. Because the number of users in the USA is much larger than other countries, we split the USA in 3 regions: East USA (Longitude $> -98.583°$), West USA (Longitude $< -98.583°$), Alaska (Latitude $> 50°$).

8.4 Experimental Setup

Figure 8.5 presents the experimental setup and the notation described in the following sections is summarised in Table 8.2. The data is comprised of a set of users $u \in U$ who have all visited at least one location $l \in \mathcal{L}$, where l is a tuple (x, y, z) of Cartesian coordinates and $\mathcal{L} \subset \mathbb{R}^3$ is the set of all geotags in our data set. The set of geotags $\mathcal{L}$ is a subset of the world $\mathcal{W}$ described by a sphere with radius 6,367,449 m centered at

[7] Collected in January 2010 from http://developer.yahoo.com/geo/geoplanet/

[8] Collected in March 2010 from http://mappinghacks.com/

Table 8.2: Notation used in this chapter. For all l, $\mathcal{L}$, f, Φ, p, $\mathcal{P}$ we use the superscript $\ldots^{s/t}$ to refer to the region of the data ($\mathcal{R}^s$ or $\mathcal{R}^t$) and the subscript $\ldots_{u_k}$ if the data is based on a single user. The locations in the co-occurrence space (c, $\mathcal{C}$) can also contain the subscript $\ldots_{u_k}$, but no superscript.

$u_k \in U$	The users in the Flickr data set
$\mathcal{W}$	The world; subspace of $\mathbb{R}^3$
$\mathcal{R}^s$, $\mathcal{R}^t$	Starting region, target region; Subspaces of $\mathcal{W}$
$l \in \mathcal{L}$	All geotags in the data set, subset of $\mathcal{W}$
f	Function describing a set of geotags
Φ	Function describing the Gaussian convolution of f
$p \in \mathcal{P}$	The peaks of Φ
$c \in \mathcal{C}$	Points in the co-occurrence space; Subset of $\mathbb{R}^6$

zero. While Flickr provides the geotags in latitude and longitude we will use Cartesian coordinates throughout this work, which is more efficient for the computation of Euclidean distances between points. The distance between two points is measured through the crust of the Earth instead of over the surface. This difference is negligible for small distances and rank equal in general.

The data from half of the users (the *training* set) will be combined in a model that captures the similarities between the most important locations in two regions. With the data from the other half of the users (the *test* set) the application of the learned co-occurrence model for personalized travel recommendations will be evaluated. We split the data in equally sized training and test sets by first ranking all users according to the number of geotags. In this order, we select users $1, 4, 5, 8, 9, \ldots$ as training users and $2, 3, 6, 7, 10, \ldots$ as test users, so the two sets will roughly follow the same distribution.

The objective of this work is to predict the visited locations of a test user $u_k \in U$

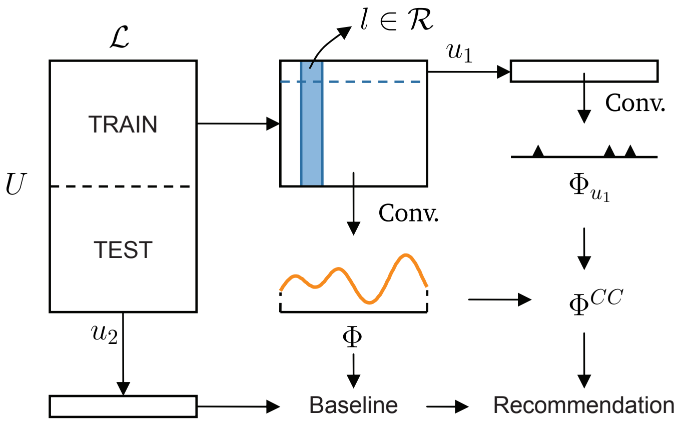


Figure 8.5: Experimental setup. The training users generate the global travel distribution Φ and the location similarity model Φ^{CC}. The performance of both models for location recommendation in a predefined region $\mathcal{R}$ is evaluated on the test users.

in a target region $\mathcal{R}^t \subset \mathcal{W}$, based on the geotags of that user in a starting region $\mathcal{R}^s \subset \mathcal{W}$. A *region* $\mathcal{R}$ can refer to either a city or a country from Table 8.1. To evaluate the performance of the location prediction we remove all the geotags of u_k that lie within $\mathcal{R}^t$ and use the geotags of u_k in an other region $\mathcal{R}^s$ to predict the location of the removed data. For this evaluation setup we need users that have visited at least 2 distinct regions. Obviously, when the recommender is operational, recommendations can already be made when a user has visited a single region.

To build the location similarity model between $\mathcal{R}^s$ and $\mathcal{R}^t$, we first find the most popular locations in these two regions. We use a kernel convolution of the training data with a Gaussian kernel to smoothly cluster the geotags that are near to each other (Section 8.5). We also find the most important locations per user by computing the kernel convolution over only the user's geotags. Both resulting distributions (Φ, Φ_{u_k}) are combined in the co-occurrence space Φ^{CC} which estimates the relations between the top locations in both regions (Section 8.6). The model Φ will be used to generate a baseline ranking (Section 8.7.1), the model Φ^{CC} will be used to predict a personalised location ranking per user (Section 8.7.2-8.7.3).

8.5 Peak Finding (Φ,Φ_{u_k})

The geotags of all users are described by the function f which has a Dirac delta pulse at the locations where one of the users created a geotag and zero otherwise:

$$f(z) = \sum_{l \in \mathcal{L}} \alpha_l \delta(z - l) \tag{8.1}$$

where α_l is a parameter that allows the assignment of different weights per geotag. In this work α_l will be set to 1 for all l, other weighting strategies will be discussed in Section 8.8.

We propose to use a Gaussian kernel convolution to obtain a smooth estimate of the density of all photos on the planet $\Phi_\sigma = f * g_\sigma$, where the Gaussian kernel is described by $g_\sigma(z) = e^{-\|z\|^2/2\sigma^2}$, for $z \in \mathbb{R}^3$. The standard deviation σ is used as a scaling parameter (or bandwidth) which gives the opportunity to set the size of the recommended locations. We do not use the common normalisation parameter of a probability density estimation with Gaussian kernels ($1/n\sqrt{2\pi\sigma^2}$, with n the number of data points) so that the convolution result will directly estimates the total number of photos taken at a certain location instead of the probability. In the rest of this work, we will drop the subscript σ for readability.

In the same way the function describing the geotag profile of a single user u_k is given by:

$$f_{u_k}(z) = \sum_{l \in \mathcal{L}_{u_k}} \alpha_l \delta(z - l) \tag{8.2}$$

And the density estimate $\Phi_{u_k} = f_{u_k} * g$.

We use $\mathcal{P}$ and $\mathcal{P}_{u_k}$ to denote the local maxima or *peaks* of Φ and Φ_{u_k} respectively. These peaks represent the most popular locations for all or a single user. A *mean shift* procedure is used to efficiently find the peaks of the functions [20]. We evaluate the peaks at 19 values of σ evenly distributed on a logarithmic scale from 10 m to 10 km

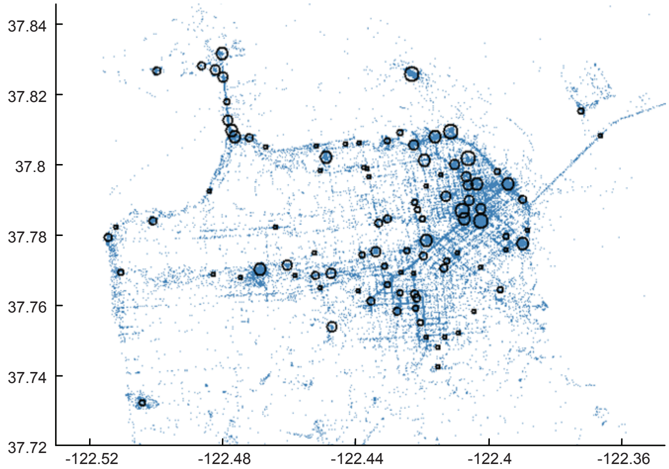


Figure 8.6: The circles indicate the top-100 peaks in San-Francisco at $\sigma = 100$ m where the radius is related to the peak amplitudes. The underlying data points clearly show the structure of the touristic part of the city.

for cities and 1 km to 1000 km on country scale. To ensure that all local maxima are found, we initiate the mean shift procedure with all individual geotags for computation on the finest scale. On each subsequent scale σ, we use the peaks from the previous scale as seeds. This procedure results in a scale-space that represents the structure of the data and allows us to analyse it at various scales.

The peaks $p \in \mathcal{P}$, found by the mean shift procedure on all geotags, can now be ranked based on their amplitude to obtain a popularity ranking of the locations in region $\mathcal{R}$ at scale σ. The application of the mean shift algorithm on geotag data was already proposed by Crandall et al. Compared to their work our scale-space will be more accurate because we use Cartesian coordinates instead of mapping latitude and longitude in a 2D plane [28]. Also, our method differs from Crandall et al. as we use a Gaussian kernel instead of a uniform disk. The Gaussian kernel convolution results in a smooth density estimate and does not generate plateau peaks. Other notable similar methods to define a popularity ranking of all locations in a given area are the scale specific clustering in Yahoo!'s World Explorer [1; 104] and the tree-based hierarchical graph used in Microsoft's GeoLife project [145]. We chose to use the Gaussian scale space as it has a strong theoretical foundation [78] and will show to provide a logical solution to the co-occurrence model.

In Figure 8.6 the data points of the training users in the city center of San Francisco are shown (the actual bounding box used in this work is larger). The top-100 peaks with largest amplitude at $\sigma = 100$ m are depicted by circles. The clustering shows that the proposed model does capture most of the well known landmarks like *Alcatraz, Union Square Park*, *Coit Tower*, *Yerba Buena Gardens* and *Pier 39*. Long stretched landmarks like the *Golden Gate Bridge*, are not represented by a single cluster but several clusters appear at the popular view points. Figure 8.7 shows the country

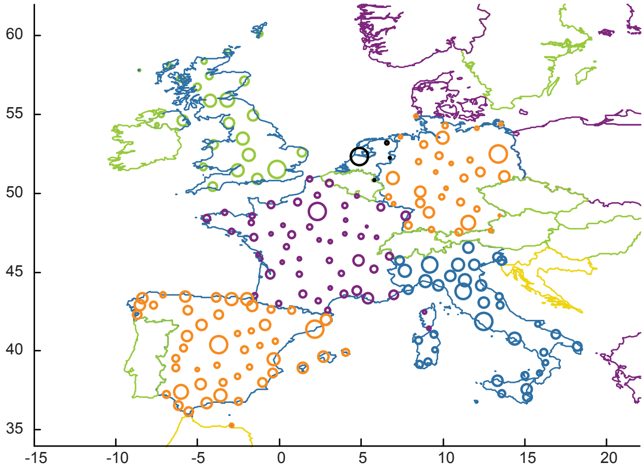


Figure 8.7: The polygons of the European countries in the top-10 most visited (Blue), top-20 (Green), top-30 (Purple) and top-40 (Yellow). The circles indicate the peaks in the top-10 most visited countries with $\sigma = 21.5$ km, the radius is related to the peak amplitudes.

polygons in western Europe and for the countries in the top-10 the clusters are shown at a scale of $\sigma = 21.5$ km. Most of the main cities are clearly visible on the map. The west of the Netherlands is grouped into a single cluster at this scale, which is reasonable as it is often seen as a single metropolitan area. At smaller scales the individual cities appear.

For computational efficiency we will only experiment with the top-500 peaks in each region. To check whether we are missing any important peaks in this step we look at the peak amplitude of the 500th peak in Figure 8.8. As the contribution of each geotag to a peak ranges between 0 and 1, the peak amplitude estimates the number of photos taken there. Because a user can make multiple photos at a single location, the number of users that contribute to the peak will be smaller: users < photos ≈ Peak amplitude.

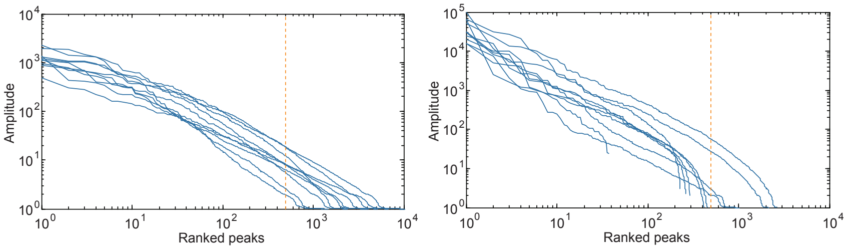


Figure 8.8: The distribution of peak amplitudes at the smallest scales that will be used for evaluation in cities and countries. Left: Each line shows the peak amplitudes in one of the top-10 cities at $\sigma = 46$ m. Right: Each line represents one of the top-10 countries at $\sigma = 6.8$ km. The dotted lines indicate the cutoff at 500 peaks.

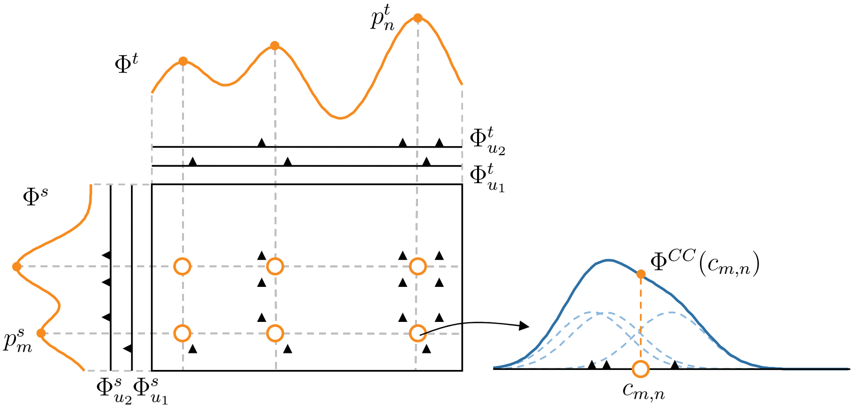


Figure 8.9: Co-occurrence model. Each user's peaks are mapped into the co-occurrence space (visualised for two users). At the Top-500 peak locations of the prior distribution Φ the result of the kernel convolution in the co-occurrence space Φ^{CC} is evaluated. For one point the computation of the contribution of both users is demonstrated. For visualisation purposes the 6D co-occurrence space is shown in 2D (left) and 1D (right).

The values chosen for σ will be explained in Section 8.7.1. At $\sigma = 46\,\text{m}$ there are only three cities where the 500th peak has an amplitude larger than 10 (London, New York, San Francisco). There are two countries (USA East and USA West) that still have large peaks after the top-500 (Amplitudes: 57 and 28). We believe that a cluster smaller than 10 photos is insignificant for our task and conclude that in most regions no important locations will be lost due to the selection of the top-500 peaks.

8.6 Co-occurrence Model (Φ^{CC})

When visiting a country or city, most users actively plan their trip and choose the landmarks to visit based on their interests. Especially, making a photo at a certain location is a clear indication of interest in that location. Based on these assumptions, we propose to estimate the similarity between two location by the number of users that have made a photo at *both* places. As geotags are continuous points in $\mathcal{W} \subset \mathbb{R}^3$, a method needs to be found that counts the contribution of each of these points to a pair of landmarks.

We propose to create the location co-occurrence model between two regions $\mathcal{R}^s$ and $\mathcal{R}^t$ as follows. At a chosen scale σ the locations visited by u_k are selected by taking his peaks $p^s_{u_k} \in \mathcal{P}^s_{u_k}$ from $\mathcal{R}^s$ and $p^t_{u_k} \in \mathcal{P}^t_{u_k}$ from $\mathcal{R}^t$. The location co-occurrences for this user between the two regions are given by $c_{u_k} \in \mathcal{C}_{u_k}$, where $\mathcal{C}_{u_k} = \left\{ \left\langle p^s_{u_k}, p^t_{u_k} \right\rangle | p^s_{u_k} \in \mathcal{P}^s_{u_k}, p^t_{u_k} \in \mathcal{P}^t_{u_k} \right\} \subset \mathbb{R}^6$ is the set of all pairwise combinations of this user's peaks in both regions. The points in the co-occurrence space are visualised for two users by the black triangles in Figure 8.9.

When all the peaks of all users are added to this co-occurrence space, the most dense regions represent location pairs that are often visited by the same users, and therefore indicate a strong similarity between the two locations. A smoothed prediction of location similarities can now be derived by computing the kernel convolution over the co-occurrence space, which will be denoted as Φ^{CC}. However, since this space may contain millions of 6 dimensional data points, applying the mean shift algorithm to find the local optima is computationally expensive.

However, the locations of the most prominent landmarks are already known from $\mathcal{P}^s$ and $\mathcal{P}^t$. Therefore we only need to evaluate the value of Φ^{CC} at the pairwise location combinations from $\mathcal{P}^s$ and $\mathcal{P}^t$, visualised as orange circles in Figure 8.9. For example, when p^s_m and p^t_n are two peaks in Φ^s and Φ^t respectively, and the combined location is given by $c_{m,n} = \langle p^s_m, p^t_n \rangle \in \mathbb{R}^6$, the co-occurrence of these two landmarks is defined by the sum over all user contributions:

$$\Phi^{CC}(c_{m,n}) = \sum_{u_k \in U} \sum_{c_{u_k} \in \mathcal{C}_{u_k}} e^{-d(c_{m,n}, c_{u_k})/2\sigma^2} \tag{8.3}$$

where $d(c_{m,n}, c_{u_k})$ is the Euclidean distance between the evaluated landmark combination $c_{m,n}$ and c_{u_k} is a location co-occurrence in the profile of u_k. As we have limited the number of peaks per region to 500 there will be maximally 250,000 evaluation points per combination of $\mathcal{R}^s$ and $\mathcal{R}^t$.

The upper left point in the co-occurrence space example in Figure 8.9 illustrates that peak intersections from Φ^s and Φ^t may exist that do not generate a peak in the co-occurrence space Φ^{CC}: if two locations are simply never visited by a single user, the co-occurrence will be zero.

We illustrate the computation of Φ^{CC} at the bottom right evaluation point in Figure 8.9. Three user points contribute significantly to the co-occurrence peak, although also the small contributions from the other peaks are taken into account. The illustration also indicates that the actual peak in the co-occurrence space might be slightly shifted to a different location. The impact of the error introduced by this approximation is discussed in Appendix A.

8.7 Results

8.7.1 Baseline Optimisation and Evaluation Criteria

As a baseline, the peaks in $\mathcal{R}^t$ will be ranked on the score determined by the general popularity: $S(p^t_n) = \Phi(p^t_n)$. This results in a static ranking, equal for all users. After ranking the locations, we compute the distance of each of the recommended locations to the nearest peak of the test user in $\mathcal{P}_{u_k}$ (at the same σ). We then set a threshold PC on this distance and count a recommended location as correct if the nearest of the user's peaks lies within this threshold. At small scale values, many peaks will be predicted close to each other. To make sure the recommender does not get rewarded for the suggestion of a single landmark multiple times, we disqualify a recommended location if it lies within distance PC from an earlier prediction.

The predicted location ranking will be evaluated on four criteria:

Precision (P@5), defined as the fraction of correct recommendations in the top-5.

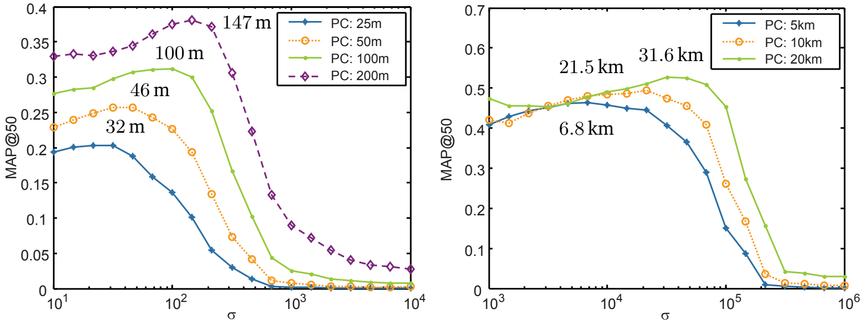


Figure 8.10: Performance of the baseline ranking using MAP@50. Left: Results on city scale, for the full range of σ and $PC \in \{25, 50, 100, 200\}$ m. Right: Results at country scale for $PC \in \{5, 10, 20\}$ km.

Mean average precision (MAP@50), the mean over the precision values after each correct recommendation in the top-50.

NDCG$_{\text{IP}}$. Similar to Zhou et al. we want to express the *surprisal value* of the recommended list in a number [147]. We propose to use the Normalised Discounted Cumulative Gain (NDCG) by Järvelin and Kekäläinen which compares the predicted ranking to the optimal possible ranking [62]. The NDCG allows the assignment of a *gain* value to account for differences in relevance between the ranked objects (please refer to [62] for details). To measure the surprisal value of the predicted ranking we set the gain of each correctly recommended location p_n^t to the inverse popularity $1/\Phi(p_n^t)$ abbreviated as IP, so that less popular locations contribute more to the result than popular locations. Then we compute NDCG$_{\text{IP}}$ over the resulting ranking. The optimal NDCG$_{\text{IP}}$ will be obtained when we correctly predict all the user's test locations, but in reverse order of popularity.

Benefit ratio (BR), the number of users who get an improved recommendation over the baseline divided by the number of users who get a deteriorated recommendation. BR can be computed over any of the previously defined evaluation methods.

To only evaluate users who have provided a decent amount of preference information, we consider those users who have at least 5 peaks at the lowest level of the scale-space ($|\mathcal{P}_{u_k}| \geq 5$). At city scale this pruning step means that users must have at least 5 peaks in Φ_{u_k} at $\sigma = 10$ m. At country scale, users need to have at least 5 peaks in Φ_{u_k} at $\sigma = 1$ km.

The optimal σ at a chosen value of PC will be estimated based on MAP@50. Compared to P@5, the results on MAP@50 more gradually change with different values of σ, therefore parameter optimisation on MAP@50 gives a more reliable estimate of the optimal setting. P@5 however gives a more intuitive evaluation on the practical usability of the recommender. We will therefore show the results on both criteria in the next sections.

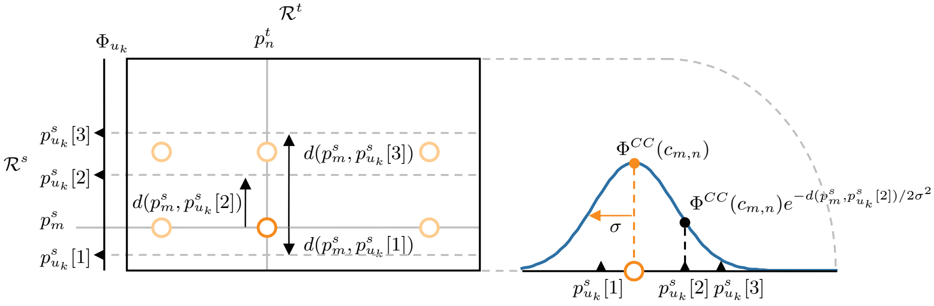


Figure 8.11: Computing recommendations with the location co-occurrence model. For each peak p^s_m in $\mathcal{R}^s$ all contributions of all the user's geotags are aggregated using a Gaussian distribution as weight function. Then the final score of a location p^t_n in $\mathcal{R}^t$ is derived from the sum over all p^s_m.

In Figure 8.10 the mean MAP@50 is plotted for the baseline ranking for the full range of σ values and various settings of PC. For all settings, the choice of σ has a clear optimum. When σ is chosen too small, multiple peaks exist at a single landmark, while for too large σ individual landmarks will be missed because they are merged into a single peak. At city scale the optimal σ is found close to the selected value of PC. At country scale we find that the optimal σ is larger. This can be explained by the fact that within a city the ratio between the point of interest size and the distance between them is larger than in a country.

At both city and country level, we select two scales for further evaluation. Within city recommendation will be evaluated at $PC = 50\,\text{m}$, $\sigma = 46\,\text{m}$ and $PC = 100\,\text{m}$, $\sigma = 100\,\text{m}$. At country scale we will evaluate recommendations at $PC = 5\,\text{km}$, $\sigma = 6.8\,\text{km}$ and $PC = 10\,\text{km}$, $\sigma = 21.5\,\text{km}$.

8.7.2 Recommendation

8.7.2.1 Generating Recommendations

We compute $\Phi^{CC}(\langle p^s_m, p^t_n\rangle)$ for all paired peaks in the top-500 $p^s_m \in \mathcal{P}^s$ and the top-500 $p^t_n \in \mathcal{P}^t$ in all combinations of $\mathcal{R}^s$ and $\mathcal{R}^t$ (the top-10 cities and countries), based on the set of training users. The derived models can now be used to generate recommendations for the test users.

As explained in Section 8.4 the geotags of test user u_k in a starting region $\mathcal{R}^s$ will be used to predict the visited locations in $\mathcal{R}^t$. The predicted location ranking in $\mathcal{R}^t$ will then be compared to the locations actually visited by u_k. In order to evaluate the performance of the predicted recommendations for a test user, the user therefore needs to have visited at least two distinct regions. In both regions we enforce the pruning settings at $|\mathcal{P}^s_{u_k}| \geq 5 \wedge |\mathcal{P}^t_{u_k}| \geq 5$ as explained in Section 8.7.1.

The score of location p^t_n in $\mathcal{R}^t$ for user u_k is now derived by:

$$S^{CC}(p^t_n, u_k) = \sum_{p^s_m \in \mathcal{P}^s} \sum_{p^s_{u_k} \in \mathcal{P}^s_{u_k}} \Phi^{CC}(\langle p^s_m, p^t_n\rangle) e^{-d(p^s_m, p^s_{u_k})/2\sigma^2} \tag{8.4}$$

Table 8.3: Results of the baseline (S) compared to the recommender (S^{CC}), for two scales at both city and country level.

	City				Country			
	$PC = 50\,\mathrm{m}$		$PC = 100\,\mathrm{m}$		$PC = 5\,\mathrm{km}$		$PC = 10\,\mathrm{km}$	
	$\sigma = 46\,\mathrm{m}$		$\sigma = 100\,\mathrm{m}$		$\sigma = 6.8\,\mathrm{km}$		$\sigma = 21.5\,\mathrm{km}$	
	S	S^{CC}	S	S^{CC}	S	S^{CC}	S	S^{CC}
P@5	0.237	0.237	0.293	0.300	0.266	0.274	0.257	0.261
MAP@50	0.311	0.312	0.370	0.377	0.437	0.445	0.482	0.488
$\mathrm{NDCG_{IP}}$	0.237	0.238	0.272	0.277	0.287	0.293	0.358	0.365
BR-P@5	1.034		1.375		1.419		1.337	
BR-MAP@50	1.046		1.246		1.248		1.298	
BR-$\mathrm{NDCG_{IP}}$	1.108		1.361		1.292		1.476	

which counts the contribution of each of the user's peaks $p^s_{u_k}$ in $\mathcal{R}^s$ to each of the landmarks p^s_m in $\mathcal{R}^s$, and weights each of these landmarks with the co-occurrence model. To predict the recommendations for u_k when traveling to $\mathcal{R}^t$, the locations p^t_n are ranked according to this score and the top ranked locations are recommended. This computation is visualised for a user u_k with three geotags in $\mathcal{R}^s$ in Figure 8.11.

8.7.2.2 Recommendation Performance

We now compare the ranking on S to the ranking predicted by S^{CC}. Table 8.3 contains the results at the two selected scales for between-city and between-country recommendation. The presented values are averaged over all possible recommendations for all city/country pairs in the top-10 lists. At city scale the results are based on 16,620 measurements, with an average user size of 9 locations (median). At country scale we can evaluate 13,476 recommendations, with a median user size of 7.

For all settings and all evaluation methods our model improves over the baseline. We test the significance of the improvement using a Wilcoxon signed rank test, which tests the hypothesis that the difference between the matched samples in the two sets comes from a distribution with zero median. At a confidence level of 1% only the results on P@5 for $\sigma = 46\,\mathrm{m}$ are not significant. Probably too many landmarks will be represented by multiple peaks at this scale, making the co-occurrence model less accurate.

The improved results on $\mathrm{NDCG_{IP}}$ indicate that not only the rank position of the test results improves, but also the surprisal value of the presented recommendations. The co-occurrence model gives better performance while less popular locations are observed at the top of the ranking. This shows that the method correctly learns how the preference of the user differs from the average.

Although BR shows a decent improvement when the recommendation model is used, the mean absolute improvement on the individual evaluation criteria is small. For many users the popularity based baseline and the personalised ranking of recommended locations are very similar. Two reasons can be given for these small differences. First, many users do not have a single preference (e.g. only visit botanical gardens), but visit many types of landmarks when they come to a new location. With the proposed co-occurrence model, the combined recommendations based on these mixed preference profiles converge to the prior ranking. Second, because many peo-

Table 8.4: Results of city scale recommendation at $\sigma = 100\,\text{m}$ for different tourist filters. The best results are obtained for the most strict filter (1x14).

	City, $PC = 100\,\text{m}$, $\sigma = 100\,\text{m}$							
	All		3x14		2x14		1x14	
	S	S^{CC}	S	S^{CC}	S	S^{CC}	S	S^{CC}
P@5	0.293	0.300	0.321	0.330	0.331	0.339	0.339	0.351
MAP@50	0.370	0.377	0.409	0.417	0.419	0.427	0.430	0.440
NDCG_{IP}	0.272	0.277	0.301	0.308	0.307	0.314	0.318	0.325
BR-P@5	1.375		1.511		1.491		1.687	
BR-MAP@50	1.246		1.338		1.358		1.422	
$\text{BR-NDCG}_{\text{IP}}$	1.361		1.491		1.547		1.614	
Recs	16,620		8576		6600		3536	

ple visit the most popular locations in the target region the evaluation method actually expects us to recommend these. This is inherent to the evaluation of recommendations with a train and test set.

In Section 8.7.3 we will see that when a single type of landmark is used as starting location and we manually asses the recommended locations, the prediction is highly accurate and we can use more extreme weighting methods to exploit the location co-occurrence.

8.7.2.3 Tourist Filter

We hypothesize that people who visit both $\mathcal{R}^s$ and $\mathcal{R}^t$ for touristic purposes will benefit more from the recommendations than people who actually live in one of the cities. To confirm this hypothesis we implement a tourist filter as follows: Based on the creation date of the photos in the Flickr data a user qualifies as tourist in a certain city if all his photos in that city are taken in n periods of 14 days. So in the 3x14 filter we allow the user to visit a single city 3 times, and all the user's photos have to be taken in at most 3 different windows of 14 days.

The results with three different tourist filters applied in both $\mathcal{R}^s$ and $\mathcal{R}^t$ are presented for $\sigma = 100\,\text{m}$ in Table 8.4. First, we observe that both the baseline and the recommendation performance go up when a more stringent filter is used. So tourists conform more to the overall visiting behaviour than city inhabitants. Second, when we set a more strict tourist filter, the performance difference between the recommender and the baseline goes up. This means that touristic behaviour in one city should be predicted by touristic behaviour in another city.

Table 8.4 also indicates the number of recommendations (*Recs*) that can be evaluated with each filter. We need a user two have made a touristic visit in at least two different cities in order to evaluate the performance. These two criteria cause the number of evaluations to drop quite quickly.

8.7.2.4 Within-City Recommendation

Song et al. showed that the every day mobility patterns of people are highly predictable 93% of the time [121]. Other related work on location prediction also focused on making recommendations close to the current location of a user [57; 128;

144]. We suspect that prediction of touristic behaviour in previously unvisited areas is a much harder task. First, touristic behaviour is less predictable than every day life behaviour. Second, remote predictions allow many more possibilities than nearby recommendations.

To test whether we can use our model for within-city recommendations we compute the co-occurrence space Φ^{CC} within each city ($\mathcal{R}^t = \mathcal{R}^s$) and set the self co-occurrence of each location to 0. For each user u_k in the test set that has been to $\mathcal{R}^t$, we cut off the last day of photos made in that city. We use the geotags created by u_k on all previous days as starting points and try to predict the user's behaviour on the final day of his stay. To split the user's data in days we use the creation date and time of the photos shifted backwards by 4.5 hours based on the results in Figure 8.2.

Table 8.5 gives the results at $\sigma = 100\,\text{m}$ averaged over all users (*No pruning*), and limited to users who have at least 5 peaks in $\mathcal{P}^t_{u_k}$ at this scale in both the test day and the training days. The absolute evaluation scores are lower than the scores reported in between-city recommendation, because we have fewer evaluation points in this setup. After pruning, the median user has 6 points on the test day, compared to a median of 9 in city to city recommendation.

The relative improvement with the personalised model is much larger for within-city recommendation than that for between-city recommendation. Especially for users with many geotags on the training and test day the personalised prediction clearly outperforms the baseline. Unfortunately, only few users (*Recs*) have provided enough data to pass the pruning settings. These findings indicate that adapting the location prediction to a user's personal interest is easier if the user stays within the same city.

We assume that the reason for this improvement is that users have a bias to make many photos within a certain area (e.g. close to the hotel). To verify this, we plot the probability density function (PDF) of the distance between two randomly selected geotags and the PDF of the distance between a geotag selected from the training days and a geotag selected from the test day of a single user (Figure 8.12). The dotted lines indicate the median of both distributions. Clearly the geotags selected from a single user have a larger probability to be close together. This location prior explains why recommendations within a single region are easier to predict than between two remote locations, confirming the second intuition given above, that remote locations allow more possibilities than nearby ones.

8.7.2.5 Conclusions

Because many users visit the same popular locations, prediction according to the prior travel probability is hard to improve upon. Although the absolute improvement is small, the co-occurrence model can give improved recommendations for most users.

Tourists can be selected by setting a maximum value on the number of days spent on a certain location. We find that tourists comply more with the general travel preference and are therefore more easy to predict by the baseline. Also, the relative improvement of the personalised model over the baseline is larger than for the average user, which shows that tourists have a clear preference that relates their behaviour in different cities. This shows that the location co-occurrence model based on the travel history of tourists can effectively be used to predict personalised travel recommenda-

	City, $\sigma = 100\,\mathrm{m}$, $PC = 100\,\mathrm{m}$			
	No pruning		$\lvert\mathcal{P}^t_{u_k}\rvert \geq 5$	
	S	S^{CC}	S	S^{CC}
P@5	0.042	0.047	0.108	0.129
MAP@50	0.099	0.119	0.197	0.244
NDCG$_{\mathrm{IP}}$	0.126	0.141	0.208	0.234
BR-P@5	2.452		5.182	
BR-MAP@50	1.966		2.690	
BR-NDCG$_{\mathrm{IP}}$	1.982		2.531	
Recs	18,344		896	

Table 8.5: Results on recommendation of the locations for the last day of a city visit.

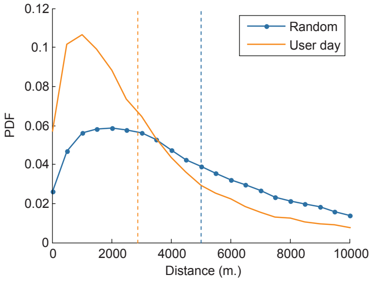


Figure 8.12: PDF of distance between two *random* geotags and between the last and previous days of a single user (*user day*).

tions. We have used a simple tourist filter and suggest that more elaborate methods could be used based on the users' profile information or textual tags.

Within-city recommendations are easier because the training data contains a location prior. If we know where the user was in the past few days, he is more likely to be in the same place the next day.

8.7.3 Serendipitous Ranking

8.7.3.1 Ranking Criteria

Using part of the users' real data points as test set, we have evaluated whether we can predict where the user will go if he is not influenced by a recommender. This evaluation is however strongly biased by the most popular locations in the target area. As most people will visit the *Eiffel Tower* when they get to Paris, it pays off to predict this with the recommender. However, the user would benefit more from a recommendation of a location that is not obvious and perhaps even unknown to the user. Related work on recommender systems has therefore argued that manual judgement of the recommended items is necessary for the evaluation of novel recommendations [19].

To test whether the proposed co-occurrence model can be used to produce serendipitous recommendations, we have manually annotated various sets of landmarks at city and country scale. We first use one of the landmarks (p^s_m) in $\mathcal{R}^s$ as starting point and try to predict the annotated landmarks (p^t_n) that fall in the same category in $\mathcal{R}^t$, using the following known ranking criteria:

Prior (S) Ranking based on $\Phi(p^t_n)$.

Direct (S^{CC}) As the user profile now consists of only a single peak from Φ in $\mathcal{R}^s$, Equation 8.4 reduces to a ranking based directly on $\Phi^{CC}(c_{m,n})$.

Cosine (CS) Ranking based on $\Phi^{CC}(c_{m,n})/\sqrt{\Phi(p^s_m)\Phi(p^t_n)}$. Cosine similarity corrects for the popularity bias by dividing the co-occurrence by the popularity of both individual landmarks.

We also propose a new ranking method, which assigns the prior amplitudes (Φ)

Table 8.6: Baseball stadium set. The prior rank is the rank index based on S in the corresponding region.

Stadium	City	Prior Rank	Longitude	Latitude
Yankee Stadium	New York	27	40.8271	-73.9281
City Field	New York	44	40.7557	-73.8481
Richmond Co. Bank Ballpark	New York	151	40.6457	-74.0761
AT&T Park	San Francisco	13	37.7785	-122.3896
Dodger Stadium	Los Angeles	12	34.0735	-118.2400
Nationals Park	Washington	22	38.8729	-77.0076
Wrigley Field	Chicago	5	41.9479	-87.6558
Cellular Field	Chicago	18	41.8300	-87.6340

Table 8.7: Modern art museum set. The prior rank is the rank index based on S in the corresponding region.

Museum	City	Prior Rank	Longitude	Latitude
Tate Modern	London	4	51.5081	-0.0990
Museum of Modern Art	New York	5	40.7610	-73.9771
Guggenheim Museum	New York	12	40.7831	-73.9591
Centre Pompidou	Paris	6	48.8604	2.3520
Hirshhorn Museum	Washington	10	38.8888	-77.0230
MACBA	Barcelona	7	41.3832	2.1668
Fundacio Miro	Barcelona	28	41.3686	2.1597
Neue Nationalgalerie	Berlin	23	52.5070	13.3681
Haus der Kulturen der Welt	Berlin	21	52.5187	13.3648
Hamburger Bahnhof Museum	Berlin	28	52.5283	13.3719

as weight to all locations and then compares the weight difference between the initial and new ranking:

> **RankDiff (RD)** Let R_1 be the rank index (position in the ranked list) of a location based on $\Phi(p_n^t)$ and R_2 the rank index of the same location in $\Phi^{CC}(c_{m,n})$. Let Ψ be the list of peak amplitudes of Φ ranked in descending order. RankDiff is now defined as $RD(p_n^t) = \Psi(R_2) - \Psi(R_1)$.

The rationale behind this method is that a location that used to be at rank R_1 and had an amplitude of $\Phi(p_n^t)$ managed to reach a new ranking of R_2 where a location with amplitude $\Phi(p_x^t)$ used to be. The difference between these two amplitudes can now be seen as the amount of evidence needed to accomplish this rank gain.

Note that we also considered other ranking algorithms, that performed worse or very similar to any of the above (i.e. *Jaccard coefficient*, *Pointwise Mutual Information (PMI)*, *Lift* [129; 96]); the results of these ranking criteria are therefore left out of the discussion.

8.7.3.2 City Scale

We manually annotate a set of baseball stadiums (Table 8.6) and a set of modern or contemporary art venues (Table 8.7) in the top-10 cities. We now repeatedly select one of the cities as $\mathcal{R}^t$ and rank all landmarks in that region based on one landmark in $\mathcal{R}^s$. As evaluation we compute the number of times a target location (from one

Table 8.8: Results on baseball stadium and modern art prediction. Mark that the number of test locations in all cities is small, therefore the maximum possible P@5 equals 0.30 for baseball stadiums and 0.32 for modern art museums.

	Baseball					Modern Art				
Method	Up	Down	P@5	R@5	P@R	Up	Down	P@5	R@5	P@R
S	0	0	0.04	0.09	0	0	0	0.07	0.26	0
S^{CC}	45	3	0.16	0.58	0.29	53	18	0.10	0.41	0.19
CS	41	7	0.15	0.47	0.24	30	49	0.07	0.27	0.06
RD	44	4	0.23	0.76	0.46	39	38	0.12	0.43	0.25

of the two sets) goes *up* or *down* in the ranking compared to a ranking based on S, the precision at 5 (P@5), recall at 5 (R@5), defined as the fraction of correct results ranked in the top-5 and precision at R (P@R), where R is the total number of correct locations that can be recommended. For all evaluations the peaks in Φ at $\sigma = 100\,\mathrm{m}$ are used, since at this scale it is easy to manually relate each peak to a single landmark.

The results in Table 8.8 show that almost all baseball stadiums are related to each other as 45 out of 48 times a stadium gets a higher ranking based on co-occurrence than on the prior (48 is the total number of possible ways to select two landmarks from different cities). A ranking directly based on S^{CC} does get the target locations higher in the list, but the more popular locations are often still at the very top of the ranking, resulting in a limited P@5, R@5 and P@R. The other methods make more mistakes on up/down, but RankDiff clearly improves precision and recall. The P@R of 0.46 indicates that RankDiff gets the target stadium(s) to the very top of the ranking in about half of the cases, which is a remarkable achievement given the relatively low prior rank of the stadiums.

Further inspection of the ranking produced by cosine similarity shows that many very small peaks are ranked at the top. Cosine similarity can easily generate a high score when an unfamiliar starting location is used, if by coincidence the users who have been there have also another location in common. RankDiff is somewhat more conservative as it is more dependent on the absolute value of Φ^{CC} than the relative difference between Φ^{CC} and Φ.

Although the modern art data set appears to be less coherent, the order of the methods is similar. Because many of the venues already have a high prior ranking it is hard to improve the prediction. RankDiff again gives the best performance on precision and recall.

To study the benefit of having more profile information from a user, Figure 8.13 shows P@R and R@5 for the three personalised methods on both data sets while the number of starting locations is increased. When two starting locations are located in different cities we simply sum the Φ^{CC} values before computing the ranking criteria. The results are averaged over all target cities and all possible combinations of N landmarks selected from the other cities.

When the recommendations from more starting points are aggregated the prediction generally gets better. The prediction of baseball stadiums based on RankDiff even reaches a R@5 of 1, meaning that in all cases the target locations are ranked in the

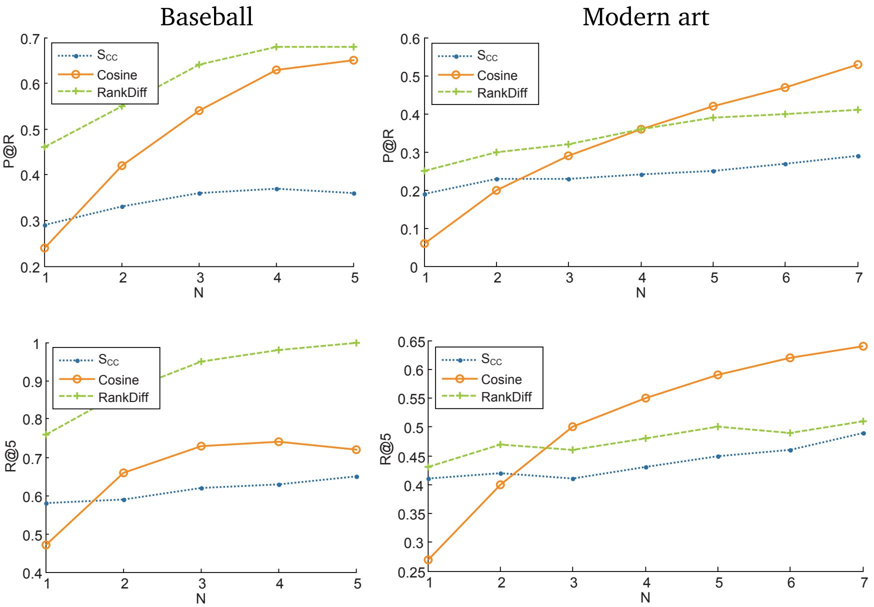


Figure 8.13: Performance difference of S^{CC}, Cosine and RankDiff with increasing profile information (N). The two left panels show the P@R and R@5 for Baseball recommendation, the right panels for Modern art.

top-5. If more information is present, Cosine similarity is less prone to mistakes and shows a steep upward trend in performance.

8.7.3.3 Country Scale

To evaluate the co-occurrence model at country scale, we manually annotate a large set of the peaks in USA West at $\sigma = 21.5$ km and select various starting locations in other countries to see how they influence the ranking in USA West.

Based on the prior ranking (not shown) the top-10 of locations in USA West contain 9 cities and only 1 national park (*Yosemite NP*). If we use *Ayers Rock* in Australia as starting point we expect recommendations that refer more to natural locations and less to cities. A ranking directly based on S^{CC} does show that some natural parks increase their ranking, but the co-occurrence with the top-4 cities is still larger, simply because their prior visit probability is larger (see Table 8.9).

We find that especially cosine similarity returns very interesting recommendations. Figure 8.14 and Table 8.9 show that almost all places in the top-10 refer to rock formations in the USA, which is quite amazing since absolutely no semantic information (like textual tags) is used in the prediction.

In this example, cosine similarity seems to give better results than RankDiff. On this scale there are hardly any obscure peaks, therefore we can take the risk of using a method that can get small peaks very high in the ranking, and cosine similarity is able to get peaks from the lower part of the ranking to the top. This introduces

Table 8.9: Top-10 recommendations based on Ayers Rock, Australia. R is the new ranking, PR is the prior ranking (based on Φ).

	S^{CC}		Cosine		Rankdiff	
R	PR	Location	PR	Location	PR	Location
1	1	San Fransisco	129	Painted Hills SP	4	Las Vegas
2	4	Las Vegas	122	Craters of the Moon NM	32	Bryce Canyon NP
3	3	Los Angeles	44	Monument Valley SP	44	Monument Valley SP
4	2	Seattle	99	Idaho Falls	36	Mt. Rushmore NM
5	32	Bryce Canyon NP	32	Bryce Canyon NP	13	Lake Tahoe
6	44	Monument Valley SP	36	Mt. Rushmore NM	14	Grand Canyon NP
7	5	Portland	62	Mt. Shasta	17	Maui
8	36	Mt. Rushmore	49	Crater lake	49	Crater lake NP
9	13	Lake Tahoe	141	Roswell	62	Mt. Shasta
10	14	Grand Canyon NP	153	Socorro / Box Canyon	122	Craters of the Moon NM

more risk in the recommender, but can also give more interesting and serendipitous recommendations.

8.7.3.4 Conclusions

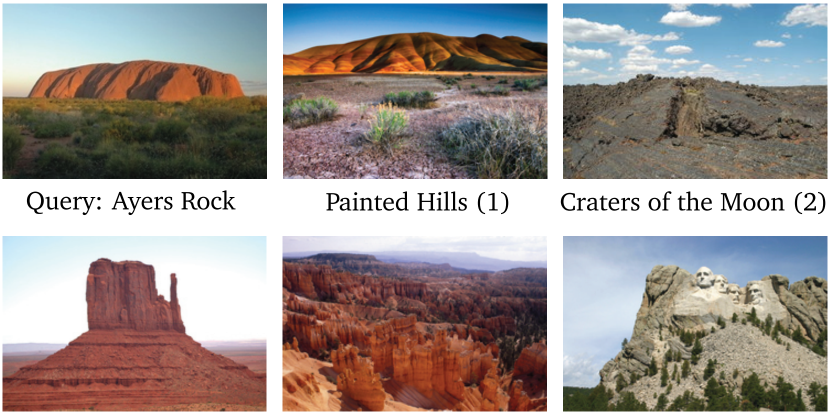
When the co-occurrence model is used to generate a location ranking based on a single preference point, we observe great performance increase over the prior ranking. A ranking based on S^{CC} directly does get the correct locations higher in the list, but not to the very top of the ranking. We find that more extreme weighting methods can be used to fully exploit the co-occurrence model.

Cosine similarity can give very small peaks as recommendations when the co-occurrence happens to be relatively large compared to the prior visiting probability. The Ayers rock example showed that this can give very interesting results. Using solely the location history of Flickr users, we were able to relate rock formations on completely opposite sides of the world.

When limited information is available the risk of recommending something unknown is high when cosine similarity is used. The proposed method *RankDiff* is more conservative, the results are more reliable but may be less surprising. On a manually annotated set of baseball stadiums we showed that the RankDiff method is able to perfectly predict where a stadium in an unvisited city is located if several other stadiums are used as starting points.

8.8 Conclusions and Discussion

We have proposed to approximate the Gaussian kernel convolution over the co-occurrence space of Flickr geotags to obtain a location similarity model. This new approach to predict recommendations in a continuous object space can effectively be used to recommend locations matching a user's preference. Recommendations can be made close to the location of the user, so that we can suggest landmarks for the next day on a city visit. More interesting, the co-occurrence model can be used to make recommendations in a previously unvisited city or country which is useful while planning a holiday. The bandwidth of the Gaussian kernel controls the size of the target locations, which



Query: Ayers Rock

Painted Hills (1)

Craters of the Moon (2)

Monument Valley (3)

Bryce Canyon (5)

Mount Rushmore (6)

Figure 8.14: When Ayers Rock in Australia is used as query, the top recommendations in USA West contain many famous rock formations.

allows application at a scale of choice (city and country level in this work). The results suggest that recommendations based on the co-occurrence model are both more accurate and more surprising than a ranking based on the prior travel probability. A simple filter to distinguish inhabitants from tourists indicates that touristic behaviour is more informative for the prediction of a user's behaviour in another city.

In this work we have set the weight of all geotags equal, but the proposed model can deal with differently valued data points. We discussed the choice to ignore the number of photos in batch uploads, but a weighting method could be proposed to integrate this information in the amplitude of the data point. Furthermore, the importance of a photo could be estimated on external information sources like the textual tags or the interestingness ranking used by Flickr.

By filtering the set of geotags on the *accuracy* value in the Flickr database we have selected only geotags that are accurate on street level, thereby losing about 40% of the original data. One could argue whether this accuracy filter is necessary if predictions are made on a larger scale (e.g. between-country recommendation). The function that describes a set of geotags is in this work defined as a collection of Dirac delta pulses. To integrate the geotag accuracy into this function, it naturally follows that each geotag could itself be described by a Gaussian distribution, where the standard deviation is dependent on the accuracy. In this way inaccurate geotags do not influence predictions on small scale, but do contribute on larger scales.

Recommendation evaluation with a training and test set has a drawback. Because of the strongly skewed prior travel distribution most of the locations in a user's test set are well-known popular places. These places will dominate the parameter optimisation of the model, resulting in a personalised model that does not differ much from the prior ranking. The popular locations are however not the most interesting places to recommend, because the user is probably already familiar with them or can easily find them in regular travel guides.

To really evaluate whether a recommender gives interesting, user specific recommendations, manual assessments are inevitable. Using manually annotated locations on both city and country scale we have shown that more strict ranking methods can be used to produce more serendipitous recommendations. A ranking based on cosine similarity can give very interesting and novel recommendations, but also has the possibility of recommending something irrelevant based on data noise. The proposed RankDiff method is more conservative but gives stable good recommendations in all experiments. Based on these results we can assume that these weighting methods will also be more effective in a recommendation system, when the full user profile is used as training data.

A Appendix: Full 6D Kernel Convolution

As indicated in the model description in Section 8.6, the computation of the co-occurrence model at the prior peak locations is an approximation of the real peaks in the co-occurrence model. To estimate the error introduced by this approximation we have used the mean-shift algorithm to compute the peaks of the full Gaussian kernel convolution on the 6D co-occurrence space for the city pair Berlin-Barcelona at $\sigma = 100\,\mathrm{m}$.

We compare the top-50 similarity relations generated by both methods in the co-occurrence space between Berlin and Barcelona. Using manual evaluation, we find that 44 out of 50 relations uniquely refer to the same landmarks. The median distance of the top-50 peaks in our approximation to the nearest peak in the full convolution is $26\,\mathrm{m}$. The measured peak amplitude at the landmark locations will always be smaller than the nearest peak in the full convolution. We find that the average decay in peak amplitude in the approximation is -2.4%.

The small differences between both models show that the approximation proposed in this work can effectively be used to predict the most co-occurring locations between two cities.

Part IV

Semantics

9

Deriving Term Specificity from Social Tagging Data

We address the task of determining the semantic specificity of terms from a social tagging corpus. We propose that the specificity of a term is strongly related to how well it represents the user's information need and demonstrate that the applicability extends beyond the tag domain and can improve effective collection access by means of search or interactive browsing.

To quantify the pairwise relationship between two tags we identify three cases: the tags are equally specific, one is more specific, or they are incomparable. Both specificity and similarity measures are combined in a machine learned approach to classify tag pairs into these classes.

Using a large tagging corpus, we compare existing and new methods to relate statistical term specificity with human assessments. We find that several proposed methods are not strongly correlated and can thus be combined in a single classifier. Using only three features we can decrease the specificity prediction error by 27.4% over a prediction based on document frequency. We demonstrate a new specificity metric that outperforms previously proposed metrics, because it takes both tags into account while computing their relation. Further, we investigate features to detect equally specific tags, and demonstrate that similarity measures that take the tag context into account can improve the prediction of related tags over traditional co-occurrence methods.

This Chapter is based on collaborative work with Yahoo! Research Barcelona: Maarten Clements, Börkur Sigurbjörnsson, Vanessa Murdock and Roelof van Zwol. Deriving Term Specificity from Social Tagging Data. *Yahoo! Research Barcelona*. 2009.

9.1 Introduction

Term specificity was related to *inverse document frequency* (IDF) by Sprck-Jones, who discussed in 1972 that a query term which occurs in many documents is not a good discriminator to satisfy the information need of the searcher [124]. She proposed to weight the terms in a query by $f(N) - f(n) + 1$ where N is the size of the collection, n is the occurrence of the term and f is a function which computes $f(n) = m$ such that $2^{m-1} < n \leq 2^m$. This heuristically proposed method has since then been extensively studied and forms the basis of many search applications [110; 107].

Sprck-Jones proposed that term specificity in an information retrieval system should be treated statistically instead of semantically [124]. In this chapter we compare statistical methods that represent semantic specificity.

The semantic specificity of a term is dependent on factors other than the occurrence in the collection alone. A specific term may occur frequently if the term happens to be popular in the community that created the collection. For example, we could consider the terms *html* and *knitting* equally specific as both describe a method of constructing objects (either web pages or clothes). However, commercial search engines return about 100 times more documents on HTML than on knitting. At the same time the Flickr[1] search engine returns about 100 times as many photos tagged with *knitting* than photos tagged with *html*. This shows that the document frequency of a term is domain dependent, and thus cannot be the sole source to derive term specificity.

With various tasks in mind different methods have been proposed to estimate semantic term hierarchy. Document frequency and term entropy have been used as an aid to create hierarchies like WordNet [36; 15; 111; 108; 65]. At the same time, the ‘clarity score’ has been proposed to estimate the query difficulty [31; 50] and vocabulary growth and entropy have been studied extensively in social media [17; 21; 83].

Joho and Sanderson recently studied the relation between document frequency and semantic specificity [65]. They compared the WordNet structure to the document frequency inferred from the Google search engine and TREC corpus[2]. When they limit their test to words that co-occur in the same documents they find an overlap between the WordNet hypernym structure and *document frequency* in different corpora to be in the range of 81.4%-84.3%.

The goal of this work is to develop a better method to define the semantic specificity relationship between two terms and to evaluate whether it can improve effective collection access by means of search or interactive browsing. We propose a framework in which we classify each pair of terms into any of the classes: More/Less specific (>,<), Equally specific (==) or Incomparable (!=). We discuss the individual performance of new and previously proposed specificity and similarity features to distinguish between these classes and evaluate which features can be combined to improve the classification performance. Using 3112 manually assessed tag pairs and the Flickr tagging corpus, we directly optimize a linear support vector machine (SVM) classifier to the semantic classification of human judges.

[1] http://flickr.com/

[2] http://trec.nist.gov

Table 9.1: Notation used in this chapter.

T	The set of all tags
t_a	tag a
I	the set of all images
i_a	image a
U	the set of all users
u_a	user a
$I(t_a)$	the set of images that are tagged with t_a
$U(t_a)$	the set of users who used t_a
$T(i_a)$	the set of tags assigned to i_a

In tagged social media the document frequency of a term is not strongly related to its specificity. While annotating pictures, users seek a trade off between specificity and generality. Using general terms as content descriptors will lead to low retrieval precision.Thus general terms are not necessarily more frequent in social tagging systems (e.g. In Flickr *animal* occurs less frequently than *dog*).

As Flickr's tagging system does not aggregate the number of times a certain tag has been added to a photo, the term frequency (*tf*) is always 0 or 1. Therefore the common ranking function $w_i = tf_i \times \log(D/df_i)$ depends solely on the global term weight (*df*). This puts a lot of pressure on the importance estimation of the query terms.

To summarize, the contributions of this work are as follows. Using a linear classifier we improve the performance of specificity and similarity detection over previously known methods. We propose a method (sub-super) that outperforms previous specificity metrics. The method takes the relation between two terms into account, whereas previous metrics are computed over each term individually. We show how the optimal classifier can be used to create a network of tags with pair-wise specificity relationships and we discuss how this work can be used in a browsing interface or directly applied to weight the terms in a search query. Finally, we show that similarity measures that take the context of tags into account can be used to separate tags that frequently co-occur from truly comparable tag pairs.

We discuss methods to estimate term specificity in Section 9.2 and term similarity in Section 9.3. Related work is interleaved in these two sections, as it relates to each metric. The variables used in these sections are listed in Table 9.1. In Section 9.4 we describe the experimental setup, including the data, the manual specificity assessments, and the inter-assessor agreement study. In Section 9.5 we present the classifier and the experimental results.

9.2 Specificity

The notion of semantic specificity is hard to define. Here we base our work on the Merriam-Webster definition that specific is *free from ambiguity*[3]. We see specificity as a gradual concept, so based on this definition a more specific tag will be less ambiguous. If a tag has little ambiguity it is more likely that it will be attached to photos with

[3] http://www.merriam-webster.com/dictionary/specific

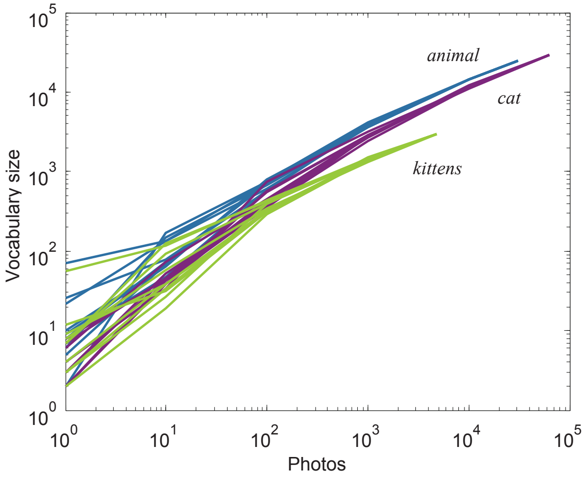


Figure 9.1: The growth rate of the vocabulary size conditioned on three different tags. The vocabulary size is the number of unique tags attached to a random set of photos that contain the target tag. Each tag is represented by 10 lines based on different permutations of the images that contain the tag.

similar content. Photos that show similar content will probably be annotated with a similar set of tags. In other words, there is more coherence in the annotations given to these photos. Therefore most of the specificity methods that we will present are based on the idea that *more specific tags co-occur with a more coherent tag set*.

9.2.1 Document Frequency (DF)

Document frequency has been used as a measure of statistical specificity, although it has always been recognised that there is a discrepancy between document frequency and semantic specificity [124]. This notion of specificity has proven to be useful in information retrieval to give less weight to common terms [124; 110; 107].

We represent the *document frequency* of a tag by the probability that a random image contains the tag:

$$DF(t_a) = P(t_a) = |I(t_a)|/|I| \tag{9.1}$$

9.2.2 Vocabulary Growth (VC)

The vocabulary size of a collection is defined as the number of unique terms occurring in it. There have been many studies of vocabulary size both in text domain and social tagging systems. Heaps' law states that as more documents are added to the collection, there will be diminishing returns in terms of discovery of the full vocabulary from which the distinct terms are drawn [4].

Cattuto et al. found that the vocabulary growth in tagging systems follows a powerlaw with an exponent around 0.8, even when conditioned on a single resource [17]. Marlow et al. looked at the vocabulary growth within individual user libraries in

Flickr [83]. Golder also showed that there is large variation between the vocabulary growth of individual users by studying the delicious network [44].

Vocabulary growth has been studied conditioned on certain users or documents, but not conditioned on a single word. Intuitively the size of the vocabulary related to a single term is a measure of specificity, because terms that describe a very specific concept always co-occur with a similar set of other terms. Figure 9.1 shows the speed of vocabulary growth conditioned on three different tags. For these three tags the growth rate of the vocabulary is clearly related to the specificity of the tag, while the document frequency (total Nr. of photos) is not.

The full tag vocabulary in Flickr shows a sublinear growth closely following a power law with exponent 0.7. The exponent of the 3 tags in Figure 9.1 ranges between 0.4 and 0.5. We find that the estimation of the power-law exponent is too unstable to use as specificity measure. We therefore use the absolute vocabulary size of 1000 randomly sampled images that contain the target tag. To be independent of temporal variations in global tagging behaviour in Flickr, we compute the vocabulary size on 10 random subsets of 1000 images and use the mean as specificity feature.

9.2.3 Entropy (EN)

We look at the entropy of all tags co-occurring with a given tag t_a. A high entropy of co-occurring tags suggests that the given tag is a broad concept and adding the term to a query would not make the query much more informative.

$$EN(t_a) = -\sum_{w \in T} P(w|t_a) \log P(w|t_a) \tag{9.2}$$

where the probability of observing a tag w given t_a is based on the number of images annotated with both tags:

$$P(w|t_a) = \frac{|I(t_a, w)|}{|I(t_a)|} \tag{9.3}$$

In textual documents, Caraballo and Charniak used the entropy of the surrounding text to identify the specificity of nouns [15]. Using this method they could predict around 80% of the hypernym relations in WordNet. In social media, Chi and Mytkowicz looked at the tag entropy evolution compared to the document entropy evolution. They conclude that the document entropy increases faster which will eventually lead to decreased retrieval performance if a social network continues to grow [21].

9.2.4 Simplified Clarity Score (CL)

The clarity score is defined as the relative entropy (or KL-Divergence [71]) between a language model based on the query and the corresponding collection language model [31; 50]. Simplified to a tag consisting of a single term t_a:

$$CL(t_a) = \sum_{w \in T} P_T(w|t_a) \log \frac{P_T(w|t_a)}{P_T(w)} \tag{9.4}$$

where $P_T(w) = \sum_{i \in I} |T(i)|/|T|$ and the probability of observing a tag w given t_a is:

$$P_T(w|t_a) = \sum_{i \in I} P_T(w|i) P_T(i|t_a) \tag{9.5}$$

For coherency with related work the probabilities are computed over the event space of all individual tag assignments instead of images as in the other features.

Similar to [125] we set the probability of observing a photo given the tag equal for all photos that contain that tag: $P_T(i|t_a) = 1/|I(t_a)|$.

As proposed by He and Ounis [51], we use the maximum likelihood approximation of $P_T(w|i)$. Because photos in Flickr do not aggregate the tags from different users, but maintain a single list, the term frequency is always one, so $P_T(w|i) = 1/|T(i)|$.

Intuitively, if the language model generated by a tag is very similar to the collection model, the tag does not add any information and is thus not useful to query the database. It has been shown that the clarity score can be used to estimate the difficulty of a query for a retrieval system [31; 50]. The query difficulty is directly related to the specificity of the query, as a more specific query is easier to answer.

9.2.5 Sub-Super (SS)

This method assumes that if two tags can be ordered by specificity, the subsets of the specific tag will also be subsets of the general tag, but not the other way around. For example, all important subsets of the tag *paris* (e.g. *louvre*, *eiffel*, *notredame*) also co-occur with the tag *france*, but not all subsets of *france* co-occur with *paris* (e.g. *toulouse*, *bordeaux*, *lyon*).

To exploit this relation we compute the sub-super relation between the two tags $SS(t_a||t_b)$ by:

```
SS(t_a||t_b) = 0;
subs = GetSubs(t_a);
for t_x in subs:
    supers = GetSupers(t_x);
    if t_b in supers:
        SS(t_a||t_b)++;
```

where `GetSubs(t_a)` gets the 100 tags t_x that give the highest $P(t_a|t_x)$, with the limitation that $|U(t_x)| > 0.01 \times |U(t_a)|$. This limitation prevents the selection of very rare terms. `GetSupers(t_a)` gets the top-10 tags t_x with the largest probability conditioned on t_a: $P(t_x|t_a)$.

The example in Figure 9.2 shows the top `Subs` and $SS(t_a||t_b)$ of the tag $t_a =$ *barcelona*. The top-10 `Subs` are clearly subsets of Barcelona (mainly prominent landmarks designed by Antoni Gaudí). The top-10 $SS(t_a||t_b)$ contain the most prominent super sets of the `Subs`. For example, *spain* is a super set of 80 out of 100 `Subs` of *barcelona*, indicating that *spain* is probably more general than *barcelona*. The tag *gaudi* also has a relatively high score ($SS(barcelona||gaudi)$ = 32), the reversed relation is however still higher ($SS(gaudi\ ||barcelona)$ = 76), indicating that the evidence of *gaudi* < *barcelona* is stronger than the reversed relation.

Instead of computing a global specificity score for each individual tag and deriving the relation from the difference between these scores, the sub-super method takes both input tags into account when computing their relation. Because of the ambiguity

Tag t_a: *barcelona*

Subs: *larambla*, *lasagradafamilia*, *casamilà*, *casamila*, *casabatlló*, *sagradafamilia*, *torreagbar*, *casabatllo*, *parcguell*, *parcgüell*

Sub-Super: [*spain*, 80], [*catalunya*, 60], [*gaudi*, 32], [*españa*, 30], [*catalonia*, 28], [*architecture*, 27], [*2007*, 19], [*europe*, 19], [*sagradafamilia*, 14], [*espagne*, 11]

Figure 9.2: Example of the top-10 subsets of the tag *barcelona* (Subs) and the top-10 tags with highest $SS(t_a||t_x)$ plus their occurrence count (Sub-Supers). Mark that Flickr concatenates multiple word tags.

of tags we believe that a method that considers both tags in the computation of their relation should outperform global specificity methods.

9.3 Similarity

The semantic specificity ordering of terms only makes sense if the terms can be compared in the same domain. Besides the relative specificity difference we define term similarity features to quantify the relatedness of the terms.

We define two sets of methods to determine the similarity between two tags. *Co-occurrence methods* use direct occurrence frequency statistics, and include the joint probability, the cosine similarity, and the jaccard coefficient. *Context methods* find the similarity between the context in which both tags occur. We represent the context as the set of tags that co-occur with the target tag. We present three context methods: Ranked list similarity, KL-divergence, and Sub-Super Sum.

9.3.1 Co-occurrence Methods

Tag co-occurrence is commonly used to derive a similarity metric between two tags [8; 118; 119]. Several similarity functions can be derived from co-occurrence:

Joint probability (JP) The probability that two tags co-occur in the annotation of a randomly drawn photo:

$$P(t_a, t_b) = \frac{|I(t_a) \cap I(t_b)|}{|I|} \tag{9.6}$$

Cosine similarity (CS) The cosine of the angle between two vectors:

$$CS(t_b, t_a) = cos(\theta) = \frac{I(t_b) \cdot I(t_a)}{\|I(t_a)\| \, \|I(t_b)\|} \tag{9.7}$$

Jaccard coefficient (JC) The co-occurrence probability of two tags, normalized by the union of both individual occurrence probabilities:

$$JC(t_b, t_a) = \frac{|t_a \cap t_b|}{|t_a \cup t_b|} \tag{9.8}$$

9.3.2 Ranked List Similarity (RLS)

Similarity between the top-k most similar tags of t_a and t_b. Motivated by Fagin et al. [35], we compute the similarity between two ranked lists by taking the mean over the intersection of both lists at different cutoffs. Let τ_a^k and τ_b^k be two partially ranked lists of tags, where $\tau(i) \in T$ and k is the length of the lists. We define the RLS as:

$$RLS(\tau_a^k, \tau_b^k) = \frac{\sum_{i=1}^{k} |\tau_a^i \cap \tau_b^i|/i}{k} \tag{9.9}$$

We compute the RLS on three different top-100 lists:

RLSCP τ_a^k consists of the k tags with largest conditional probability $P(t_x|t_a)$, with $k = 100$.

RLSCPR τ_a^k consists of the k tags with largest conditional probability $P(t_a|t_x)$, with $k = 100$ and $|U(t_x)| > 0.01 \times |U(t_a)|$.

RLSCS τ_a^k consists of the k tags with largest cosine similarity $CS(t_x, t_a)$, with $k = 100$.

9.3.3 KL-divergence (KLD)

The KL-Divergence between two probability distributions P and Q of a discrete random variable is commonly defined as:

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{9.10}$$

To obtain a symmetrical distance measure, we use the sum of both asymmetric computations, as proposed by Kullback and Leibler [71]:

$$D_{KL}(P, Q) = D_{KL}(P||Q) + D_{KL}(Q||P) \tag{9.11}$$

We compute the KL-Divergence on the top-100 conditional probabilities $P(t_x|t_a)$ of both input tags. We create the probability distributions by first inserting an epsilon value of 10^{-8} where t_x occurs in the top-100 of t_a but not in the top-100 of t_b (or vise versa) and then normalize the distributions to sum to one. Weinberger et al. used the KL-Divergence to find the tags in Flickr that give the optimal disambiguation of the query [138].

9.3.4 Sub-Super Sum (SSS)

The Sub-Super method is the only specificity method that takes both tags into account. Therefore it contains both a specificity component and a relatedness component. If two tags are very similar, they will also have similar subsets, in this case the sub-super relation will be high, independent of the order of the tags.

We use the sum of both sub-super relations as similarity measure:

$$SSS(t_a, t_b) = SS(t_a||t_b) + SS(t_b||t_a) \tag{9.12}$$

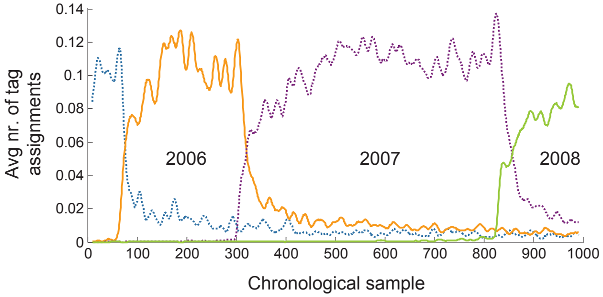


Figure 9.3: The occurrence frequency of the tags *2005*, *2006*, *2007* and *2008*. Each point shows the mean frequency over 10,000 photos sampled at 1000 equally spaced points in the chronologically ordered dataset.

9.4 Experimental Setup

For the analysis in this chapter we use a large collection of Flickr photo annotations. The collection contains tag annotations of publicly available photos uploaded to Flickr any time before early 2008. In order to give an indication of the scale of the collection, it contains annotations of hundreds of millions of photos (10^8), billions of tag assignments (10^9), and millions of unique tag strings (10^6).

Photo annotations are an appropriate corpus for our purpose since they are known to contain various types of tags. Naaman et al. identify 3 main types of photo annotations: place, activity, and depictions [91]. Overell et al. provide a more fine grained analysis of Flickr tag classes [98]. They report that about 20% refer to locations; 15% to artifacts or objects; 16% to people or organization, 4% to actions and events; and 7% to time. As an example of the last class Figure 9.3 shows the occurrence of the year tags in our dataset.

9.4.1 Sampling of Tag Pairs

This work attempts to establish the nature of the relation between two tags. For this task, pairs of tags are only useful if there is at least some sort of relation between them. All tags that have a low co-occurrence can be assumed to be unrelated as they are never used to describe the same concept. Therefore, we only select pairs of tags that have a high co-occurrence.

We select tag pairs by first drawing a random tag t_a from the full distribution of tag occurrences, giving a higher probability to frequently occurring tags. For this tag, we now draw a tag t_b randomly from the top-10 probabilities conditioned on t_a: $P(t_b|t_a)$. We also draw a tag t_c from the top-10 joint probabilities $P(t_a, t_c)$ and t_d from the top-10 tags that give the highest probability of drawing t_a: $P(t_a|t_d)$.

In the top of $P(t_a|t_d)$ many obscure tags occur, because if a tag t_d is used only once in the entire collection and it happens to co-occur with t_a the probability $P(t_a|t_d)$ will be one. We ensure that t_d is known to many users by selecting only tags that satisfy the criterion $|U(t_d)| > 0.01 \times |U(t_a)|$.

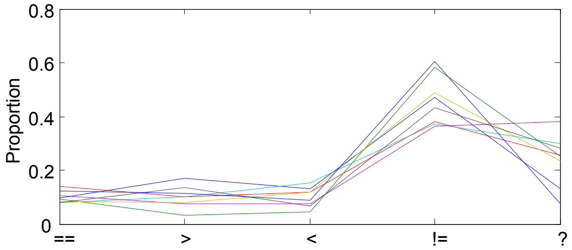


Figure 9.4: Assessment class distribution for all 8 human judges.

The tag pairs $\{t_a, t_b\}$,$\{t_a, t_c\}$,$\{t_a, t_d\}$ are added to our test collection. This procedure is repeated 1200 times, resulting in 3600 tag pairs. After removal of doubles and non Latin tags we retain 3112 tag pairs for our evaluation.

9.4.2 Manual Assessments

To classify the selected tag pairs we have used an interface that allowed users to assign the tags to any of the classes:

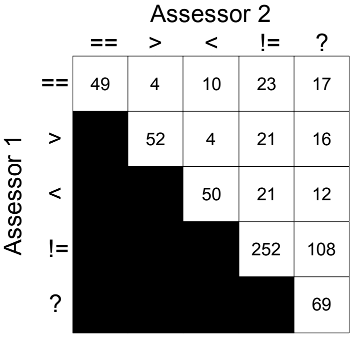
- **C1**: A $==$ B, A is equally specific as B
- **C2**: A $>$ B, A is more general than B
- **C3**: A $<$ B, A is more specific than B
- **C4**: A $!=$ B, A is incomparable to B
- **C5**: A ? B, I cannot judge these tags

For each assessment we presented a random tag pair in random order to the assessor and asked the user to assign it to any of the 5 classes. We explicitly asked the assessors to only classify a tag pair into one of the classes **C1-3** if the tags can be ordered in the same domain. For example, although *sushi* might be a more specific concept than *japan* there is no clear hierarchical specificity relation between them, whereas *sushi* and *food* can be unambiguously ranked. With this interface we have collected assessments from eight different human judges for all 3112 tag pairs. The assessments are distributed over the classes as follows:

C1	**C2**	**C3**	**C4**	**C5**
326	346	319	1559	562

Clearly, most of the pairs are assigned to the class *incomparable* (**C4**), although they all have a high co-occurrence. Many of these tags are composite tags like *las* and *vegas*, others just do not have a clear specificity relation, like *color* and *art*. Figure 9.4 shows the class distribution per assessor, where the mean Pearson correlation between the assessor distributions is 0.87. Section 9.4.3 will give a more elaborate discussion on the agreement between the eight assessors.

In all further experiments we neglect the pairs in **C5**.


Assessor 2
Assessor 1

	==	>	<	!=	?
==	49	4	10	23	17
>		52	4	21	16
<			50	21	12
!=				252	108
?					69

Figure 9.5: The agreement between two assessors on the 708 doubly assessed tag pairs.

9.4.3 Assessor Agreement

To compute the agreement between our assessors we have collected double assessments for 708 of the tag pairs. For each of these tag pairs we asked a randomly chosen assessor, different from the initial assessor, to classify the tag pair. We find that the average agreement between assessors is 0.83.

The confusion between classes can be seen in Figure 9.5. When we ignore **C5**, clearly most of the disagreement occurs when one of the assessors does not consider the two tags to be semantically related (and therefore classifies it into **C4**) and the other assessor does. This judgement was often difficult because the selection process caused all tag pairs to be related in a certain way.

The four confusions between **C2** and **C3** are:

2008 vs. *feb*
crater vs. *volcano*
venice vs. *carnevale*
2006 vs. *november*

These confusions can be explained as {*feb*, *crater*, *carnevale*, *november*} occurs in {*2008*, *volcano*, *venice*, *2006*}, but also in other places/times; Therefore, both tags can be seen as the most specific one. This type of confusion corresponds to the examples {*halloween*, *costumes*} and {*wedding*, *ceremony*} we will see in Figure 9.7.

The confusion between **C1** and either **C2** or **C3** mostly arises with singular vs. plural tags. Some assessors have labeled the plural tags as ‘more general’ while others have considered them equally specific.

9.5 Results

We use the implementation of a linear SVM classifier by Joachims [64]. For all classifiers the presented results are based on 10-fold cross validation, where 90% of the tag pairs are used as training data and 10% as test data. All features are normalized by subtracting the mean and dividing by the variance.

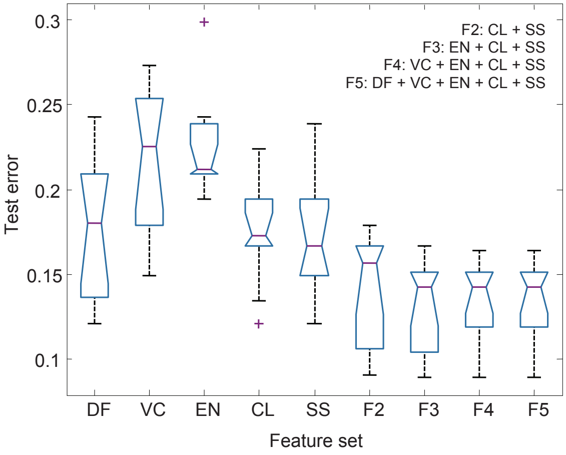


Figure 9.6: Boxplot of the 10-fold cross validation results on the specificity ranking task for individual features and various feature sets. The test error indicates the fraction of incorrectly ranked tag pairs.

The classification is executed in 3 steps. First we optimise a ranking classifier to order t_a and t_b given that they are part of **C2** or **C3**. Based on the optimal specificity classifier and similarity features we then classify the tag pairs that are considered equally specific (**C1**). Finally, we optimize a classifier to separate the classes **C1-3** and **C4**.

For these three classifiers we primarily focus on the relative performance of the proposed features, and pay less attention to the absolute performance of the final classifier.

9.5.1 Specificity Ranking

We start by only considering the tag pairs assessed as **C2** or **C3**. Since these two classes are symmetric (if $t_a > t_b$ then $t_b < t_a$), we treat the specificity ordering of two tags as a ranking problem on a list of size 2. We implement this task as a linear classifier with an unbiased hyperplane (decision boundary through the origin of the feature space). To get a set of features that describes the relation between two terms, we use the difference between the individual feature values of the two tags as input for the classifier (e.g. $\Delta DF(t_a, t_b) = DF(t_a) - DF(t_b)$). For sub-super we use the difference between both directional computations ($\Delta SS(t_a, t_b) = SS(t_a||t_b) - SS(t_b||t_a)$).

In Figure 9.6 we show a boxplot of the classification error of ten-fold cross-validation for the five individual features and for the best combinations of features. Both the clarity score and the sub-super method individually outperform the document frequency (See also Table 9.2).

For the combined feature classifiers we only show the optimal combinations for each number of features (F2-F5). The best individual features also give the best combination of two (F2 = CL + SS). With only three features the best classification result is obtained (F3 = EN + CL + SS). This classifier gives an error rate decrease

Table 9.2: Specificity ranking. The mean (μ) and standard deviation (σ) of the test error using 10-fold cross validation. The optimal result is obtained with F3 (EN+CL+SS).

Feat.	DF	VC	EN	CL	SS
μ	0.182	0.214	0.226	0.174	0.174
σ	0.041	0.045	0.030	0.032	0.039
Feat.	F2	F3	F4	F5	
μ	0.141	0.132	0.134	0.134	
σ	0.032	0.029	0.026	0.026	

of 27.4% over using document frequency alone. We further find that the proposed sub-super method gives the best individual performance and is part of each optimal feature set.

The weights of the optimal classifier are [EN:0.52, CL:-0.53, SS:-0.42] which shows that the three features have almost equal contribution in the final classifier. SS and CL have a negative weight, because they have an inverted relation to specificity (e.g. $CL(t_a) > CL(t_b) \Rightarrow t_a < t_b$).

Figure 9.7 shows the predicted relations for a subset of our training pairs. The left graph based on document frequency shows that more general concepts like *party* or *fun* occur less frequently than a more focused description like *wedding*, therefore a document frequency based ranking will fail to correctly rank these tags. The optimal classifier ranks the general terms *party* and *fun* above the more specific event names. The ordering of (*wedding*, *ceremony*) and (*halloween*, *costumes*) are still debatable, but they are identical for both classifiers.

9.5.2 Finding Similar Terms

Tag pairs that are equally specific (**C1**) can be detected using both specificity and similarity features. We use the absolute difference of the specificity features as input

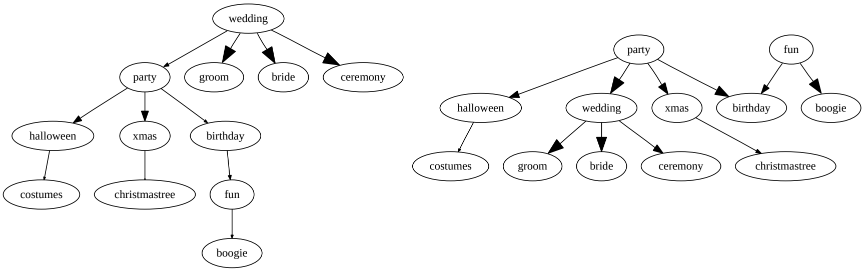
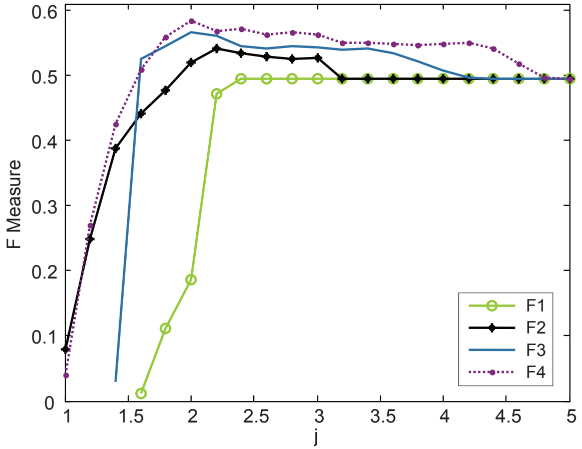


Figure 9.7: Specificity relations computed for a subset of our training set. The left graph is based on DF alone. The right graph is based on the optimal specificity classifier (F3). An edge between two tags only exists if the tag pair was judged by a human assessor (missing edges are not due to the method). The size of the arrow is proportional to the classification certainty (distance to classifier boundary). The absolute position in the graph does not carry any information.



F1: JP + CS + JC
F2: RLSCP + RLSCPR + RLSCS + KLD + SSS
F3: EN + CL + SS
F4: EN + CL + SS + JP + RLSCS + KLD + SSS

Figure 9.8: F-Measure for increasing (j), which represents the relative weight of the positive class (**C1**) compared to the negative class (**C2**+**C3**). F1: the optimal co-occurrence features, F2: the optimal context features, F3: the optimal specificity features, F4: the optimal set from all features.

for the similarity classifier (e.g. $\Delta DF(t_a, t_b) = |DF(t_a) - DF(t_b)|$).

The performance of the classifier is evaluated using the F-measure, which is defined as the harmonic mean of precision and recall. In this experiment the positive class is given by **C1** and the combination of **C2** and **C3** constitutes the negative class. Because of the large sample size difference between the two classes (326 vs. 665), we use a cost-factor j proposed by Morik et al. [90] to set the weight of test errors on positive examples versus errors on negative examples. The classes appear to be hard to separate as almost all samples end up in the negative class when $j = 1$. In this case the precision will be 0, resulting in an F-Measure of 0. For very large j all test pairs will be in the positive class giving an F-Measure of $(2 * 1 * 326/991)/(1 + 326/991) = 0.495$.

The results for four different feature sets are plotted in Figure 9.8. The tag pairs in our study are all selected based on high co-occurrence, therefore the co-occurrence features (F1) do not have any positive influence on the classification. The similarity features that take the tag context into account can give some improvement to find more similar tags (F2, optimal at $j = 2.2$, $F = 0.542$). The ten most similar tags according to the similarity classifier (Also indicated on the horizontal axis in Figure 9.9), and the actual classification given by the assessor are:

fraktal == *fraktals*
rainbows == *storms*
matt == *matthew*
blanc == *noir*
beautiful == *pretty*
pretty == *gorgeous*
747 == *737*
rust < *steel*
recreation == *leisure*
actress < *celebrity*

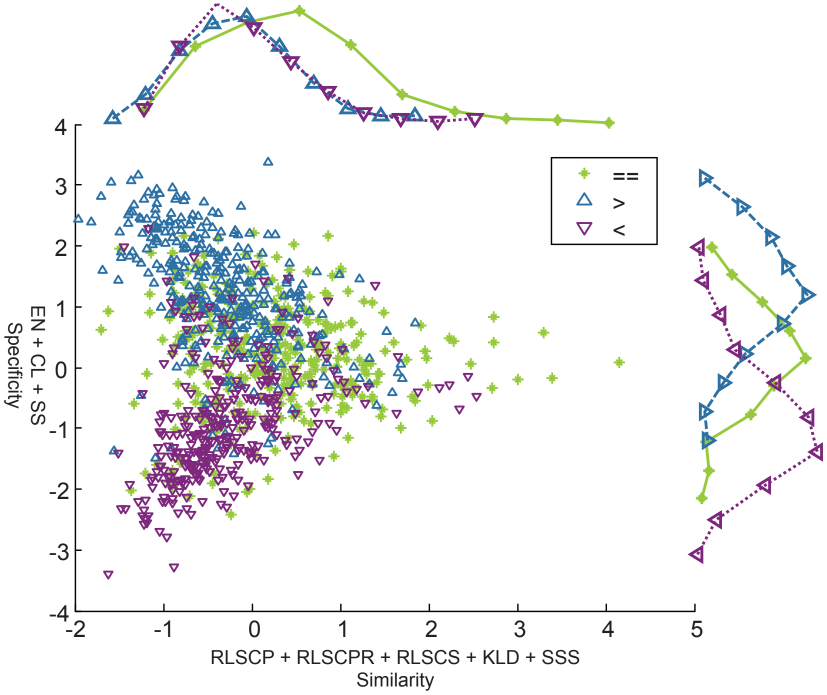


Figure 9.9: Scatterplot of the classes **C1**, **C2** and **C3**. The y-axis shows the distance to the classification boundary of the optimal specificity classifier. The x-axis shows the optimal similarity classifier without specificity features.

We further note that the specificity features (F3, optimal at $j = 2.0$, $F = 0.566$) outperform the set of best similarity features. This corresponds to the set-up of our experiments as we defined the class $==$ as 'equally specific'.

Figure 9.9 shows a scatter plot of classes **C1-3** on the optimal specificity and similarity classifier. As shown before, **C2** and **C3** are well separable using the 3 specificity features. As expected **C1** lies between the other two classes on the specificity axis and more to the right on the similarity axis. This indicates that both sets of features can be combined in a single similarity classifier. The combined feature set (F4 in Figure 9.8) shows that a combination of specificity and similarity features gives the best performance over a large range of values for j (F4, optimal at $j = 2.0$, $F = 0.583$).

The results in this section indicate that similarity metrics that take the context of the tags into account outperform the commonly used co-occurrence features on the detection of tags with a similar semantic specificity.

9.5.3 Remove Incomparable

Even though all selected tag pairs should somehow be related (due to the nature of the selection process), the assessors have assigned many tags to the incomparable

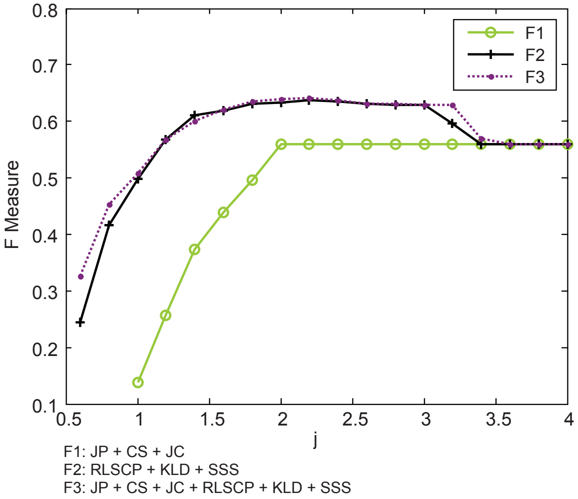


Figure 9.10: F-Measure for increasing settings of j for classification of comparable and incomparable terms. F1 contains the optimal set of co-occurrence features, F2 contains the optimal context features, F3 contains the combined feature sets.

class (**C4**) (about 56% more than **C1-3** combined). We train a classifier to separate the comparable (**C1-3**) from the incomparable (**C4**) tag pairs. In Figure 9.10 the F-Measure is plotted against j for several feature sets, where j is the relative weight of **C1-3** compared to **C4**. Again it is clear that co-occurrence features do not have much effect on the separation of the two classes. The optimal set of context features (RLSCP, KLD and SSS) give a large improvement over a big range of values for j. However, the optimal point $j = 2.2$ (with precision = 0.55 and recall = 0.75) is not good enough to directly use this method as a selection for specifically related terms. Additional work needs to be conducted to select the tags that can unambiguously be ordered in a network of tag with pairwise specificity relationships.

9.6 Example Applications

A network of tags with pairwise specificity relationships, like the one described in this chapter has various applications. In this section we will describe two use-cases. First, we will show that the notion of specificity can be used to improve search quality for web or image search. Second, we will discuss its application to interactive query refinement. The application of the tag specificity network is not limited to these examples but can potentially be used in more applications.

9.6.1 Term Weight Estimation

To optimise search results it is important to know which of the terms in a multi-term query is most representative for the user's information need. Inverse document frequency is commonly used as a weighing mechanism to increase the influence of less common query terms. Although the exact relation is debatable, it is commonly accepted that the inverse document frequency is related to the amount of information carried by a term [107]. We define semantic term specificity as a collection independent notion of information content. As our classifier gives a better estimation of term specificity it is a better method to weigh the terms in a query.

To evaluate which term is most representative for the full query information, we propose to use the web as an independent test collection, and use six different commercial search engines for both image and text search to validate the specificity prediction of our classifier. Our analysis is based on the assumption that two queries with similar information need will return a similar set of documents. This assumption was shown to be useful by the work of Yomtov et al. who used search engine results to estimate query difficulty [143].

We use the 665 tag pairs that have been manually assigned to class **C2** or **C3** as queries. For these queries we compare the result list of the full query consisting of both terms Q_b to the results of one of the individual terms, where the query created by the predicted specific term is denoted as Q_s and the query consisting of the general term Q_g. The set of top-100 search results for a given query Q is denoted as $R(Q)$. We now compare the number of overlapping results between the full query and the predicted specific term ($O_s = |R(Q_b) \cap R(Q_s)|$) to the number of overlapping results between the full query and the predicted general term ($O_g = |R(Q_b) \cap R(Q_g)|$).

Using the optimal specificity classifier, the results returned by the most specific term should be more similar to the full query results than the results based on the general term and thus: $O_s > O_g$.

Table 9.3 gives the results for the baseline classifier (based on DF alone) and our optimal classifier (F3, based on EN + CL + SS). The reported values are: *F/T*: The number of incorrect classifications ($O_s < O_g$) vs. the number of correct classifications ($O_s > O_g$). $\overline{O}_s$ and $\overline{O}_g$: the average values of O_s and O_g over all queries.

Our classifier gives a better prediction of the most specific term for all deployed search methods. This means that using the classifier, we can give a better prediction of term importance in a two-term query, which can result in improved user satisfaction.

Because the commercial search engines have a *black box* ranking algorithm, we validate the presented results on a public data set using a common ranking algorithm. We used the *wt10g TREC Web Corpus* and the Lemur implementation of the Inquery

Table 9.3: Search result overlap for the DF and F3 classifier on 6 public search engines and the TREC collection.

	DF			F3		
	F/T	$\overline{O}_s$	$\overline{O}_g$	F/T	$\overline{O}_s$	$\overline{O}_g$
Bing Web	96/471	13.34	2.14	83/484	13.60	1.87
Google Web	80/420	3.88	0.74	62/438	4.00	0.61
Yahoo! Web	97/477	10.02	2.26	80/494	10.33	1.95
TREC wt10g	115/293	32.26	18.72	109/299	32.42	18.56
Bing Image	80/394	8.23	1.01	66/408	8.28	0.97
Google Image	46/336	2.13	0.33	42/340	2.15	0.32
Yahoo! Image	38/321	3.31	0.24	23/336	3.36	0.18

retrieval model [14], with boolean AND operator and default parameters. The results on this collection support our findings that the F3 classifier outperforms DF on the prediction of specificity (See Table 9.3).

Next to the classification difference between DF and F3 we observe an interesting difference between the search engine response. In web search the average result list overlap is notably smaller for Google's search engine than for the other two. This could indicate that Google has a larger index but recent measurements show that Yahoo! has a larger index for web search[4]. Apparently Google has a significantly different method to deal with 2-term queries resulting in more diversification of search results. Looking at the image search results, we find that Yahoo! gives almost equally diverse results as Google, while Bing again shows a relatively large result overlap.

9.6.2 Interactive Query Refinement

Another application for a tag specificity network could be tag-based browsing tools like TagExplorer[5], which currently allows users to browse Flickr by showing related tags, grouped by concept (e.g. places, times, names, activities). The notion of specificity could extend the browsing interface of tag-explorer by giving the user the option to browse for more specific, similar or more general terms.

As an example, for the given query *wedding*, the tool currently offers the users the query refinement terms *party*, *groom*, and *bride* without indicating the effect of the refinement. Using the method proposed in this chapter the system could offer more intelligent support to the user by saying: do you want to narrow your search to the more specific topics *wedding bride* or *wedding groom*?; or do you want to explore the more general topic *party*?

9.7 Conclusions

In this work we have compared new and previously proposed methods to determine the specificity relation between two terms. The relative specificity of terms can be used to improve term weighting in a query. Also, we show that term specificity detection can be used to construct term ontologies and in an interactive browsing session the notion of specificity could allow the user to explore a more specific or more general concept.

We define three types of relationships that span the spectrum of relations that can be defined between two terms, i.e. the terms can be (1) incomparable, (2) of similar specificity, or (3) one term is more or less specific than the other. Using the Flickr tag corpus and a set of manually classified tag pairs we have evaluated the effectiveness of individually proposed metrics described in prior work in an SVM classifier. The clarity score and the proposed sub-super method both individually outperform a ranking on document frequency. Also, because the compared features are not strongly correlated, the combined classifier based on three features can improve the specificity classification significantly over all individual features. We even reach a 27.4% decrease in error rate over the traditionally used document frequency.

[4] http://www.worldwidewebsize.com/

[5] http://tagexplorer.sandbox.yahoo.com

We have shown that the detection of equally specific tags benefits from both specificity and similarity features. Because of the selection procedure of the evaluated tag pairs, we have only evaluated our classifier on tags with a high co-occurrence. All the tag pairs are therefore somehow related. We have shown that using context features instead of simple co-occurrence methods improves the selection of comparable tag pairs from those which are frequently used together.

We have introduced the sub-super method which contains both a specificity and similarity component. The difference in directed sub-super scores gives an indication of specificity difference, while the sum of both directed sub-super scores indicates if the two tags are similar. This method outperformed the previously proposed measures on specificity ranking and was included in the optimal feature set for all three classification tasks. Because this method takes both input tags into account instead of computing a global specificity score for each individual tag, this method is able to give a better estimation of the relation between both tags.

We have shown that an improved notion of specificity can be applied to estimate the weight of query terms in web and image search, and we propose useful applications to interactive browsing interfaces.

Appendix

Features that were included in our experimental evaluation, but did not improve performance in the final classifiers or give any other interesting insights:

Specificity

User-Image fraction (UI) It appears that the specificity of tags that describe an area or time fragment can be determined by the fraction:

$$UI(t_a) = |U(t_a)|/|I(t_a)| \tag{9.13}$$

This can be explained by the use of the 'batch annotation' function in Flickr. If someone has been on holiday in *France* in *2007*, he could use this function to annotate all the photos made during the trip with both these tags. If some people use this function on large sets of photos, this will result in a small UI ratio for big countries or large time spans.

KL-Divergence The KL-Divergence is proposed as a directed measure. If one of the tags (t_a) is a subset of the other (t_b) one might expect that $D_{KL}(t_a||t_b) < D_{KL}(t_b||t_a)$; Because all the tags co-occurring with t_a also co-occur with t_b, but not the other way around. This relation however did not prove to be strong enough to improve the classifier.

Sampled entropy The entropy based on a sample of 1000 images. This feature appeared to be strongly correlated to the vocabulary growth. The computation of the entropy (Eq. 9.2) depends on the absolute number of co-occurring tags (vocabulary size) and the co-occurrence frequency. The variation in co-occurrence frequency distribution between different tags seems to be very small, therefore the main component in the entropy computation is the vocabulary size.

Co-occurring The absolute number of unique tags that co-occur with t_a. This feature actually performed slightly better than DF individually, but did not improve the final classifier.

Annotation overlap frequency The average overlap between the annotations that contain t_a. We repeatedly draw two images from $I(t_a)$ and compute the fraction of the tags that overlap in the annotation of the two images. A more specific tag will be more likely to co-occur frequently with a small set of other tags.

Similarity

KLD10/50 The KL-Divergence based on the top 10 or 50 conditional probabilities appeared to perform slightly less than the top 100.

10

Detecting Synonyms in Social Tagging Systems to Improve Content Retrieval

Collaborative tagging used in online social content systems is naturally characterized by many synonyms, causing low precision retrieval. We propose a mechanism based on user preference profiles to identify synonyms that can be used to retrieve more relevant documents by expanding the user's query. Using a popular online book catalog we discuss the effectiveness of our method over usual similarity based expansion methods.

This chapter is an extended version of: Maarten Clements, Arjen P. de Vries, and Marcel J. T. Reinders. Detecting Synonyms in Social Tagging Systems to Improve Content Retrieval. In *SIGIR 08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 739-740, New York, NY, USA, 2008. ACM.

10.1 Introduction

Social networks have become popular platforms to share and retrieve multimedia content. To enable the retrieval of the unstructured data in these *social content systems*, collaborative tagging has shown to be an effective annotation mechanism. Many systems allow people to attach the terms they consider relevant for the content and tag-clouds are used to retrieve items introduced by others.

One of the problems in tagging systems is the fact that people use different terms to describe the content, resulting in low retrieval performance. Begelman et al. proposed to cluster the tags, in order to expand the users' queries with semantically related tags [8]. Other work has investigated the possibility to suggest tags to people when they have to annotate content, in order to increase the coherence of the *folksonomy* [141]. Tag suggestions induce the problem that new users can issue wrong queries, as they are not aware of the tagging policy in the network. Furnas et al. already advocated in 1987 that the optimal system would aggregate as many different descriptions as possible [40]. In this work, we show that similarity based clustering methods might be too rigorous in grouping terms and we propose a new method to identify true synonyms in social content systems. Our method does not enforce a certain tagging policy on users, but uses the naturally emerging structure to identify synonyms that can be used to expand the initial query.

10.2 Synonym Detection

To explain our synonym detection method we follow the example shown in Figure 10.1. A frequently used tag in many social content systems is the term *humour*. This term is however written differently in US-English (*humor*, also German and Spanish) and UK-English (*humour*, also French).

We postulate that synonyms are terms that are applied frequently on the same content, but are used by different user groups. Because people often prefer one of

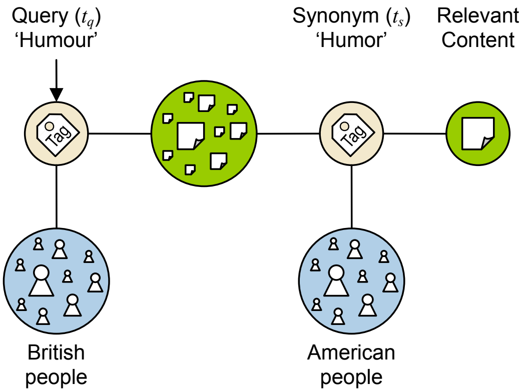


Figure 10.1: As an example of two synonyms we use the British 'humour' and the American 'humor'. True synonyms are applied to the same content set, while users often prefer one of the two terms.

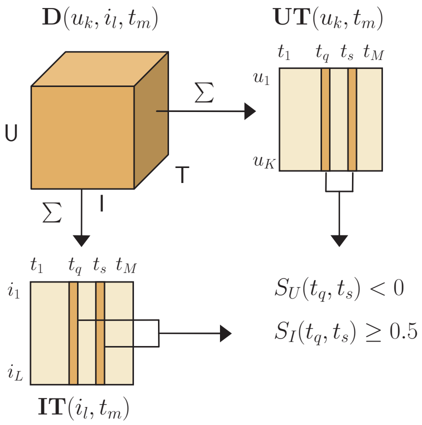


Figure 10.2: Synonyms are characterised by a large item similarity and a negative user similarity.

the synonyms (especially in language differences), different groups will emerge. We integrate these two characteristics in our model to identify synonymous terms.

The ternary relations in collaborative tagging systems can be visualized in a 3D matrix, see Figure 10.2. To derive binary tag relations, we compute the sum over the user and item dimensions of $\mathbf{D}$ and obtain:

- User-Tag matrix: $\mathbf{UT}(u_k, t_m) = \sum_{l=1}^{l=L} \mathbf{D}(u_k, i_l, t_m)$, indicating how many items each user tagged with which tag. A column from this matrix is referred to as a tag's user profile, containing a distribution of the most prominent tag users.

- Item-Tag matrix: $\mathbf{IT}(i_l, t_m) = \sum_{k=1}^{k=K} \mathbf{D}(u_k, i_l, t_m)$, indicating how many users tagged each item with which tag. A column from this matrix is referred to as a tag's item profile, containing a distribution of the most relevant items to that tag.

Based on these two binary relations, we can derive the similarity between two tags based on user or item overlap. The *item similarity* between two tags ($S_I(t_q, t_s)$) is derived by computing the Pearson correlation between the two profiles as follows:

$$S_I(t_q, t_s) = \rho(\mathbf{IT}_q, \mathbf{IT}_s) = \frac{\sum_{l \in L} (\mathbf{IT}_{l,q} - \mu_{\mathbf{IT}_q})(\mathbf{IT}_{l,s} - \mu_{\mathbf{IT}_s})}{\sigma_{\mathbf{IT}_q} \sigma_{\mathbf{IT}_s}}$$

where t_q is the query tag, t_s is the potential synonym, $\mathbf{IT}_{l,q}$ is the abbreviation of $\mathbf{IT}(i_l, t_q)$. The *user similarity* between two tags ($S_U(t_q, t_s)$) is computed analogously.

Our method first finds all terms that have an item similarity larger than 0.5 ($S_I(t_q, t_s) \geq 0.5$). On these similar terms, we compute the user similarity and retain all terms with a negative correlation as synonyms ($S_U(t_q, t_s) < 0$).

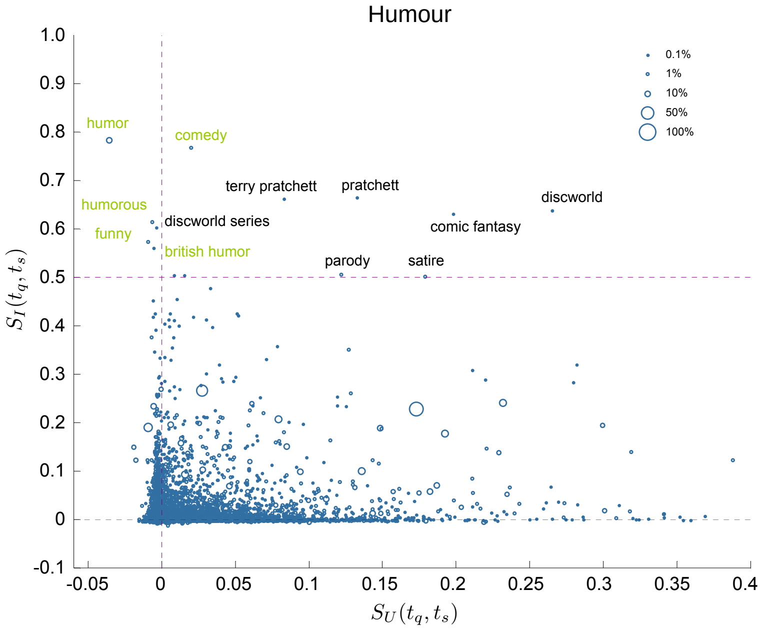


Figure 10.3: A scatter plot of the user and item similarity of all tags compared to 'Humour'. The size of the bubbles indicates the percentage of items that was given that tag at least once. The dotted lines show the manually defined classification rules.

10.3 LibraryThing

LibraryThing[1] is an online web service that allows users to create a tagged catalog of the books they own or have read. The popularity of the system has resulted in a database that contains almost 3 million unique works, collaboratively added by more than 300,000 users.

We have collected a trace from the LibraryThing network, containing 25,295 actively tagging users[2]. After pruning this data set we retain 7279 users that have all supplied tags to at least 20 books. We remove books and tags that occur in fewer than 5 user profiles, resulting in 37,232 unique works and 10,559 unique tags. This pruned data set contains 2,056,487 UIT relations, resulting in a density of $7.2 * 10^{-7}$ (fraction of non empty cells in $\mathbf{D}$). The derived $\mathbf{UT}$ and $\mathbf{IT}$ matrices have a density of respectively: $5.2 \cdot 10^{-3}$ and $2.0 \cdot 10^{-3}$.

10.4 Results

We use two examples to demonstrate the effect or our model. First we look at the query tag *Humour*, used on 2527 items in our data set. All similarities with the other tags in the data are scattered in Figure 10.2. The tags that we consider to be synonyms

[1] http://www.librarything.com
[2] Crawled in July 2007

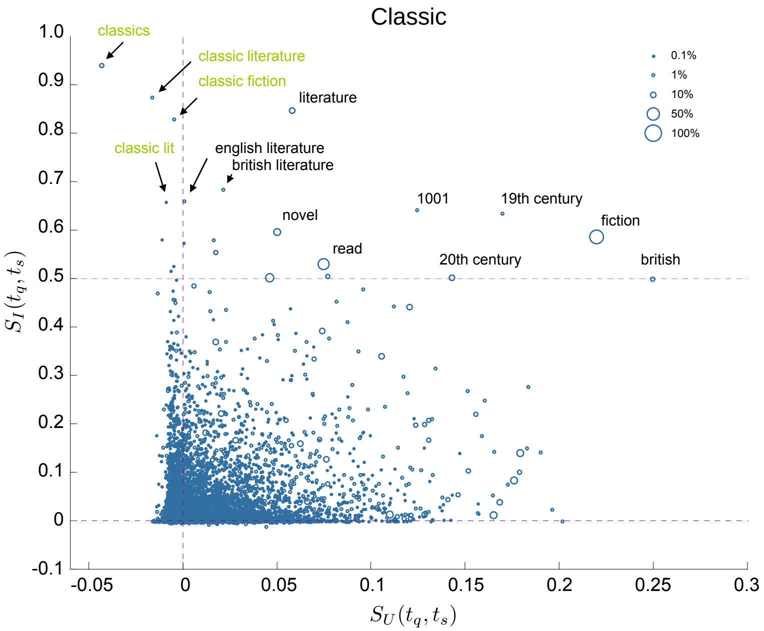


Figure 10.4: A scatter plot of the user and item similarity of all tags compared to ‘Classic’.

are shown in a green font, and clearly show a negative relation with user similarity. Table 10.1 shows the tags that have both $S_I(t_q, t_s) \geq 0.5$ and $S_U(t_q, t_s) < 0$, ‘Items’ indicates the amount of items it was used on and ‘New’ is the number of items not annotated by the query tag.

We see that the American form *humor* has a strong item correlation and a clearly negative user correlation (even the smallest user correlation in the entire data set). Only one of the terms is a truly incorrect result, the tag *discworld series*. If we use all proposed synonyms to enrich the initial query, we retrieve 3162 true positives and only 1 false positive.

If we would have ranked similar tags only on item similarity, the top-5 would contain the terms *pratchett*,*terry pratchett* and *discworld*, all related to the same book series. The *Discworld* series are generally regarded as humorous books, however we do not want to enforce them on people searching for the much more general term *humour*.

Table 10.1: Query: *Humour* (Items: 2527)

Proposed synonym	S_i	S_u	Items	New
humor	0.7829	-0.0356	4323	2511
funny	0.5738	-0.0091	1209	510
humorous	0.6144	-0.0065	364	132
british humor	0.5614	-0.0057	99	9
discworld series	0.6031	-0.0035	36	1

Table 10.2: Query: *Classic* (Items: 2872)

Proposed synonym	S_i	S_u	Items	New
classics	0.9407	-0.043	2094	824
classic literature	0.8742	-0.0164	494	72
19th century literature	0.5811	-0.0112	162	24
classic lit	0.6584	-0.0089	132	18
bbc big read	0.5162	-0.0066	53	6
assigned	0.5253	-0.0049	70	11
classic fiction	0.8288	-0.0048	430	60

The second example we discuss is the general tag *classic*. Figure 10.4 shows the scatter plot of all similarities and Table 10.2 contains the tags that qualify our synonym criterion. Most tags in the table are true synonyms. The term *19th century literature* is not synonymous, however most 19th century books that are still popular are considered classics. Only the tag *assigned* truly introduces wrong results, this term is only used on 70 books which makes the negative effect of this tag very limited

The item similarity between *classic* and *literature* is very high ($S_I(t_q, t_s) = 0.85$), therefore a clustering scheme based on content similarity alone could easily group these terms together. However, these terms have a positive user similarity ($S_U(t_q, t_s) = 0.06$), so our method correctly identifies that they are no synonyms. The exploitation of information about user groups allows our model to distinguish between frequently co-occurring terms and true synonyms.

Part V

The road ahead

11 Discussion and Conclusion

11.1 Contributions

In this thesis, we have studied many personalised retrieval tasks to improve the understanding of collaborative annotation phenomena. The presented work has investigated whether and how collaborative annotation corpora can be used to improve personalised data access in social media. Seemingly similar personalisation tasks strongly depend on variations in exact task definition, system design and data characteristics.

We have studied the tasks of item, tag or user recommendation from different viewpoints. The most common recommendation task in literature is focused on rating prediction for a given user and item. This task is often approached with memory based collaborative filtering and dimensionality reduction techniques. We have shown that if only a limited number of user profiles can be stored, the recommender should base its decisions on users with many ratings instead of just the top most similar users. If item recommendation is approached as a content ranking task, graph based methods perform much better. These methods can however not deal with the negative relevance indications contributed by the users' low ratings. We propose a combination of two separate graphs to overcome this limitation and showed that low ratings can have a positive contribution in the ranking of appreciated content.

To add more contextual information, the user-item graph can be extended with tags. Because tags indicate the specific aspects that relate certain items or users this tripartite graph can improve item recommendation or the recommendation of new social relations. The dynamics of this graph however strongly depend on the tagging rights in the system and whether the system assists the users in choosing their tags. If a system suggests tags previously used by others, users choose exactly the same tag if they agree on the content description, or they decide to add a new tag if they consider the tag suggestions incomplete or incorrect. In this way, there is a larger agreement on the description of the content compared to a system without tag suggestions.

We have proposed the task of recommending locations in a previously unvisited region, based on geotags. This task differs from traditional recommendation, as the objects of interest are continuously valued data points. In a continuous object space, traditional recommender techniques will not be able to find any similarities, because no two objects are identical. The scale at which the data is observed determines which geotags are related to the same object. Using a scale-space based on a Gaussian density estimation of the data we have been able to evaluate different location recommendation tasks at various scales. We have shown that many users have mixed location preferences, making an item-based approach more effective than a user-based approach. For location recommendation it appears to be harder to improve over the popularity baseline than on traditional movie, music or book data sets. This can be explained because almost all users enjoy the most popular landmarks in a city, while the most popular movies are still highly debated. The findings on location recommendation show that the characteristics of the target objects strongly determine which retrieval models can be effective for personalised recommendations.

Recommendation and search are two highly related topics. Personalised search is the task to predict a ranking of items based on both a query and the user's history. In this thesis we have used the tags assigned by users as possible queries. Hereby we assume that tags and queries are generated by the same process when a user makes a conceptual model of the content. Tags and queries are thus described by the same language model, which is a common assumption when queries and documents are concerned in information retrieval.

If a tagging system is designed in such a way that only the contributer of the content is allowed to add tags to that item, annotations will be too sparse to allow effective data access. The retrieval model needs to apply a smoothing method that integrates latent semantic relations to other tags. When the user however increases the length of his query, the need for personalisation and smoothing disappears, as longer queries are less ambiguous. By investigating the pairwise relation between terms we have been able to improve the understanding of multiple term queries. An improved understanding of tag semantics has many applications in both search and browsing interfaces.

11.2 Data Driven Approach

Information retrieval and recommender system research is often focused on the optimisation of the retrieval system for a specific data set. Data corpora like the ones provided by the TREC[1] initiative and the Netflix[2] competition have great value for scientific research, but the focus should not solely be on the optimisation of retrieval performance on these collections. A highly parametric method can always be tuned to achieve the optimal ranking performance, this does however only have scientific value if the parameters can still be related to real world phenomena.

In this thesis we have taken a data driven approach to analyse the retrieval effectiveness in social media. The optimisation of the model parameters has not been the final goal, but a means to learn about the data and understand the collaborative annotation phenomena. By repeatedly reformulating the retrieval tasks and evaluation criteria we have been able to reveal interesting parameter differences that explain the

[1] http://trec.nist.gov

[2] http://www.netflixprize.com/

user incentives that underly the data. Predefined ontologies will not be able to capture subtle differences in the way these data sources are created. A common believe is that once enough data is present, the optimal model can be derived from the data itself [48]. The abundance of the data contributed by the collaboration of many users enables accurate parameter estimation of data driven methods.

We have substantiated the belief that effective ranking methods should explicitly exploit the graph structure of the data [6; 92]. Graph ranking methods do not only take the direct relation between entities into account but also include the indirect relations determined by the paths over different entities. These graph methods are also versatile enough to integrate data with different characteristics and can therefore deal with the dynamic nature of social media.

11.3 Open Issues

Privacy In social media, a large part of the contributed data or interactions with data contain privacy sensitive information. In this work we have left the issue of privacy completely in the hands of the user. All the annotations contributed by the users are directly used in the retrieval algorithms, making personal data accessible for the system owner or even other users. If all privacy sensitive information would be kept away from the retrieval system, many interesting data mining applications will be missed. Even when a user contributes private information it should be possible to exploit this information for personalisation and retrieval purposes. Recent work has shown that many simple mathematic operations can be executed in the encrypted domain, so that recommendations can be made without revealing the data [34]. Further development of these methods would even make it possible to suggest medication to patients because it worked on other people with similar symptoms, without revealing all files to the medical practitioner.

Rights Only a few years ago, most of the photos and videos online were made of celebrities. The question of privacy and rights was often ignored because these people had made an informed decision to become famous. Now everyone has the experience that information about their personal life is published by friends or relatives. Although many platforms allow the specification of many degrees of *creative commons* rights, this decision is left to the contributer of the content, without consulting the people that actually appear in the content. Many users feel that it should be made easier to manage your own contributions and determine who can access and use the content. In this thesis we have not used any methods that exploit the actual content and therefore we have not touched this issue. The combination of tagging and face detection or recognition has however shown to be an effective combination to annotate people in photos[3] and tags from social media can even be used to improve the facial recognition system[4]. In this way, the collaborative tagging effort can contribute to the annotation of all people appearing in the content and therefore help users to locate all the content that concerns them.

[3] http://face.com/

[4] http://www.polarrose.com/

Scalability Personalisation requires different computations for each individual. In standard collaborative filtering problems the item-based approach has shown to scale better than the user-based approach. The item-based approach allows for offline computation of the item similarity model and therefore the online computation scales independently of the number of customers and number of items in the product catalog [79]. The location recommender that was proposed in Chapter 8 is based on the same principle of item similarity and is therefore applicable to large data sets.

The personalised random walk that was used in this thesis is an expensive procedure if it has to be evaluated for each query by each user. Computing all similarities offline would however require a storage capacity of $(|U| + |I| + |T|)^2$. As most tasks will only require the most relevant entities to a certain query a large part of all similarities are irrelevant. Dependent on the available space, the system could be designed to only store the top-N similarities of each entity and update these similarities offline.

Scattered Data Driven by the value of user data for personalised advertisements, the main battle between social media sites is focused on collecting and exploiting as much data as possible. These companies are therefore reluctant to share the information with others and even have conflicts with their users on the ownership of the data. As a result, all the user contributed data is currently scattered over many platforms. For most accurate personalisation, all user data on the Internet should however be aggregated into a single model. One initiative to aggregate data is Facebook's recently released open graph concept[5]. This protocol allows them to gain access to preference indications given by users on content on completely different domains. These and similar efforts will continue to drive the battle for data in the coming years. Because Internet users are increasingly aware of the possibilities created by extensive data mining on their personal data, only systems that adequately engage their community in the process of setting rights will be able to sustain their expansion.

11.4 Future Prospect

It will not be long before all locations and interactions between people can be stored and made available online, so that at any time and place real-time information will be available about a user's surroundings, including both objects [146] and people [18]. Data filtering methods need to be able to deal with these vast amounts of data and find patterns in the information obtained though many different channels.

Context based filtering methods relying on the current time, place, social company or even bodily functions are being developed to accurately assist the user. Because of the size and dynamic nature of this data, we believe that data driven approaches will be most versatile to deal with this information. Especially graph algorithms are repeatedly proving their value in the ranking of organically created data structures.

Complete knowledge about the current and past states of the user is the key to providing accurate information in real time. The development of sensor technology like GPS and Near Field Communication will stimulate the measurement of the full

[5] http://opengraphprotocol.org/

user context. Currently much research is also focused on brain-computer interaction to replace traditional queries by thought. Direct neural interfaces that can understand signals emerging from the brain will allow people to control the information filtering process by thought. Whether detailed cognitive information models can ever be understood from brain measurements is still a mystery, but simple tasks can already be accomplished in this manner [100; 89].

Once the context of the user can accurately be modeled, advertisement, recommendation and search will converge to a single information channel as they all have the same goal: match the right information to the user at the right time. Driven by someone's previous interactions or experiences of peers in a similar situation the system has to estimate the the user's current information need, whether it is a product or general knowledge. Therefore it will be vital for product or information selling companies to engage the community in their marketing strategy.

How these developments will be received by the community is hard to predict. The difference in adoption of new technology between, but also within, generations appears to be increasing. Where some people have actively integrated online interaction in their daily routine, the Internet is still a mystery to others. Many people are discouraged to provide personal data because there are no clear rules regarding the privacy and rights of this data. For many users, the new opportunities created by current technologies outweigh the downsides and it is unlikely that the upward trend in the amount of shared data will bend down. Only the future can tell how these changes will find a place in society.

11.5 Conclusion

The emergence of social media has generated an unprecedented growth of information. The world has come to a state where everyone generates digital content and everyone can and expects to be informed about the most relevant content for his individual preference. The many traces that are left either explicitly or implicitly by each individual can effectively be used to adapt retrieval and recommender systems to a user's preference. Personalised retrieval based on these collaborative annotations is a big step forward in the history of information access.

The optimal relevance model is highly dependent on the formulation of the retrieval task. Variations in the context of the user, object scale or the way the data is presented have a big impact on the optimal parameters of the prediction method. Also, the characteristics of the collaborative annotations and the annotated content should be taken into account when developing a personalised retrieval model. These characteristics are determined by the system design, but can also be inherently related to the data type.

Once we completely understand how the user's preference is extracted from his interactions with the data, it will be possible to accurately use this information for personalised data delivery. Using data driven approaches we have studied how different collaborative annotation methods can be used to learn the personal preference of an individual. With this study, we have been able to reveal new personalisation opportunities and we have taken a step towards accurate personalised access to social media.

Bibliography

[1] Shane Ahern, Mor Naaman, Rahul Nair, and Jeannie H. Yang. World explorer: visualizing aggregate data from unstructured text in geo-referenced collections. In *JCDL '07: Proceedings of the 2007 conference on Digital libraries*, pages 1–10, New York, NY, USA, 2007. ACM Press.

[2] Sihem Amer-Yahia, Michael Benedikt, and Philip Bohannon. Challenges in searching online communities. *IEEE Data Eng. Bull.*, 30(2):23–31, 2007.

[3] Avi Arampatzis and Jaap Kamps. A study of query length. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 811–812, New York, NY, USA, 2008. ACM.

[4] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[5] Shenghua Bao, Guirong Xue, Xiaoyuan Wu, Yong Yu, Ben Fei, and Zhong Su. Optimizing web search using social annotations. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 501–510, New York, NY, USA, 2007. ACM Press.

[6] Albert-László Barabási. *Linked: The New Science of Networks*. Perseus Books Group, Cambridge, MA, USA, April 2002.

[7] Albert-László Barabási. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211, 2005.

[8] Grigory Begelman, Philipp Keller, and Frank Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *WWW2006: Proceedings of the Collaborative Web Tagging Workshop*, Edinburgh, Scotland, 2006.

[9] Nicholas J. Belkin, Diane Kelly, G. Kim, Ja-Young Kim, Hyuk-Jin Lee, Gheorghe Muresan, Muy-Chyun Tang, Xiao-Jun Yuan, and Colleen Cool. Query length in interactive information retrieval. In Nicholas, editor, *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 205–212, New York, NY, USA, 2003. ACM Press.

[10] Tim Berners-Lee. Information management: A proposal. Technical report, CERN, March 1989.

[11] Toine Bogers. *Recommender Systems for Social Bookmarking*. PhD thesis, Tilburg University, Tilburg, The Netherlands, 2009.

[12] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52, San Francisco, 1998. Morgan Kaufmann.

[13] Vannevar Bush. As we may think. *The Atlantic Monthly*, 176:101–108, July 1945.

[14] James P. Callan, W. Bruce Croft, and Stephen M. Harding. The inquery retrieval system. In *Proceedings of the Third International Conference on Database and Expert Systems Applications*, pages 78–83. Springer-Verlag, September 1992.

[15] Sharon A. Caraballo and Eugene Charniak. Determining the specificity of nouns from text. In *Proceedings of SIGDAT-99*, pages 63–70, 1999.

[16] Vincenza Carchiolo, Michele Malgeri, Giuseppe Mangioni, and Vincenzo Nicosia. Social behaviours applied to p2p systems: An efficient algorithm for resources organisation. In *2nd International Workshop on Collaborative P2P Information Systems*, Manchester, UK, June 2006.

[17] Ciro Cattuto, Andrea Baldassarri, Vito D. P. Servedio, and Vittorio Loreto. Vocabulary growth in collaborative tagging systems. *arXiv:0704.3316v1 [cs.IR]*, April 2007.

[18] Ciro Cattuto, Wouter Van den Broeck, Alain Barrat, Vittoria Colizza, Jean-François Pinton, and Alessandro Vespignani. Dynamics of person-to-person interactions from distributed rfid sensor networks. *PLoS ONE*, 5(7):e11596+, July 2010.

[19] Òscar Celma and Perfecto Herrera. A new approach to evaluating novel recommendations. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 179–186, New York, NY, USA, 2008. ACM.

[20] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, Aug 1995.

[21] Ed H. Chi and Todd Mytkowicz. Understanding the efficiency of social tagging systems using information theory. In *HT '08: Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pages 81–88, New York, NY, USA, 2008. ACM.

[22] Maarten Clements, Arjen P. de Vries, and Marcel J. T. Reinders. Detecting synonyms in social tagging systems to improve content retrieval. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 739–740, New York, NY, USA, 2008. ACM.

[23] Maarten Clements, Arjen P. de Vries, and Marcel J. T. Reinders. Optimizing single term queries using a personalized markov random walk over the social graph. In *Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR)*, March 2008.

[24] Maarten Clements, Arjen P. de Vries, and Marcel J. T. Reinders. Exploiting positive and negative graded relevance assessments for content recommendation. In K. Avrachenkov, D. Donato, and N. Litvak, editors, *WAW'09: Proceedings of the 6th International Workshop on Algorithms and Models for the Web-Graph, LNCS 5427*, pages 155–166. Springer-Verlag, Berlin, Heidelberg, February 2009.

[25] Maarten Clements, Arjen P. de Vries, and Marcel J. T. Reinders. The influence of personalization on tag query length in social media search. *Information Processing & Management*, 46:403–412, May 2010.

[26] Maarten Clements, Arjen P. De Vries, and Marcel J. T. Reinders. The task dependent effect of tags and ratings on social media access. *ACM Transactions on Information Systems*, 28(4), October 2010.

[27] Maarten Clements, Pavel Serdyukov, Arjen P. de Vries, and Marcel J. T. Reinders. Finding wormholes with flickr geotags. In Cathal Gurrin, Yulan He, Gabriella Kazai, Udo Kruschwitz, Suzanne Little, Thomas Roelleke, Stefan Rüger, and Keith Rijsbergen, editors, *ECIR 2010, LNCS 5993*, pages 658–661. Springer-Verlag, Berlin, Heidelberg, 2010.

[28] David Crandall, Lars Backstrom, Dan Huttenlocher, and Jon Kleinberg. Mapping the world's photos. In *WWW '09: Proceeding of the 18th international conference on World Wide Web*, pages 761–770, 2009.

[29] Nick Craswell and Martin Szummer. Random walks on the click graph. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 239–246, New York, NY, USA, 2007. ACM.

[30] Arturo Crespo and Hector G. Molina. Semantic overlay networks for p2p systems. Technical report, Computer Science Department, Stanford University, 2002.

[31] Steve Cronen-Townsend and Bruce W. Croft. Quantifying query ambiguity. In *Proceedings of the second international conference on Human Language Technology Research*, pages 104–109, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[32] Denis Diderot. The definition of an encyclopedia. In Keith M. Baker, editor, *The Old Regime and the French Revolution*. The University of Chicago Press, Chicago, 1987.

[33] Darcy DiNucci. Fragmented future. *Print*, 53(4):32, 1999.

[34] Zekeriya Erkin. *Secure Signal Processing: Privacy Preserving Cryptographic Protocols for Multimedia*. PhD thesis, TU Delft Mediamatica.

[35] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 301–312, New York, NY, USA, 2003. ACM.

[36] Christiane Fellbaum. *Wordnet: An Electronic Lexical Database*. Bradford Books, May 1998.

[37] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):355–369, March 2007.

[38] Peter M. Fraser. *Ptolemaic Alexandria*. Clarendon Press, Oxford, 1972.

[39] Simon Funk. http://sifter.org/˜simon/journal/20061211.html, 2006.

[40] George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. The vocabulary problem in human-system communication. *Commun. ACM*, 30(11):964–971, November 1987.

[41] Eric Gilbert and Karrie Karahalios. Predicting tie strength with social media. In *In Proceedings of the Conferece on Human Factors in Computing Systems (CHI'09)*, pages 211–220, New York, NY, USA, April 2009. ACM.

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

[43] Scott A. Golder and Bernardo A. Huberman. The structure of collaborative tagging systems. Technical report, Information Dynamics Lab, HP Labs, 2006.

[44] Scott A. Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006.

[45] Marco Gori and Augusto Pucci. Research paper recommender systems: A random-walk based approach. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 778–781, Washington, DC, USA, 2006. IEEE Computer Society.

[46] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, pages 576–587. VLDB Endowment, 2004.

[47] Katie Hafner. And if you liked the movie, a netflix contest may reward you handsomely. In *New York Times*, October 2006.

[48] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, March 2009.

[49] Harry Halpin, Valentin Robu, and Hana Shepherd. The complex dynamics of collaborative tagging. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 211–220, New York, NY, USA, 2007. ACM.

[50] Claudia Hauff, Vanessa Murdock, and Ricardo Baeza-Yates. Improved query difficulty prediction for the web. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 439–448, New York, NY, USA, 2008. ACM.

[51] Ben He and Iadh Ounis. Inferring query performance using pre-retrieval predictors. In *In Proc. Symposium on String Processing and Information Retrieval*, pages 43–54, Berlin, Heidelberg, 2004. Springer-Verlag.

[52] Jon Herlocker, Joseph A. Konstan, and John Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Inf. Retr.*, 5(4):287–310, October 2002.

[53] Jonathan L. Herlocker and Joseph A. Konstan. Content-independent task-focused recommendation. *IEEE Internet Computing*, 5(6):40–47, November 2001.

[54] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, New York, NY, USA, 1999. ACM.

[55] Williamina A. Himwich, Eugene Garfield, Helen O. Field, John M. Whittock, and Sanford V. Larkey. Final report on machine methods for information searching. Technical report, The Johns Hopkins University, Baltimore, Maryland, 1955.

[56] Susan Hockey. The history of humanities computing. In Susan Schreibman, Ray Siemens, and John Unsworth, editors, *A Companion to Digital Humanities*. Blackwell, Oxford, 2004.

[57] Tzvetan Horozov, Nitya Narasimhan, and Venu Vasudevan. Using location for personalized poi recommendations in mobile environments. In *SAINT '06: Proceedings of the International Symposium on Applications on Internet*, pages 124–129, Washington, DC, USA, 2006. IEEE Computer Society.

[58] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In *ESWC 2006, LNCS 4011*, pages 411–426, Berlin, Heidelberg, 2006. Springer-Verlag.

[59] Zan Huang, Hsinchun Chen, and Daniel Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):116–142, January 2004.

[60] Peter Ingwersen. Polyrepresentation of information needs and semantic entities: elements of a cognitive theory for information retrieval interaction. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 101–110, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[61] Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Inf. Process. Manage.*, 36(2):207–227, March 2000.

[62] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.

[63] Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in folksonomies. In *Knowledge Discovery in Databases: PKDD 2007. LNCS 4702*, pages 506–514. Springer-Verlag, Berlin, Heidelberg, 2007.

[64] Thorsten Joachims. Making large-scale support vector machine learning practical. *Advances in Kernel Methods - Support Vector Learning*, pages 169–184, 1999.

[65] H. Joho and M. Sanderson. Document frequency and term specificity. In *Proceedings of the Recherche d'Information Assistée par Ordinateur Conference (RIAO)*, Pittsburgh, PA, USA, June 2007.

[66] Evangelos Kalogerakis, Olga Vesselova, James Hays, Alexei A. Efros, and Aaron Hertzmann. Image sequence geolocation with human travel priors. In *ICCV '09: Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[67] Owen Kaser and Daniel Lemire. Tag-cloud drawing: Algorithms for cloud visualization. In *Tagging and Metadata for Social Information Organization Workshop (WWW 2007)*, May 2007.

[68] K. Keenoy and M. Levene. Personalisation of web search. In B. Mobasher and S. S. Anand, editors, *ITWP 2003, LNAI 3169*, pages 201–228, Berlin, 2005. Springer-Verlag Berlin Heidelberg.

[69] Lyndon Kennedy, Mor Naaman, Shane Ahern, Rahul Nair, and Tye Rattenbury. How flickr helps us make sense of the world: context and content in community-contributed media collections. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 631–640, New York, NY, USA, 2007. ACM.

[70] Jon M. K. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, September 1999.

[71] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[72] Renaud Lambiotte and Marcel Ausloos. Collaborative tagging as a tripartite network. In *ICCS 2006, LNCS 3993*, pages 1114–1117. Springer-Verlag, Berlin, Heidelberg, May 2006.

[73] Sanford V. Larkey. The army medical library research project at the welch medical library. *Bulletin of the Medical Library Association*, 37:121–124, 1949.

[74] Sang S. Lee, Dongwoo Won, and Dennis Mcleod. Tag-geotag correlation in social networks. In *SSM '08: Proceeding of the 2008 ACM workshop on Search in social media*, pages 59–66, New York, NY, USA, 2008. ACM.

[75] Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, and Stephen Wolff. A brief history of the internet. Technical report, Internet Society, February 1997.

[76] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, New York, NY, USA, 2003. ACM.

[77] Philip Lieberman. The evolution of human speech: Its anatomical and neural bases.

Current Anthropology, 48(1):39–66, February 2007.

[78] Tony Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224–270, 1994.

[79] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003.

[80] Marek Lipczak. Tag recommendation for folksonomies oriented towards individual users. In *Proceedings of ECML PKDD Discovery Challenge (RSDC08)*, pages 84–95, 2008.

[81] Nathan N. Liu and Qiang Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 83–90, New York, NY, USA, 2008. ACM.

[82] Alberto Manguel. *Een geschiedenis van het lezen (trans. Tinke Davids)*. AMBO, Amsterdam, 1996.

[83] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40, New York, NY, USA, 2006. ACM.

[84] Matthew R. Mclaughlin and Jonathan L. Herlocker. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 329–336, New York, NY, USA, 2004. ACM.

[85] Prem Melville, Raymond J. Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering. In *Eighteenth national conference on Artificial intelligence*, pages 187–192, Menlo Park, CA, USA, 2001. American Association for Artificial Intelligence.

[86] Peter Mika. Ontologies are us: A unified model of social networks and semantics. In Yolanda Gil, Enrico Motta, Richard V. Benjamins, and Mark Musen, editors, *ISWC 2005, LNCS 3729*, pages 522–536. Springer-Verlag, Berlin, Heidelberg, 2005.

[87] Batul J. Mirza, Benjamin J. Keller, and Naren Ramakrishnan. Studying recommendation algorithms by graph analysis. *Journal of Intelligent Information Systems*, 20(2):131–160, March 2003.

[88] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42, New York, NY, USA, 2007. ACM.

[89] Yoichi Miyawaki, Hajime Uchida, Okito Yamashita, Masa-aki Sato, Yusuke Morito, Hiroki C. Tanabe, Norihiro Sadato, and Yukiyasu Kamitani. Visual image reconstruction from human brain activity using a combination of multiscale local image decoders. *Neuron*, 60(5):915–929, December 2008.

[90] Katharina Morik, Peter Brockhausen, and Thorsten Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *ICML '99: Proceedings of the Sixteenth International Conference on Machine Learning*, pages 268–277, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[91] Mor Naaman, Susumu Harada, QianYing Wang, Hector G. Molina, and Andreas Paepcke. Context data in geo-referenced digital photo collections. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 196–203, New York, NY, USA, 2004. ACM.

[92] Mark E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[93] Mark E. J. Newman. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46(5):323–351, 2005.

[94] Michael Noll and Christoph Meinel. Web search personalization via social bookmarking and tagging. In *6th International and 2nd Asian Semantic Web Conference (ISWC2007+ASWC2007)*, pages 365–378, November 2007.

[95] E. Ogston, A. Bakker, and Van M. Steen. On the value of random opinions in decentralized recommendation. In *DAIS 2006*, pages 84–98, 2006.

[96] Miho Ohsaki, Shinya Kitaguchi, Hideto Yokoi, and Takahira Yamaguchi. Investigation of rule interestingness in medical data mining. In *Active Mining*, pages 174–189. 2005.

[97] Paul Otlet and W. Boyd Rayward. *International Organisation and Dissemination of Knowledge: Selected Essays of Paul Otlet*. Elsevier, Amsterdam, New York, 1990.

[98] Simon Overell, Börkur Sigurbjörnsson, and Roelof van Zwol. Classifying tags using open content resources. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 64–73, New York, NY, USA, 2009. ACM.

[99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The Pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[100] Mark Peplow. Mental ping-pong could aid paraplegics. *Nature*, August 2004.

[101] Adrian Popescu, Gregory Grefenstette, and Pierre-Alain Moëllic. Mining tourist information from user-supplied collections. In *CIKM '09: The 18th ACM Conference on Information and Knowledge Management*. ACM press, November 2009.

[102] Johan A. Pouwelse, Pawel Garbacki, Jun Wang, Arno Bakker, Jie Yang, and Alexandru Iosup. Tribler: A social-based peer-to-peer system. In *Proceedings of the 5th International P2P conference (IPTPS 2006)*, 2006.

[103] Naren Ramakrishnan, Benjamin J. Keller, Batul J. Mirza, Ananth Y. Grama, and George Karypis. Privacy risks in recommender systems. *IEEE Internet Computing*, 5(6):54–62, November 2001.

[104] Tye Rattenbury, Nathaniel Good, and Mor Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 103–110, New York, NY, USA, 2007. ACM.

[105] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM Press.

[106] Paul Resnick and Hal R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, March 1997.

[107] Stephen Robertson. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation*, 60(5):503–520, 2004.

[108] P. M. Ryu and K. S. Choi. Measuring the specificity of terms for automatic hierarchy construction. In *ECAI-2004 Workshop on Ontology Learning and Population*, 2004.

[109] Gerard Salton. *A Theory of Indexing*. J.W. Arrowsmith Ltd, Bristol, UK, 1975.

[110] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[111] Mark Sanderson and Bruce Croft. Deriving concept hierarchies from text. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–213, New York, NY, USA, 1999. ACM.

[112] Badrul Sarwar, George Karypis, Joseph Konstan, and John Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 285–295, New York, NY, USA, 2001. ACM.

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

[114] Ralf Schenkel, Tom Crecelius, Mouna Kacimi, Sebastian Michel, Thomas Neumann, Josiane X. Parreira, and Gerhard Weikum. Efficient top-k querying over social-tagging networks. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 523–530, New York, NY, USA, 2008. ACM.

[115] Richard Seltzer, Deborah S. Ray, and Eric J. Ray. *The AltaVista Revolution: How to Find Anything on the Internet*. Osborne/McGraw-Hill, Berkeley, CA, USA, 1996.

[116] Shilad Sen, Shyong K. Lam, Al M. Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl. tagging, communities, vocabulary, evolution. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 181–190, New York, NY, USA, 2006. ACM.

[117] Pavel Serdyukov, Vanessa Murdock, and Roelof van Zwol. Placing flickr photos on a map. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 484–491, New York, NY, USA, 2009. ACM.

[118] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher, and Robin Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 259–266, New York, NY, USA, 2008. ACM.

[119] Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 327–336, New York, NY, USA, 2008. ACM.

[120] Barry Smyth and Evelyn Balfe. Anonymous personalization in collaborative web search. *Information Retrieval*, 9(2):165–190, March 2006.

[121] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-Laszlo Barabasi. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, February 2010.

[122] Indeok Song, Robert Larose, Matthew S. Eastin, and Carolyn A. Lin. Internet gratifications and internet addiction: On the uses and abuses of new media. *CyberPsychology & Behavior*, 7(4):384–394, August 2004.

[123] Yang Song, Ziming Zhuang, Huajing Li, Qiankun Zhao, Jia Li, Wang C. Lee, and C. Lee Giles. Real-time automatic tag recommendation. In Sung H. Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat S. Chua, and Mun K. Leong, editors, *SIGIR*, pages 515–522, New York, NY, USA, 2008. ACM.

[124] Karen Spärck-Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[125] A. Sun and A. Datta. On stability, clarity, and co-occurrence of self-tagging. In *WSDM (Late Breaking-Results)*. ACM, 2009.

[126] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 43–50, New York, NY, USA, 2008. ACM.

[127] Martin Szummer and Tommi Jaakkola. Partially labeled classification with Markov random walks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, pages 945–952. MIT Press, 2001.

[128] Yuichiro Takeuchi and Masanori Sugimoto. Cityvoyager: An outdoor recommendation system based on user location history. In Jianhua Ma, Hai Jin, Laurence T. Yang, and Jeffrey J. P. Tsai, editors, *Ubiquitous Intelligence and Computing*, volume 4159, chapter 64, pages 625–636. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[129] Pang N. Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 32–41, New York, NY, USA, 2002. ACM.

[130] M. Tatu, M. Srikanth, and T. D'Silva. Rsdc'08: Tag recommendations using bookmark content. In *Proceedings of ECML PKDD Discovery Challenge (RSDC08)*, pages 96–107, 2008.

[131] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, December 1969.

[132] Thomas Vander Wal. Explaining and showing broad and narrow folksonomies. http://www.vanderwal.net/random/entrysel.php?blog=1635, 2005.

[133] Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508, New York, NY, 2006. ACM Press.

[134] Jun Wang, Johan Pouwelse, Reginald L. Lagendijk, and Marcel J. T. Reinders. Distributed collaborative filtering for peer-to-peer file sharing systems. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 1026–1030, New York, NY, USA, 2006. ACM.

[135] Xuanhui Wang, Jian-Tao Sun, Zheng Chen, and Chengxiang Zhai. Latent semantic analysis for multiple-type interrelated data objects. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 236–243, New York, NY, USA, 2006. ACM Press.

[136] Stanley Wasserman and Katherine Faust. *Social network analysis: methods and applications*. Cambridge University Press, Cambridge, 1994.

[137] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[138] Kilian Q. Weinberger, Malcolm Slaney, and Roelof Van Zwol. Resolving tag ambiguity. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 111–120, New York, NY, USA, 2008. ACM.

[139] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[140] Wensi Xi, Edward A. Fox, Weiguo Fan, Benyu Zhang, Zheng Chen, Jun Yan, and Dong Zhuang. Simfusion: measuring similarity using unified relationship matrix. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 130–137, New York, NY, USA, 2005. ACM.

[141] Zhichen Xu, Yun Fu, Jianchang Mao, and Difu Su. Towards the semantic web: Collaborative tag suggestions. In *WWW2006: Proceedings of the Collaborative Web Tagging Workshop*, Edinburgh, Scotland, 2006.

[142] Roy D. Yates and David J. Goodman. *Probability and Stochastic Processes*. John Wiley & Sons, Inc., New York, NY, USA, 1999.

[143] Elad Yom-Tov, Shai Fine, David Carmel, and Adam Darlow. Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 512–519, New York, NY, USA, 2005. ACM.

[144] Vincent W. Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative location and activity recommendations with gps history data. In *WWW '10: Proceeding of the 19th international conference on World Wide Web*, page 10, New York, NY, USA, April 2010. ACM.

[145] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 791–800, New York, NY, USA, 2009. ACM.

[146] Feng Zhou, Henry Duh Been-Lirn, and Mark Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *ISMAR '08: Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 193–202, Washington, DC, USA, September 2008. IEEE Computer Society.

[147] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph R. Wakeling, and Yi-Cheng Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515, March 2010.

Summary

Personalised Access to Social Media

In the last few years the World Wide Web has developed from a static information platform into a dynamic network. Because of this transition the Internet has acquired a crucial role in our society. On many websites users can personally contribute information, ranging from short text messages to photos and videos. Users can see the information contributed by others and respond to it. These *social media* actively engage their community in the structuring of the collection by making use of collaborative annotation methods. Social interactions and the open character of the system stimulate users to contribute annotations that are both useful for themselves and others. Next to an improved description of the collection, collaborative annotations give insight in the personal preferences of individual users. Through all interactions with the data, users leave traces that can be exploited by the system to learn this preference and personalise social media access for each individual.

Because collaborative annotation methods have only recently been used on a large scale, the study of the data characteristics and personalised collection access based on these annotations is still in its early days. This thesis contributes to the understanding of social media and collaborative annotation data by studying various data filtering tasks. Different types of collaborative annotations are used to adapt the collection access to the preference of individual users. The deployed data filtering methods are used as a means to learn about the factors that contribute to the accessibility of the information in the system. This is done by selecting the model parameters so that they relate to external factors that might influence the task. In this way, the variation of the parameter settings reveals insights in the data that can be related to system design or user behaviour. By iteratively changing the task definition and finding the optimal model parameters, this thesis simultaneously finds task specific personalised retrieval methods and increases the understanding of the underlying data.

The results in this thesis show that small variations in data type, user interface and other system aspects appear to have large influence on the access possibilities of social media. By increasing the understanding of collaborative annotation data and the aspects that influence this data, this thesis has been able to improve existing data filtering methods and propose new opportunities for effective personalised access to social media.

The results presented in this thesis can be used in the development of personalised social media access methods and gives insight in which system design choices result in the most descriptive annotation data. Personalised data access has made it easier for the user to discover interesting books, movies, touristic travel locations or other information. After the content has been observed by the user, the system can assist the user in the annotation of the content, thereby the system learns about the preferences of the user and it becomes easier for other users to retrieve the content. In this way, this thesis contributes to a world where less time needs to be spent on the retrieval of relevant information and the provided information is more accurately adjusted to the needs of each individual.

Maarten Clements

Samenvatting

Gepersonaliseerde Toegang tot Sociale Media

In de afgelopen jaren is het Wereld Wijde Web ontwikkeld van een statisch informatieplatform in een dynamisch netwerk. Door deze transitie heeft het Internet een cruciale rol in onze samenleving verworven. Op veel websites kunnen gebruikers zelf informatie toevoegen, uiteenlopend van korte tekstberichtjes tot foto's en video's. Gebruikers kunnen de informatie van anderen zien en hier op reageren. Deze *sociale media* betrekken hun gebruikers actief bij de structurering van de collectie door gebruik te maken van gezamenlijke annotatiemethoden. Sociale interacties en het open karakter van het systeem stimuleren gebruikers om annotaties toe te voegen die zowel voor henzelf als anderen waardevol zijn. Naast een verbeterde beschrijving van de collectie geven gezamenlijke annotatiemethoden ook inzicht in de persoonlijke voorkeuren van gebruikers. Doordat gebruikers bij alle interacties met de data sporen achterlaten over hun voorkeuren, is het mogelijk geworden om de toegang tot sociale media aan te passen aan de smaak van individuele gebruikers.

Omdat gezamenlijke annotatiemethoden pas recent op grote schaal toegepast worden, staat de studie van de datakarakteristieken en persoonlijke collectietoegang op basis van deze annotaties nog in de kinderschoenen. Dit proefschrift vergroot het begrip van sociale media en gezamenlijke annotatiemethoden door verschillende datafilteringstaken te behandelen. De data afkomstig van verschillende annotatiemethoden wordt gebruikt om de collectietoegang aan te passen aan de voorkeur van individuele gebruikers. Hierbij worden de gebruikte filtermethoden gezien als hulpmiddel om te leren welke factoren bijdragen aan de toegankelijkheid van de informatie in het systeem. Dit wordt gedaan door de parameters van het model zo te kiezen, dat ze gerelateerd zijn aan externe factoren die de taak kunnen beïnvloeden. Op deze manier geeft het varieren van de parameters inzichten in de data die gerelateerd kunnen worden aan systeemontwerp of gebruikersgedrag. Door beurtelings de definitie van de taak te veranderen en de optimale parameterinstellingen te zoeken, vindt dit proefschrift gelijktijdig taak afhanklijke gepersonaliseerde zoekmethoden en een beter begrip van de onderliggende data.

De resultaten in dit proefschrift tonen aan dat kleine variaties in datasoort, gebruikers interface en andere systeemaspecten grote invloed blijken te hebben op de toegangsmogelijkheden van sociale media. Door het begrip van gezamenlijke annotatiedata en de aspecten die deze data beïnvloeden te vergroten, heeft dit proefschrift bestaande datafilteringsmethoden kunnen verbeteren en nieuwe kansen gevonden voor effectieve persoonsgebonden toegang tot sociale media.

De resultaten gepresenteerd in dit proefschrift kunnen gebruikt worden bij de ontwikkeling van persoonsgebonden toegangsmethoden voor sociale media en geven inzicht in de ontwerpkeuzes die resulteren in de meest bruikbare annotatie data. Persoonsgebonden datatoegang maakt het voor de gebruiker makkelijker om interessante boeken, films, toeristische reisbestemmingen of andere informatie te ontdekken. Nadat de inhoud door de gebruiker gezien is kan het systeem de gebruiker begeleiden bij de annotatie van de data, waardoor het systeem leert over de voorkeuren van de gebruiker en andere gebruikers de inhoud gemakkelijker terug kunnen vinden. Op deze manier draagt dit proefschrift bij aan een wereld waarin minder tijd besteed hoeft te worden aan het vinden van relevante informatie en de aangeleverde informatie beter gericht is op de behoeften van elk individu.

Maarten Clements

Curriculum Vitae

Maarten Clements was born in Barendrecht, The Netherlands, on August 25, 1981. After obtaining his high school degree from Farel College in Ridderkerk in 1999, he started a bachelor study in Electrical Engineering at the Technical University in Delft.

During his bachelor, Maarten gained interest in data mining and pattern discovery which motivated him to follow the Media and Knowledge Engineering variant of the Electrical Engineering master. In this master program, Maarten worked for 3 months on an internship at British Telecom in Ipswich (England, 2004) with the goal to develop a system that automatically detects event-based clusters in home videos. In 2006, after taking part in the biomedical minor program, Maarten started his thesis project in the Bioinformatics group. With his thesis, Maarten improved the understanding of the baker's yeast organism by combining transcription factor binding information with gene expression profiles in a unified method to detect cooperating genes.

In 2006, the emerging dynamic field of web2.0 and recommendation technology motivated Maarten to start his PhD. project, of which the result is presented in this thesis. This project, supervised by Prof. dr. ir. Marcel J. T. Reinders and Prof. dr. ir. Arjen P. de Vries, was partially executed at the Technical University in Delft and partially at CWI, Amsterdam. In 2009 Maarten spent 3 months at Yahoo! Research in Barcelona (Spain, 2009) to work on the analysis of tag semantics. The results of this project are presented in Chapter 9 of this thesis.

In 2010 Maarten started working as a research engineer at TomTom, to improve Point Of Interest discovery in navigation systems.

A THOUGHT

People always tell you to think outside the box, but even when you think well outside the box, there will always be a larger box to encompass all your thoughts. One might however envision a spherical shaped box which as we know provides the maximal volume to surface ratio, giving rise to the optimal freedom of thought without running into the boundaries of imagination. Also, one does not risk getting stuck in one of the corners of his mind.