Characterizing and Modeling Web Sessions with Applications

Luca Chiarandini

TESI DOCTORAL UPF / 2014

Directores de la tesi

Prof. Dr. Ricardo Baeza-Yates
Department of Information and Communication Technologies

Dr. Alejandro Jaimes Larrarte
Yahoo Labs

UNIVERSITAT POMPEU FABRA

To Barcelona.

Acknowledgements

This research was conducted at the Web Research Group at Universitat Pompeu Fabra under the supervision of Prof. Dr. Ricardo Baeza-Yates and Dr. Alejandro Jaimes. I would like to express my gratitude to my supervisors for their understanding, patience, and support during all these years.

This work would not be possible without the help of all people I met at Yahoo Labs Barcelona. I would like to thank (in alphabetical order): Ahmed Abdelwahab, Luca Maria Aiello, Jordi Atserias, Nicola Barbieri, Ludovico Boratto, Ulf Brefeld, Ilaria Bordino, Francesco Bonchi, Sergio Duarte, Edoardo Galimberti, Francesco Gullo, Peter Haider, Farshad Kooti, Marek Lipczak, Lucrezia Macchia, Michael Matthews, Neil O'Hare, Jaram Park, Natalia Pou, Miriam Redi, Estefania Ricart, Diego Sáez Trumper, Ricard Sierra, Gaurav Singh, Pancho Tolchinsky, Antti Ukkonen, and Anh Thu Vu.

A special thanks goes to my fellows Ph.D. students, Eduardo, Michele, Janette, and Ruth.

I would like to hug my family and friends for the support during all these years.

A very special thank to my wife María, for being the most wonderful partner now and in the years to come.

Abstract

This thesis focuses on the analysis and modeling of web sessions, groups of requests made by a single user for a single navigation purpose. Understanding how people browse through websites is important, helping us to improve interfaces and provide to better content.

After first conducting a statistical analysis of web sessions, we go on to present algorithms to summarize and model web sessions. Finally, we describe applications that use novel browsing methods, in particular parallel browsing.

We observe that people tend to browse images in a sequences and that those sequences can be considered as units of content in their own right. The session summarization algorithm presented in this thesis tackles a novel pattern mining problem, and this algorithm can also be applied to other fields, such as information propagation. From the statistical analysis and the models presented, we show that contextual information, such as the referrer domain and the time of day, plays a major role in the evolution of sessions. To understand browsing one should therefore take into account the context in which it takes place.

Resumen

Esta tesis se centra en el análisis y modelaje de sesiones web: grupos de solicitudes realizadas por un único usuario para un sólo propósito de navegación. La comprensión de cómo la gente navega a través de los sitios web es importante para mejorar la interfaz y ofrecer un mejor contenido.

En primer lugar, se realiza un análisis estadístico de las sesiones web. En segundo lugar, se presentan los algoritmos para identificar los patrones de navegación frecuentes y modelar las sesiones web. Finalmente, se describen varias aplicaciones que utilizan nuevas formas de navegación: la navegación paralela.

A través del análisis de los registros de uso se observa que las personas tienden a navegar por las imágenes en modo secuencial y que esas secuencias pueden ser consideradas como unidades de contenido. La generación de resumenes de sesiones presentada en esta tesis es un problema nuevo de extracción de patrones y se puede aplicar también a otros campos como el de la propagación de información. A partir del análisis y los modelos presentados entendemos que la información contextual, como el dominio previo de acceso o la hora del día, juega un papel importante en la evolución de las sesiones. Para entender la navegación no se debe, por tanto, olvidar el contexto en que esta se lleva a cabo.

Resum

Aquesta tesi es centra en l'anàlisi i modelatge de sessions web: grups de sol·licituds realitzades per un únic usuari per un sol propòsit de navegació. La comprensió de com navega la gent a través de llocs web es important per millorar la interfície i oferir un millor contingut.

Primerament, es realitza un anàlisi estadístic de les sessions web. Seguidament, es presenten els algorismes d'identificació de patrons freqüents de navegació i modelatge de les sessions web. Finalment, es descriuen varies aplicacions que utilitzen noves formes de navegació: la navegació paral·lela.

Mitjançant el análisis dels registres d'ús s'observa que les persones tendeixen a navegar per les imatges de forma seqüencial i que aquestes seqüencies poden ser considerades com unitats de contingut. La generació de resums de sessions presentada en aquesta tesi es un problema nou d'extracció de patrons i pot ésser aplicat també a altres camps com el de la propagació de la informació. A partir del análisis i dels models presentats entenem que la informació contextual, com el domini previ d'accés o la hora del dia, juguen un paper important en la evolució de les sessions. Per entendre la navegació no s'ha d'oblidar, per tant, el context en el que aquesta es porta a terme.

Sommario

Questa tesi si concentra sull'analisi e modellazione di sessioni web, ovvero di gruppi di richieste presentate da un singolo utente per un unico scopo di navigazione. Capire come le persone navigano attraverso siti web è importante, dal momento che ci aiuta a migliorare l'interfaccia e a fornire migliori contenuti.

Dopo aver condotto un'analisi statistica delle sessioni web, si presentano algoritmi per sintetizzare e modellare sessioni web. Infine, si descrivono applicazioni che utilizzano nuovi paradigmi di navigazione web, in particolare la navigazione parallela.

Dall'analisi si evince che gli utenti tendono a visualizzare le immagini in sequenza e che tali sequenze possono essere considerate unit di contenuto. L'algoritmo di sintesi di sessioni presentato in questa tesi affronta un nuovo problema di *data mining*. Questo algoritmo può essere applicato anche ad altri campi, come la propagazione dell'informazione. Dall'analisi statistica dei modelli presentati, dimostriamo che l'informazione contestuale, come ad esempio il dominio *referrer* e l'ora del giorno, gioca un ruolo fondamentale nell'evoluzione delle sessioni. Per capire la navigazione si deve quindi tener conto del contesto in cui si svolge.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

It is crucial for service providers to understand the needs and preferences of their customers. This knowledge is useful in a number of applications such as improving the service or designing new products. In non-digital commerce, sources of information are mainly market studies done by means of surveys and interviews with customers. Such methods, although useful, are often expensive and limited in sample size since they require manual work.

With the penetration of the World Wide Web, people started to use websites to buy goods or just browse content. Through websites and mobile apps, people can get to know and share new ideas and products. Through browsing, one can understand, explore, and make sense of the large amount of data items. Web servers are therefore not only providers of services to the users, but also a useful source of information for their owners. It is indeed possible to record visits to web pages and track them in an automated and cheap way. Each time a user visits a page, the server adds a new line to the *server log* containing the time of the request, the identifier of the page, and other optional information.

The availability of server logs opens new horizons towards the understanding of user behavior. It is possible to precisely record what the user is seeing on the screen, capture the steps that led the user to undertake a particular action (*e.g.*, like a particular photo, buy a product), and understand how the user reaches a particular web page.

While server logs have been intensively analyzed in the case of *web search*, not much work has been carried out in the case of general browsing. According to Morse [110], browsing may be defined as a search, hopefully serendipitous. Differently than the case of search, while browsing the goal is less clear than a query or may not be so easily expressed by means of a text phrase. It is also possible that the user does not have any goal at all (*capricious browsing* [11]).

The challenge is therefore how to process the enormous amount of data in the server logs to extract useful knowledge and develop intelligent applications. The algorithms should be robust enough to cope with the noise derived from capricious browsing and should be able to understand the collective signal embodied in the actions of the users.

1.1. Goals

Service providers collect a large amount of information about users, about sessions, and about the content that is browsed. It is not trivial to figure out what part of this information is useful to understand the browsing behavior and what is irrelevant. For example, is the user gender, age, or geographical location a good indicator of what the user is going to see? How large is the influence of the context of the session, *e.g.*, time of day or referrer URL? There is still little work on it and we plan to tackle it in this thesis.

In addition to the study of the factors that influence the browsing behavior of users, we are also interested in the structure of web sessions, in terms of transitions among visited pages. Baeza-Yates *et al.* [6] states that the task in which users of a retrieval system are engaged may be of two distinct types: information or data *querying* (search) and *browsing*. Both are ways in which people access information, the first one being random access, and the latter being sequential access. One can expect that this distinction also exists in the structure of web sessions. Search may correspond to short sessions with branching and backtracking (caused by the use of multiple tabs and of the "back" button in the web browser). Browsing, on the contrary, may be more linear and correspond to long sequences of pages.

Modern websites make a clear distinction between search and browsing, up to the point that there are websites (search engines) solely dedicated to search. In this thesis, we explore ways of combining search and browsing. Fast retrieval of information as well as the exploration of content is possible, by taking advantage of both tasks at the same time. The structure of the sessions will also be different, showing branching and backtracking together with long browsing sequences of pages. We conjecture that this is helpful to go in depth into the results of queries and to understand the information space that is being browsed.

In this thesis, we will also focus on the content that is browsed. When a user clicks on a link, the choice is not random, but there is often a reason behind it. Therefore, pages that are browsed in a session are likely to be related somehow. This may have interesting applications. First of all, it should be possible to use the page transitions to build browsing graphs and these graphs would show communities of similar pages. Moreover, if there exist frequent information needs shared by many people, there should be frequent paths that people take in pages in order to satisfy such needs. Processing server logs should therefore allow us to discover them.

The goal of this thesis is to analyze all these aspects in order to gain insights about user browsing behavior.

1.2. Contributions

In order to validate the intuitions in Section 1.1 we perform research and extend the state of the art by:

- Improving the understanding of user browsing behavior on multimedia platforms. This is done through an analysis of server logs coming from Flickr. The analysis focuses on different phases of the life of a session: the moment it begins and its evolution. [Chapters 4 and 5]
- Developing a summarization algorithm to extract frequent browsing patterns. A summary corresponds to a set of sessions that browse the same pages in the same order. We evaluate the algorithm on several datasets. [Chapter 6]
- Proposing models of browsing sessions. The models take into account a variety of aspects of browsing behavior of the user such as the context, the content, and the social signal. The models allow us to understand which aspects have the greatest influence on the browsing behavior. [Chapters 7 and 8]
- Describing an extension of the traditional tree-structure model of sessions: parallel browsing. In parallel browsing the user follows multiple browsing threads at the same time. To illustrate and evaluate the usefulness of this model, we propose *PRiSMA*, a application to explore and retrieve photos from a large collection. [Chapter 9]

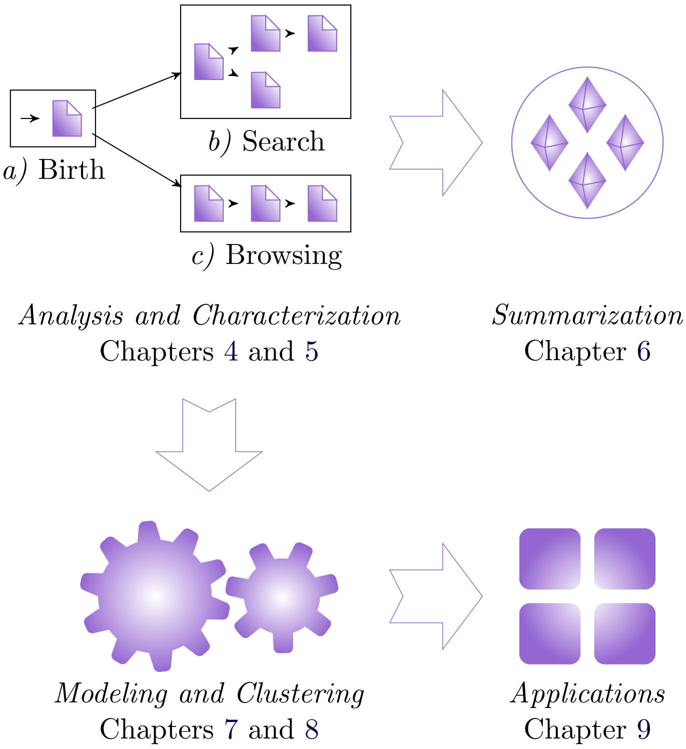


Figure 1.1: The conceptual flow of the thesis. The thesis starts from a characterization of web sessions (top left). Next, it continues by presenting an algorithm to extract frequent browsing patterns (top right). Given the characterization, models of browsing are presented (bottom left). The models can be used to cluster sessions. Finally, the thesis concludes by proposing new models of sessions (bottom right): parallel browsing and multidimensional browsing.

1.3. Outline

In this thesis, we characterize, model and exploit browsing sessions for developing intelligent applications.

Chapter 2 contains a review on related work. Chapter 3 introduces general elements that will be useful throughout the thesis and describes the datasets used in the experiments. The rest of the work is divided into four parts. Figure 1.1 illustrates the conceptual flow among them. We will overview each one individually.

Starting from a large sample of server logs, we characterize the behavior of users in a photo-sharing platform. First of all, we study how sessions are born, *i.e.*, what happens when users first enter a website. We show that the referrer URL plays a major role in the evolution of sessions (Chapter 4). This work has been done together with Michele Trevisiol [31]. The results of this chapter are published in:

- Luca Chiarandini, Michele Trevisiol, and Alejandro Jaimes. Discovering social photo navigation patterns. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages 31–36. IEEE, 2012.

We then characterize sessions in different tasks, such as search and browsing of images (Chapter 5). For the case of search, we show that the search task is often hierarchical in the sense that users follow a different search path each time and roll back if it leads to a dead end. On the contrary, when browsing images, it is more frequent to browse sequences of photos. We call such sequences *photostreams*. This leads to the intuition that photostreams can be considered as content units in their own right and allows us to aggregate and compact the representation. Section 5.2.1 is joint work with Michele Trevisiol [31]. This chapter led to the following publications:

- Silviu Maniu, Neil O'Hare, Luca Maria Aiello, Luca Chiarandini, and Alejandro Jaimes. Search behaviour on photo sharing platforms. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.
- Luca Chiarandini, Przemyslaw A. Grabowicz, Michele Trevisiol, and Alejandro Jaimes. Leveraging browsing patterns for topic discovery and photostream recommendation. In *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media, Cambridge, MA*, pages 71–80, 2013.
- Luca Chiarandini and Alejandro Jaimes. Browsing-based content discovery. In *Proceedings of the Designing Interactive Systems Conference. ACM*, 2012.

Given this characterization, we move towards models of user browsing behavior. We begin by tackling the problem of *summarizing* web sessions (Chapter 6). A summary is a group of sessions that visit more or less the same pages in the same order. Summaries are useful since they condense a particular user behavior. The results of this chapter were published in:

- Lucrezia Macchia, Francesco Bonchi, Francesco Gullo, and Luca Chiarandini. Mining summaries of propagations. *Proceedings of the 13th IEEE International Conference on Data Mining (ICDM'13)*, 0:498–507, 2013.

We then model sessions using a simple, yet reliable model (Chapter 7). We take into account many features, such as the interests of the user, the context of the session (*e.g.*, referrer URL, user location), and the content that is browsed. In this chapter we confirm the observations done in the characterization chapters by means of an unsupervised learning model. For example, the referrer URL is a very important predictor of the actions of the users. We then use a contextual session model to cluster sessions and we show the results in Chapter 8. This model is based on the features that give the greater contribution in the previous model. We evaluate the model on a large dataset of news browsing. The results of the modeling chapters were published in:

- Luca Chiarandini, Ricardo Baeza-Yates, and Alejandro Jaimes. User Browsing Behavior: Characterization and Modeling. *under review.*
- Peter Haider, Luca Chiarandini, Ulf Brefeld, and Alejandro Jaimes. Contextual models for user interaction on the web. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD 2012)*, 2012.

Chapter 9 is aimed at exploring new models of sessions by means of user applications. We first present an application in which sessions are composed by independent parallel browsing threads. Users can see on the screen multiple queries and browse them in parallel. A user study shows the advantages and disadvantages of this technique. In the second application users can explore people participation in public events. Users can move through all three dimensions of the interface, *i.e.*, people, photos and events. For example, one can look for a particular person, see all events in which a person took part, list all people in a photo, *etc.*A session is therefore a multidimensional path (in the people-photos-events information space). If we consider the session as a linear sequence of pages, we fail to capture the structure of user browsing. The following papers were published based on the results of Chapter 9:

- Pancho Tolchinsky, Luca Chiarandini, and Alejandro Jaimes. Prisma: searching images in parallel. In *Proceedings of the 13th ACM international conference on Multimedia*, pages 985–988. ACM, 2012.

- Luca Chiarandini, Luca Maria Aiello, Neil O'Hare, and Alejandro Jaimes. Metro: Exploring participation in public events. In *Proceedings of the 5th conference on Social Informatics*, pages 40–45. Springer, 2013.

Finally, Chapter 10 draws the conclusions and presents directions and ideas for future research.

Chapter 2

State of the Art

In this chapter we present other works that are relevant to this thesis. We organize related work according to the structure of the following chapters. First of all, in Section 2.1 we list works that deal with analysis and characterization of online user browsing behavior. This will be useful in Chapters 4 and 5. Secondly, we present a survey on summarization (Section 2.2), which is related to the work in Chapter 6. We then list work in the context of modeling in Section 2.3. This gives the context for Chapters 7 and 8. Section 2.4 presents related work on models for clustering browsing sessions, which is particularly useful in Chapter 8. Finally, the last section of the chapter gives an overview about applications for browsing images.

2.1. Characterizing Web Sessions

Much work has been done in characterizing and analyzing web sessions in social networks and multimedia platforms [85, 74, 26, 94]. Several authors have analyzed sessions and browsing behavior for various purposes. Benevenuto *et al.* [14] show a clickstream study over several social networks, proposing a clickstream model to characterize user behavior, while Jiang *et al.* [84] study the Chinese social network Renren, creating latent interaction graphs as a different representation of interaction based on “profiles” of browsing events.

We can divide the related work into characterization of image search and analysis of user browsing behavior.

2.1.1. Characterizing Image Search

There has been much work on analyzing the logs of commercial web search engines, uncovering relationships between queries [20] and using log analysis to improve search engine rankings [159]. Broder [22] proposes three distinct types of queries based on user intent: informational, navigational, and transactional. Other work automatically classifies queries within this taxonomy [83].

Studies of user behavior using web server logs are often limited by the fact that the logs only record interactions with the search engine itself, with subsequent actions not recorded in the logs. White and Drunker [157] circumvent this problem by inviting users to install a browser plug-in which logged all their browsing activities, and analyze the entire search sessions of over 2,000 participants, characterizing users based on *search trails*, similar to our *search trees*. The availability of tabbed browsing on modern web browsers means web browsing session are rarely linear, and models for tabbed browsing have been proposed by Chierichetti *et al.* [34]. Additionally, the use of the 'back' button can also result in browsing sequences that are not adequately represented by linear models. Much of the work on understanding image search behavior has focused on professional users, using a combination of qualitative methods and automatic analysis of search logs [154]. Such studies tend to show that a variety of search strategies are used, and that browsing and exploration are often important strategies [155]. In one such study, Westman and Oittinen [154] use interviews, observation, and analysis of image queries to understand the types of queries and search strategies used, while in other work they conduct user experiments to understand search strategies, comparing professionals to non-professionals [155].

Image search logs have been studied. Jansen *et al.* [82] analyze audio, video and image searches from the Alta Vista search engine. Andre *et al.* [5] analyze a large image search log and note that, compared with general web search sessions, image search session have greater average depth (number of results pages clicked for a query), that they have more results clicked, and that users spend more time looking at results pages, inferring that image search is more exploratory than web search. Other work looks at query modification patterns in image search [147], noting that users tend to replace search terms rather than adding or deleting them.

Researchers studied taxonomies for image search, attempting to adapt Broder's [22] taxonomy of web search to image search [102]. Query taxonomies for image search differ significantly from those used for web search, which is unsurprising since navigational and transactional queries are not really applicable to images. Instead, taxonomies of image search have looked at the type of objects and concepts that the query refers to. Enser [49], for example, distinguishes between unique (*e.g.*, specific people) and non-unique requests, each of which could be refined or not. Smeulders *et al.* [138] distinguish between target, category search, and search by association, with target and category search similar to unique and non-unique.

Difference from Previous Work. In this thesis we broadly follow those taxonomies for classifying queries, distinguishing between general and specific queries. Various works also show that specific people and other objects are particularly important in image search [154, 145]. The current work differs from previous work in image search analysis in that we have access to the users entire behavior within search sessions (specifically within search trees, as defined below in Section 5.1.1), in order to highlight common search behavior in a photo-sharing platform. Distinct from other work, after classifying the majority of queries into a taxonomy of query types, we then investigate the relationship between search behavior and query type, and also the extent to which search behavior is user-dependent.

2.1.2. Characterizing Image Browsing

Browsing behavior has been studied in many contexts [151, 26, 94, 107]. Some authors have studied user navigation patterns in Flickr. Most notably, Lerman and Jones [96] study how users find new images on Flickr, highlighting that people often navigate through photo streams of their contacts. They refer to such behavior as "social browsing" because users tend to browse the photos of their closest contacts. Other authors have also highlighted such behavior (*e.g.*, [92, 149]). Lipczak *et al.* [100] perform a similar study in Flickr also considering user behavior. However, they focused their attention on explicit user actions, in particular on favorites.

Huang *et al.* [78] take into account parallel browsing, *i.e.*, when a user navigates using multiple browser windows at the same time. Other researchers model user behavior considering the content of the pages [99] and even use it for tag recommendation in Flickr [142].

Gamon and König [58], study session logs collected from the Microsoft Live Toolbar. They group URLs into categories, for a manually defined list of websites, obtaining 5 categories. A somewhat related approach is proposed by Kumar and Tomkins [94], in which a URL taxonomy is generated by an automatic categorization. Other authors have focused on clustering or using Markov Chains (*e.g.*, Sharma et al. [136] and Vakali *et al.* [148]) to model user sessions.

Figueiredo *et al.* [52] and Yang and Leskovec [163] analyze popularity of content in online media. They show that the referrer has a strong influence on the popularity of items and could be used to predict it. Although not related to user browsing, these works are still relevant to this thesis, since they acknowledge the importance of the referrer domain.

Srikant and Yang [139] use implicit information extracted from server logs to improve the design of a website. In particular, the authors analyze the server logs in order to suggest modifications to the website link structure, to make content easier to find for the users.

Difference from Previous Work. The main difference between our work and previous work is that we take into account the referrer URL in order to model user behavior. Most work on session analysis on the Web focuses on modeling behavior independently of where the user comes from when visiting a website. In addition, our work differs from Lerman and Jones' [96] in the fact that we do not focus only on new images. More specifically, we take into account not just which photos users view, but also consider categories of pages within the Flickr site and, given the referral information, explicitly analyze users' behavior. Moreover, most of the above takes into account individual photos and does not consider photostreams as content units.

2.2. Session Summarization

The problem of session summarization (Section 6.2) represents an original type of structured pattern-mining problem, for which not much related prior research exists. We however briefly discuss the work in similar areas.

Graph Pattern Mining. The problem of graph pattern mining is to extract graph patterns (*e.g.*, trees or subgraphs) that appear frequently in a graph database, *i.e.*, a database composed by a large set of graphs. This research area has been quite active in the last decade and a lot of algorithms have been defined, such as AGM [80], FSG [95], gSpan [161], FFSM [76], SPIN [77], and Gaston [113]. Moreover, Yan *et al.* [162] propose a general framework to mine different graph patterns with possibly non-monotonic objective functions.

Graph Summarization. The problem of graph summarization is to create a coarser-grained version of a graph such that the most important features of the graph are retained. The typical approach is based on identifying and aggregating sets of similar nodes so that the error deriving from the aggregation is minimized. Navlakha *et al.* [111] exploit the MDL principle to summarize a graph with accuracy guarantees. Tian *et al.* [144] define a graph-summarization method that allows the user to specify the granularity level of the summarization in real-time.

Applications of Session Summarization. The study of the spread of information and influence through a social network has a long history in the social sciences. The first investigations focused on the adoption of medical [38] and agricultural innovations [150]. Later marketing researchers have investigated the "word-of-mouth" diffusion process for *viral marketing* applications [9, 60], which has then attracted most of the attention of the data-mining community, fueled by the seminal work by Domingos and Richardson [46] and Kempe *et al.* [90]. The main computational problems in this area are: (*i*) distinguishing genuine social influence from "homophily" and other factors of correlation [3, 40, 53]; (*ii*) measuring the strength of social influence over each social link [61, 129]; and (*iii*) discovering a set of influential users [46, 90, 62]. Finally, a large amount of literature exists on the analysis of social influence in specific domains: for instance, studying person-to-person recommendation for purchasing books and videos [97], telecommunication services [72], or studying information cascades driven by social influence in Twitter [7, 125]. Session summarization can also find applications in the field of *website usage analysis and re-organization* [139]. Typical browsing patterns can be exploited for reorganizing a website, creating *quicklinks* [27], and, in general, making the navigation in the website more efficient for the users.

Difference from Previous Work. The problem we study in this thesis departs from graph pattern mining, as we do not mine frequent substructures from a set of graphs. Rather we look for sets of graphs that satisfy certain requirements when merged together. Our problem is evidently different from graph summarization. The output of graph summarization is a reduced version of a single graph, whereas our summaries are sets of structurally-similar graphs.

2.3. Models of Browsing Sessions

In general, techniques to model web user navigation patterns usually operate on a per-session or a per-user basis, and usually the deployed models are intertwined with clustering techniques to identify and group similar users or navigation patterns. Some proposed approaches are based on Markov processes [23, 104], hidden Markov models [167, 45], or relational hidden Markov models [4]. Models have been built for general browsing [109], tabbed browsing [34], parallel browsing of multiple websites [117], or to predict when the user is likely to stop browsing [130].

Other researchers focus on user intent [59], behavior [69, 73, 104, 167], implicit feedback [87, 89], or modeling usability and interaction [42, 43, 158]. Choo *et al.* [36] present an integrated model for browsing and searching on the web. SNIF-ACT [118] is a computational cognitive model to explain navigation behavior on the World Wide Web.

Some work focuses on visualizing [23], discovering [29], and in gaining insights from navigation patterns [16, 116], while some research focuses on modeling the behavior of users pursuing specific known information seeking tasks [118, 158].

Finally, several techniques have been developed in the context of news [17, 42, 98]. Billsus and Pazzani [17] model short-term changes in the behavior of users using a hybrid user model composed of two parts: a short-term component based on k-nearest-neighbor, aimed at understanding user interest in stories similar to the ones she has already read, and a Naive Bayes classifier that builds a model of the user based on the words and features that guide her interests. Researchers devised models that are able to predict future popularity of content, based on past data [141] or favorites and ratings [28], or signals from social media [25, 2, 8].

Difference from Previous Work. This thesis models user browsing behavior on a per-session basis. Our work differs from previous model-based clustering approaches [4, 23, 104, 167] that rely solely on the order in which web pages are requested. We model not only what content the user consumes but also the context in which he or she operates and the interaction with the website.

2.4. Clustering of Browsing Sessions

Hassan and Karim [69] evaluate the impact of clustering on the performance of predicting page views. Using a heuristic-based clustering method instead of a model-based one, they arrive at the conclusion that multiple clusters do not benefit accuracy. Other researchers study methods to evaluate the quality of clustered user models and model-based recommendations. Li *et al.* [98] investigate offline evaluation of contextual-bandit-based news article recommendation algorithms. Pallis *et al.* [116] develop a statistical test to measure the difference between clusters, obtained by clustering according to Markov process parameters, which is then also used to visualize the model. In this way, clusterings can be validated, however without regard to the behavior's context. Although our framework can be applied for behavioral analysis, visualization, and for gaining insights on user behavior, it is in general closest to approaches based on clustering. Therefore, we discuss those in further detail.

Often, approaches for modeling user behavior focus on deriving user-based models and estimating personalized stochastic processes from historic user data (*e.g.*, [73, 29], Markov processes such as those mentioned above, and sequence alignment-based methods [71]). Other methods include relational models [4], association rule mining [42, 43], and higher-order Markov models [45]. Hoebel and Zicari [73] cluster website-visitors using a combination of hierarchical clustering with a heuristic centroid-based criterion, aiming at discovering groups of users with similar interests in several topics, while Gündüz and Özsu [64] define a similarity measure among navigation sessions and cluster them using a graph-based approach. For every cluster, a click-stream-tree is constructed and used for recommendation. Finally, Haider *et al.* [67] describe a discriminative clustering method for market segmentation on Yahoo News. Instead of aiming to understand navigation behavior as in this thesis, their goal is to classify behavior instances into simultaneously optimized segments.

Difference from Previous Work. Our model extends Markov process-based clustering models by dynamically including context, and explicitly captures periodic behavior by using a time distribution that is a mixture of periodic Gaussians. In contrast to the results of Hassan and Karim [69], the fully probabilistic model we present in this thesis takes significant advantage of multiple clusters.

2.5. Applications

2.5.1. Photo Browsing Interfaces and Parallel Browsing

Various interfaces have been considered for image browsing. Fan *et al.* [50] describe *JustClick*, which recommends images via interactive exploratory search. They build a topic network based on Flickr tags, and propose an interactive interface that allows the user to express a query by selecting images. They perform experiments on a big Flickr dataset of 1.5 billion images with $4,000$ different topics. Xu *et al.* [160] present an innovative visual search interface based on topic clustering. Given the query and the results from a search engine, latent topics are detected and clustered and then the clusters are shown in an intuitive layout. Ren and Calic [123] present an interactive interface for browsing large-scale image collections. Their system is based on two main parts, an image clustering module and an interface generation component in order to retrieve the images in a more efficient way. Strong *et al.* [140] present an approach for browsing images based on conceptual and visual similarity, with the main benefit being that the displayed images are grouped together. Zavesky *et al.* [168] propose a new framework called *Visual Island*, a novel organization algorithm for interactively displaying results. The aim is to organize the images in order to improve human comprehensibility and reduce required inspection time.

With the growing popularity of tablet computers an increasing number of commercial and research applications have been devoted for these mid-sized touch screen, mobile devices. Research efforts related to image search in tablets include optimizing the use of the screen's real-state, typically limited in mobile devices [24], better organizing personal image collections [133], or exploring diverse location-based services [166]. There are however mobile applications such as PULSE[1] or FLUDE[2] where users can browse through their news feeds in parallel, using horizontal sliding strips.

Google's Image Swirl [86] arranges search results as an exemplar-hierarchy, based on the images' visual and semantic similarity. Using a balloon-tree layout, users can navigate the clusters selecting the different branches of the tree. Users can only explore one branch at a time and cannot define different criteria for branching their search.

As already seen in Section 2.1, the term parallel browsing has been used in the literature [78] to indicate that one navigation session may involve the

[1] http://www.pulse.me/

[2] http://www.flud.it/

exploration of a number of topics at the same time, usually through the use of the browser's tabs, or opening multiple windows of the browser. Then, when users search in parallel [143], they have to switch tabs or windows. Namely, the search is not actually *simultaneous*. Furthermore, results in a different tab or window are mutually agnostic, allowing for repetition in the results.

Difference from Previous Work. We propose a recommender system that is well integrated in the standard photo-browsing interface and uses only anonymous browsing and content data. To the best of our knowledge, however, no application has addressed the parallel image search paradigm, neither for tablet computers nor for desktop applications.

2.5.2. People Interaction Exploration Interfaces

Tools to visualize and explore interactions between entities in time tend to focus either on the structural or the temporal dimension. On one hand, tools to animate *dynamic graphs* [164, 10] can visualize the evolution of the whole set of interactions in the system, but they do not provide a way to explore the history of relations. On the other hand, *timelines* [119, 114] and their variants, such as stacked lines charts and stream graphs [70], foster the exploratory visualization of temporal data by explicitly displaying temporal sequences of events as lines on a reference plane. However, being focused on the representation of temporal information only, the interaction between entities is not easily represented in such displays.

Attempts to produce visualizations between these two extremes have been made in the past. Tools for the exploration of genealogical data explicitly represent both time and interactions, but are bound to the visual paradigm of the tree [15]. Visualizations with *metro maps* [135, 134] allow a more generic layout, but relax the constraint on time representation, being more similar to graphs than timelines. *Alluvial diagrams* are used to represent changes in network structure over time [127]. In such representations, each line is a cluster of entities and one can see how entities move across them in time. *TimeNets* is a tool for genealogical data visualization [91]. People's lives are represented on a horizontal timeline as lines spanning from the year of birth to the year of death. Lines of different people join and split correspond to weddings or separations.

Difference from Previous Work. In *Metro*, the interface presented in this thesis, the focus is on entities and their interactions rather than on clusters. *Metro* is different from *Timenets* since it allows exploration by query and is not tailored just towards genealogical data, where the interactions between people are few and, on average, span a long period of time (*e.g.*, the duration of two people's marriage).

Chapter 3

Background

In this chapter we present some elements that are useful throughout the thesis.

3.1. Browsing Session

When a person is visiting a website, he or she may visit more than one page. It makes sense to think that such pages are interdependent, *i.e.*, which page the user will visit next depends on the previous one. Therefore, when a user visits a service provider, this interaction takes the form of a dialogue in which the two parts exchange information. We can therefore group entries in the server logs into *browsing sessions*. Borrowing the definition made by Huang *et al.* [79], a browsing session is a group of requests made by a single user for a single navigation purpose. The most common way to identify sessions is by means of a timeout, but more elaborated ways have also been devised (*e.g.*, [79]).

There are mainly two ways of looking at sessions: *a*) as a *set* (or bag): a session is just a set of items in which the order of the items is not important; or *b*) as an ordered *list*: a session is a list of items, in which the order matters. The choice influences the methodologies used in the analysis. In the first case, discarding the order makes the complexity of the problem lower but may lose information. In the latter case, the complexity becomes larger but sequential pattern mining becomes a useful tool for the analysis.

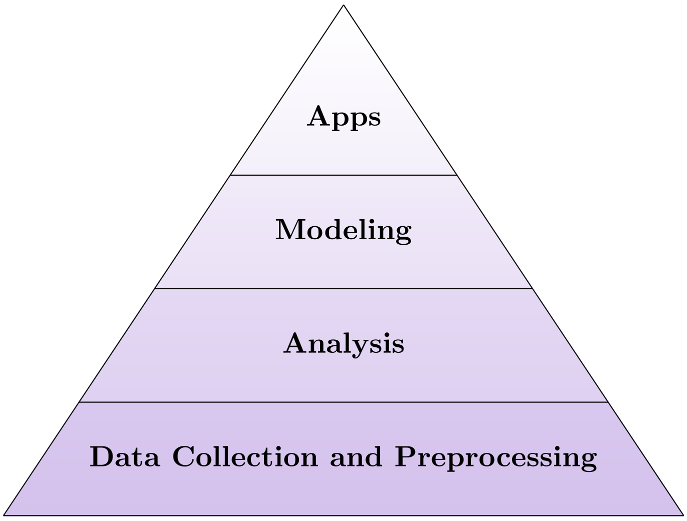


Figure 3.1: The framework of the thesis. The pyramid represents the bottom-up approach taken in the thesis.

3.2. Methodology

In this section, we describe the methodology adopted while developing this thesis.

In general, our approach is bottom-up and data-driven. This means that the starting point is the data itself. Since each dataset may have different characteristics, all methods and applications depend on it. Our path of understanding of browsing behavior is inspired by the "Knowledge Discovery in Database" process [51]. Figure 3.1 condenses the hierarchy of the thesis. Each level is summarized below.

Data Collection and Preprocessing. The first phase is the *data collection.* We collect server logs from the servers. The logs are very large since they contain actions of millions of users, and the collection may be space and time demanding. We make use of distributed systems, namely Apache Hadoop,[1] to process the large amount of data.

We then perform *preprocessing and filtering* on the raw data. This phase is aimed at parsing the data, removing errors or inconsistencies that may appear, and extracting the segment of the data we are interested in. For

[1] http://hadoop.apache.org/

example, we could focus on a particular section of the website or in a particular segment of users.

Analysis. Next, we perform the *data exploration.* This phase is aimed at understanding the coarse-grained characteristics of the data by manually exploring it. Due to the large size of the datasets, data visualization techniques are very useful since they allow a compact and interpretable view. During the data exploration, the researcher poses simple hypothesis and validate them in the data.

Based on the knowledge acquired during the data exploration, we use data mining techniques to extract frequent browsing patterns, which we call *summaries.* We evaluate the approach quantitatively and qualitatively. Quantitative evaluation is aimed at assessing the efficiency of the algorithm, while qualitative evaluation validates its soundness by presenting examples of output.

Even if we use relatively complex techniques as in the case of pattern mining, this level of the pyramid does not abstract from the data. All the results, including the summaries, are given in terms of the data points themselves. The next level, on the contrary, uses models create an abstraction from the raw data.

Modeling. We then model browsing sessions using probabilistic generative models. For each of these models, we evaluate the performance using automated tests, in the sense that do not involve testing with users. In Information Retrieval and Machine Learning, there are many automatic tests designed for different situations. In general, they consist in splitting the data in two sets: a *training* and a *test* sets. The training set is used to train the model and learn its parameters. The test set is used to evaluate the performance of the learned model. Since server logs are ordered in time, the datasets should be split chronologically into training and test sets. K-fold validation and other random split evaluation methodologies are therefore not applicable in this case.

Apps. The last level is the level of applications. We tried to look beyond the models and traditional structures of browsing sessions. We presented interfaces to browse images using completely new paradigms, *e.g.*, parallel browsing in the case of *PRiSMA*.

Since automated testing is not possible, we perform user studies. To emulate the behavior of a person browsing a website, the study is performed in an environment in which the user is free to browse the content for a limited amount of time. We then perform individual semi-structured interviews.

3.3. Datasets

We now introduce the datasets used in the experiments. There are in total 8 datasets:

- `FlickrBrowsing` (Section 3.3.2): a set of browsing sessions extracted from Flickr.[2] To reduce the sparsity of the data, we constructed a categorization of the external domain from the session starts and a categorization of the layouts of pages in Flickr;
- `YahooNewsBrowsing-UK` (Section 3.3.3) and `YahooNewsBrowsing-USA` (Section 3.3.4): two sets of browsing sessions extracted from Yahoo News;[3]
- `Twitter` (Section 3.3.5): a sample of the propagation network of popular tweets in Twitter;[4]
- `Last.fm` (Section 3.3.6): a sample of the social network and songs listened by users in Last.fm;[5]
- `Flixster` (Section 3.3.7): a sample of the social network and movies watched by users in Flixter;[6]
- `WikipediaBrowsing` (Section 3.3.8): a set of browsing sessions of users in Wikipedia;[7]
- `GettyImages` (Section 3.3.9): a set of photos and metadata from Getty Images.[8]

Before describing the datasets, we introduce the method we use to identify sessions from the server logs.

[2] http://www.flickr.com/
[3] http://news.yahoo.com/
[4] http://www.twitter.com/
[5] http://www.last.fm/
[6] http://www.flixster.com/
[7] http://www.wikipedia.com/
[8] http://www.getty.com/

3.3.1. Session Identification

Since user behavior varies over time, we group page views into *sessions*. We define a session as a sequence of click and page view events. While clicks realize transitions between web pages, page views encode intermediate events such as displaying an article or a picture. Depending on the use case, we may attach additional information to the clicks, to the page views, or to the entire session.

To perform our analysis, we split the activity of a single user into different sessions when either of these two conditions holds:

- *Timeout*: the inactivity between two page views is longer than 25 minutes. This value has been used in other works (*e.g.*, [26]), as well as in production systems.[9]

- *External URL*: if a user visiting Flickr leaves the site and returns from a different domain, the current session ends even if previous visits are within the 25 minute threshold (we make the assumption that if a user is viewing a page on Flickr and visits another domain, then the session ends).

3.3.2. `FlickrBrowsing` Dataset

The `FlickrBrowsing` dataset consists of a sample of the *page views* of more than 10 million anonymous users from approximately two months of Flickr user log data, from August to October 2011. The page views are represented as plain text files that contain a line for each HTTP request satisfied by the web server.

For each page view, our dataset contains the following fields: *a*) the *UserId* is a unique anonymized identifier computed from the Flickr user identifier in case of logged-in users and from a browser cookie otherwise; *b*) the *CurrentURL* and the *ReferrerURL* represent the current page the user is visiting and the page the user visited before; *c*) the *User-Agent* identifies the browser in use; and *d*) the *Timestamp* indicates when the page was visited. All the data processing was anonymous and used aggregated. Flickr allows users to set specific pages to “private”, so in our analysis we considered only public pages.

[9] https://support.google.com/analytics/answer/2731565

Page View Filtering and Data Selection. In order to obtain a coherent dataset in terms of both time zone and activity, we focused on users who were located in the United States (US) by extracting the location of the IP address from the source of the HTTP request and filtering out non-US locations. We then removed traffic derived from web crawlers by preserving only the entries whose User-Agent field contains a well-known browser identifier, namely Mozilla Firefox, Google Chrome, Apple Safari, and Opera Browser. In spite of this filtering, there are cases in which the User-Agent field indicates that a legitimate browser was used, but the corresponding "users" have a very large number of page views. The frequency, however, suggests that such server requests could not have been made by humans, but instead were done automatically for malicious crawling. We therefore apply an additional filter by which we set a maximum threshold on the total number of page views per user. The threshold is set to remove a small percentage of the users (1% of the total amount). After applying the filtering steps described above, our sample contains approximately 309 million page views.

We identify sessions as explained in Section 3.3.1 and extract a total of $40,446,676$ sessions from $10,912,431$ unique users.

3.3.3. `YahooNewsBrowsing-UK` Dataset

The `YahooNewsBrowsing-UK` dataset is large data sample from Yahoo News United Kingdom. We use a sample of data from June and July 2011 and use the former month for parameter estimation and the latter for evaluation. All processing is anonymous and aggregated.

We identify sessions as described in Section 3.3.1 and we attach additional information. More specifically, a session x of length M is formalized as a 5-tuple $x = (t, r, \vec{v}, \vec{s}, \vec{w})$, where t is the timestamp of the session, r is the referrer domain, $\vec{v} = v_1, \ldots, v_M$ and $\vec{s} = s_1, \ldots, s_{M-1}$ are sequences of page view categories and click locations, and $\vec{w} = w_1, \ldots, w_{M-1}$ are the clicked anchor texts in a vectorial bag-of-words representation. Since we only consider navigation clicks within the website, there is no click s_M associated to the last page view v_M. In addition:

- The location of a clicked link is s, which for simplicity is a discrete identifier that encodes either the clicked component (*e.g.*, widget, module, *etc.*) or area (*e.g.*, North, NorthWest). Other representations, such as relative/absolute (x, y) coordinates could also be used with appropriate distributions.

- The link anchor text is represented by a bag-of-words w. If no anchor text is associated with the link, then $w_i = \emptyset$.
- Every page view has a category $v_m \in \mathcal{C}$ where $\mathcal{C}$ contains a finite set of categories.

3.3.4. `YahooNewsBrowsing-USA` Dataset

The `YahooNewsBrowsing-USA` dataset is quite similar to the `YahooNewsBrowsing-UK` dataset (Section 3.3.3), with the difference that it is sampled from Yahoo News United States during the first months of 2014 and contains slightly different information.

The data contains all actions executed by the user on the website, including page views (*i.e.*, when an user requests a page from the server), shares (*i.e.*, when a user shares a page on a social media platform), and comments (*i.e.*, when a user comments on a news article).

For each action, we collect the following data: *a*) *Timestamp*; *b*) *UserId*: the anonymized reference to the user; *c*) *CurrentURL* and *ReferrerURL*; *d*) *Age, Gender* as stated by the user when creating the profile; *e*) *Geolocation* of the request, aggregated at a regional level (*i.e.*, state of the United States or region for other countries); *f*) *Metadata*: category to which the article belongs. The category is organized in a hierarchical taxonomy, but we only consider the first level; and *g*) *Publication date*: the time and date in which the news article is published.

All data processing has been performed in aggregate and no personal information has been made available to the people involved in the data processing.

We then split all page view actions (*i.e.*, discarding share and comment actions) into sessions, with the same method described in Section 3.3.1. Table 3.1 (first block) shows statistics on the sessions in the dataset.

Session Filtering. Among all browsing sessions, some are very short and thus cannot be considered as browsing. Indeed, it often happens that people enter news portals to read a single article and then leave. Therefore, we preprocess the dataset in order to remove such short visits.

Number of users	11 M
Number of sessions	284 M
Avg sessions per user per day	4.56
Avg distinct articles per session	1.33
Referrer sessions	85 M
0-page sessions	30 M
1-page sessions	83 M
Remaining	87 M

Table 3.1: Statistics on the `YahooNewsBrowsing-USA` dataset and on the preprocessing.

We identify three possible cases:

- Referrer sessions: these sessions originate from people opening one or more news articles from another website, without clicking on any other article on the site. This happens for example, when a user opens a URL shared by a friend from a social networking site.
- 0-page sessions: Sessions in which users browse the sections of the news website without opening any article page.
- 1-page sessions: Sessions in which users browse the sections of the news website opening a single article page. Manual inspection of 1-page sessions showed that they are mainly aimed at checking the most recent news article, thus do not contain proper browsing.

We filter the dataset and keep only the sessions that do not meet any of the conditions above. Table 3.1 (lower block) shows the amount of sessions falling in each category.

3.3.5. `Twitter` Dataset

We obtain the dataset by crawling the public timeline of the popular online microblogging service. The nodes of the graph G are the Twitter users, while each arc (u, v) expresses the fact that v is a follower of u. We additionally

	$\lvert\mathbb{O}\rvert$	$\lvert V\rvert$	$\lvert A\rvert$
`Twitter`	$580,141$	$28,185$	$1,636,451$
`Last.fm`	$1,208,640$	$1,372$	$14,708$
`Flixster`	$6,529,012$	$29,357$	$425,228$
`WikipediaBrowsing`	$50,092$	$39,756$	$46,610$

Table 3.2: Number of observations ($\lvert\mathbb{O}\rvert$), nodes ($\lvert V\rvert$), and arcs ($\lvert A\rvert$) in the input graph G for `Twitter`, `Last.fm`, `Flixster`, and `WikipediaBrowsing` datasets.

crawl a set of entities $\mathcal{E}$ and a set of observations of their propagation in the graph $\mathbb{O}$. The entities in $\mathcal{E}$ correspond to URLs share on Twitter, while an observation $\langle u, \phi, t\rangle \in \mathbb{O}$ means that the user u (re-)tweets (for the first time) the URL ϕ at time t. Table 3.2 shows the characteristics of the `Twitter` and the next three datasets.

3.3.6. `Last.fm` Dataset

Last.fm is a music website where users listen to their favorite tracks and communicate with each other. The dataset is created starting from the *HetRec 2011 Workshop* dataset available at `http://www.grouplens.org/node/462/`, and enriching it by crawling. The graph G corresponds to the friendship graph of the service. Similarly to the `Twitter` dataset, the entities in $\mathcal{E}$ are the songs listened by the users. An observation $\langle u, \phi, t\rangle \in \mathbb{O}$ means that the first time that the user u listens to the song ϕ happens at time t.

3.3.7. `Flixster` Dataset

Flixster is a social movie site where people can meet each other based on tastes in movies. The graph G corresponds to the social network underlying the site. The entities in $\mathcal{E}$ are movies, and an observation $\langle u, \phi, t\rangle$ is included in $\mathbb{O}$ when the user u rates for the first time the movie ϕ with the rating happening at time t. We do not take into account the value of the ratings since we are only interested in the propagation of movies in the social network.

3.3.8. `WikipediaBrowsing` Dataset

We create this dataset by looking at the browsing sessions of the popular online encyclopedia. Any node of the graph G corresponds to a Wikipedia page, while an arc (u, v) is present in G if there exists a browsing session where the page v has been reached from the page u, even making use of intermediate pages external to the website. An entity $\phi \in \mathcal{E}$ corresponds to a browsing session. An observation $\langle u, \phi, t\rangle \in \mathbb{O}$ means that the page u is visited during the session ϕ at time t. For each page, we consider only the first visit among the multiple ones possibly performed within the same session.

3.3.9. `GettyImages` Dataset

The `GettyImages` dataset is powered by the data of approximately 9 million images, taken between years 2000 to 2011, from a well-known stock photo agency [115]. Photo metadata include a set of keywords defining the *people* depicted in the photo and possibly the identifier and description of the *event* the photo relates to, for a total of around 45 thousand unique person names and 420 thousand unique events. To enhance the visualization with more accurate people description, we crawl the Wikipedia information of all the available person names. Among all the people in our corpus, 78% have a Wikipedia page.

In the dataset, almost all events are at most one day long. Therefore it makes sense to consider an event as a particular day. Among all 229 thousand days with events, 156 thousand have only one event per person, 36 thousand have two, and 15.6 thousand have three. It is therefore not rare to see the same person appearing in more than one event per day. In terms of photos, 91.5 thousand events only contain one photo per person, 51 thousands contain 2, and 36 thousands contain 3. In terms of participation, we see an average of 251.2 events per person and an average of around 2 people per event. Events do not have an explicit description in the data but for most events (346 thousands) all photos share the same title. Therefore we compute the event's description as the title that appears most often among its photographs.

3.4. Categorizations

In this section we describe the categorizations of URLs, which we use to reduce their sparsity in the `FlickrBrowsing` dataset (Section 3.3.2).

Source category	%
search: search.yahoo.com, google.com, *etc.*	34.87
social: facebook.com, tumblr.com, *etc.*	26.95
mail: mail.yahoo.com, gmail.com, *etc.*	13.22
aggregator: reddit.com, stumbleupon.com, *etc.*	7.76
blog: blogspot.com, blogger.com, *etc.*	6.65
photo: flickrhivemind.net, compfight.com, *etc.*	2.32
microblog: twitter.com, *etc.*	2.26
forum: discussion forums	2.00
news: news.yahoo.com, cnn.com, *etc.*	1.67
shop: ebay.com, *etc.*	0.85

Table 3.3: Top ten most frequent source categories in the dataset.

3.4.1. Source URL Taxonomy

In order to analyze the referrer URLs (*i.e.*, the websites from where users arrive to Flickr), we built a taxonomy for external URLs (*i.e.*, whose domains are different from `www.flickr.com`). The first attempt of categorizing URLs was based on the Open Directory Project[10] and the Yahoo Directory.[11] However, by manually inspecting the results, we realized that the classification was too detailed and did not capture the aspects we were interested in. More specifically, URL categorization usually works by *topic* (*e.g.*, travel, economy, food, *etc.*) whereas in our study we are interested on a categorization by *type* (*e.g.*, blog, social networking site, search, *etc.*). We therefore opted to annotate them manually (*e.g.*, `search.google.com` as *search*, *etc.*) and focused on defining 15 categories that we considered important, called *source categories*. We created a set of regular expressions in order to identify about 210 different external URL domains. Table 3.3 shows the most frequent categories. For the complete list of categories, refer to Section A.1 in the Appendix.

[10] Netscape (AOL), "Open directory", `http://www.dmoz.org/`, June 1998.

[11] Yahoo, "Yahoo directory", `http://dir.yahoo.com/`, March 1995.

Page layout	Description	%
Display all user photos	Displays the photos of a user on a grid	26.71
Browse user photos	Displays full-page photo of a user and allows browsing to the next and previous photos	20.67
Browse user album	Displays full-page photo of an album and allows browsing to the next and previous photos	14.12
Display single photo	Displays full-size photo	7.22
Homepage	Home page of Flickr	5.60
View user albums	Lists the album of a user	4.59
Browse group photos	Displays full-page photo of a group and allows browsing to the next and previous photos	2.63
Search photos	Photo search in Flickr	2.38
Browse user fav.	Displays full-page photo of the favorite photos of a user and allows browsing to the next and previous photos	2.09
Group photos	Displays the photos of a group on a grid	1.79

Table 3.4: Top ten most frequent page layouts in the dataset.

3.4.2. Page View Layouts

In most websites, multiple URLs can map to exactly the same page "layout". For example, on Flickr, the URL of a page that shows a single image contains a unique identifier for the image, thus two URLs for two different images are different even though the page layout is the same. Since our interest is in modeling navigation patterns in Flickr, we must map all URLs that refer to the same layout to a single page layout (*e.g.*, "single image page"). For this purpose, we define a hierarchical taxonomy of URLs: the *page layout*. We manually created a set of regular expressions to classify the URLs to obtain a total of 96 different layouts. Examples of layouts include the following: *display all user photos*, *search photos*, *browse group photos*, *add contacts*, *accept invitation to join Flickr*, *etc.*Table 3.4 summarizes the most frequent page layouts. For the complete list of page layouts, refer to Section A.2 in the Appendix.

Chapter **4**

Birth of a Session

Insights into how users behave within a website or domain are extremely important in informing business decisions, in developing strategies to provide new functionalities, and in general for devising new algorithms that directly improve such services. For instance, having deep insights on what pages or sections are visited most and when, can be used not just to create better user models, but also to improve the design of such pages and the overall "flow" of the website (*e.g.*, by highlighting certain sections on particular page layouts).

Flickr has become a rich resource for research in multimedia, in large part because its clear copyright policies and APIs have facilitated the gathering and analysis of its data. While a lot is known about the data that resides in Flickr, there are not many insights into how people actually use Flickr, and, in particular, on their social navigation patterns.

As the functionality of the Web has become more complex, and sharing of content (*e.g.*, Flickr photos) is done in multiple ways (*e.g.*, by posting to social networks such as Facebook, or information networks such as Twitter; posting on blogs, in news articles, *etc.*), it has become increasingly more difficult to understand the dynamics of how users browse and look at (*i.e.*, "consume") photos once they arrive at Flickr from other sources. Although Flickr remains very popular, there are many similar services for social sharing and viewing of photos, thus the work we present here should provide insights that, although computed from Flickr data, should easily generalize.

In this chapter, we perform an in-depth analysis of social navigation patterns on Flickr. In particular, we analyze a sample of user logs from approximately two months, by clustering sessions, and specifically considering the referrer domain (*i.e.*, the site or domain the user visited before arriving at Flickr).

Our work aims at addressing several questions, among which we include the following: *a*) are photo social navigation patterns different depending on the referrer website? if there are differences, what kinds of differences are there? *b*) do similar types of websites (*e.g.*, "search" lead to similar behavior?); *c*) what types of pages (*e.g.*, within Flickr) are more popular depending on the referrer website?; and *d*) does user behavior in terms of time spent vary depending on the referrer website?

Although there is a reasonable amount of published work on Flickr, with a few exceptions [96, 52], there is little knowledge on how users actually behave within the service and the relationship between such behavior and the referrer pages. Our main contribution is thus providing insights into social photo navigation patterns. Such insights may be useful for understanding the dynamics of photo-sharing sites, although the same type of analysis could be extended to other domains.

The results presented in this chapter were published in [31].

4.1. Session Analysis

For our analysis we use the `FlickrBrowsing` dataset (Section 3.3.2). The number of distinct types of page layouts present in the sessions is 1.83. The value suggests that a large number of sessions tend to consist of only a few page categories. However, the "complex" use of Flickr is not infrequent, proven by the fact that there are sessions visiting many page layouts, up to a maximum of 39 different page layouts per session.

Table 3.4 shows the ten most visited page layouts in the dataset. We can see that there are a few page layouts that are visited most frequently: although we define a total of 96 page layouts, users tend to navigate through a small subset of them, namely to explore *photos of users* and *groups*. Indeed, the top 10 layouts account for 87, 80% of all page views. This is compatible with the small average number of distinct page layouts per session, which shows that users usually browse in just a few categories during one session. We will now move our focus to the *source URL*, which is the referrer of the session.

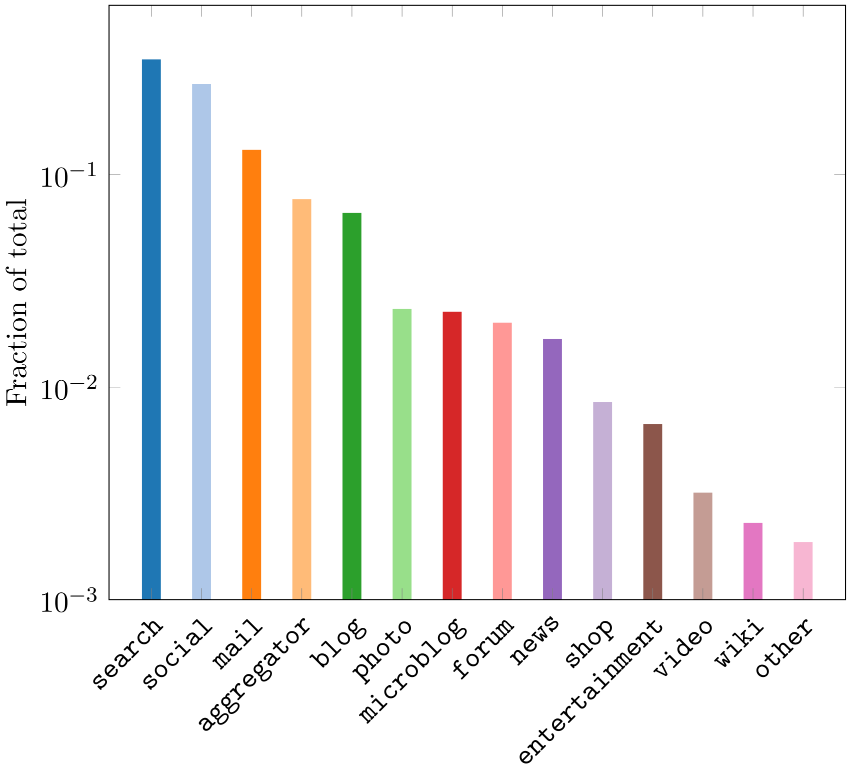


Figure 4.1: Distribution of the 14 top categories for the external source URLs.

4.2. Source URL Category Analysis

One of our main assumptions is that there is a relationship between the source URL and the type of navigation behavior of the user.

In Table 3.3 we show the most frequent domain categories from which the user arrives to Flickr pages and some example URLs. The histogram in Figure 4.1 shows the distribution of the source URL categories. The two most frequent sources are *search* and *social*. The presence of search is reasonable due to the contribution of image search and navigational queries. While most photo websites retain proprietary rights on the retrieved results or do not have clear photo licensing policies, we can assume that Flickr is one of the main sources of Creative Commons-licensed material. We will confirm this assumption later in the chapter (Section 4.3.2). Social network websites, such as Facebook, constitute very popular access points

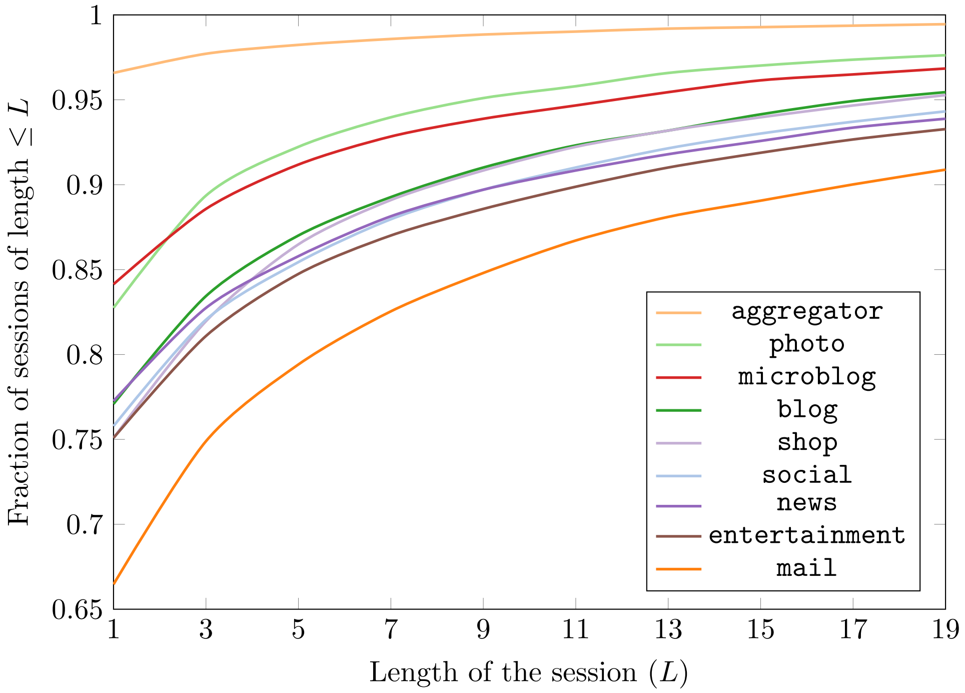


Figure 4.2: Cumulative distribution of the nine most popular categories of source URLs.

to Flickr since users are highly interested in photos shared by friends. We did not expect *mail* to have high importance, as usually the attachments are sent within the message itself and not as external links. As we will see in Section 4.3.2, many sessions derive from invitations of friends to join Flickr. The fact that many sessions come from the *news* domain is indicative that the image is often considered as appealing or significant as the actual text of the article.

The raw analysis of volume gives us the first insight into how the initial context may affect navigation patterns. However, we understand this even better by observing the cumulative distribution of session lengths given in Figure 4.2. In the figure we represent only the 9 most frequent categories. The categories have a different behavior from one another. The lines that reach value 1 sooner correspond to the situation in which the user spends less time on Flickr on average. On the contrary, the ones that grow slower correspond to longer sessions on average. We see that the shortest sessions originate from aggregators. One example is http://www.reddit.com/, in which the links to Flickr appear inside news posts.

It may appear strange that the sessions deriving from *news* sites last longer. An explanation for this might be that the visual material in news sites (such as Yahoo News) is curated by professional editors and photographers and often consists not only of a single photo, but also of a collection of photos related to a particular event. For example, an article about the earthquake in Japan is linked to a group or a set of photos all related to that topic. The user is therefore prone to see more than one picture.

Extreme behavior is observed in the *mail* category where the users spend the longest time interacting with Flickr. One possible explanation might be that only the "closest" contacts send e-mail, and thus a stronger bond exists between the sender and the receiver of the message. Moreover one could assume that users that share links via e-mail, may share entire sets or albums, which contain many photos, leading to longer and more complex interactions with Flickr.

4.3. Clustering of Sessions

In this section we describe the clustering of sessions and analyze the clusters' general characteristics in terms of the page layouts (see Section 3.4.2) and in terms of browsing behavior depending on the referrer domain categories (*i.e.*, the type of domain that users visit before arriving at Flickr).

We model each session s as a vector $v = (v_1, v_2, ..., v_P)$ where each v_i counts the views of page layout i in session s. *Cosine similarity* is used to compare vectors since it is not affected by the absolute number of page views but only by the relative distribution across the page layouts. We apply the Canopy algorithm [106] on the vectors to initialize the centroids. We choose empirically the parameters ($T_1 = 60$ and $T_2 = 40$). Then, we run K-Means clustering to extract clusters of sessions to obtain a total of 62 clusters.

4.3.1. Patterns in Session Clusters

Although our hypothesis is that user browsing patterns are different depending on the source website, we first examine session clusters without taking into account how users arrived at Flickr. We will then remove this constraint in Section 4.3.2.

	Cluster						
Page layout	k_1	k_2	k_3	k_4	k_5	k_6	k_7
`Browse user fav.`	0	0	0	0.02	0	0	0
`Photos of group`	0	0	0	0	0	0	0.06
`Browse user album`	0	0	0	0	0.92	0.28	0
`Browse user photos`	0.05	0.06	0.38	0	0	0	0
`View user albums`	0.02	0	0	0.02	0.06	0.61	0
`Search photos`	0	0	0.02	0	0	0	0
`User profile`	0.02	0	0	0.03	0	0	0.08
`Add contact`	0	0	0	0.57	0	0	0
`Display single photo`	0	0	0.02	0.05	0	0	0.17
`Group page`	0	0	0	0	0	0	0.53
`Display all user photos`	0.43	0.81	0.46	0.08	0.02	0.03	0.05
`Homepage`	0.11	0.02	0.03	0.03	0	0.02	0.03

Figure 4.3: Heat-map of $p(layout \mid k)$ for the most frequent clusters. Darker squares indicate a higher presence of the relative page layout views (row) in the current cluster (column).

We want to extract general behavior of users browsing Flickr independently from the referrer URL. For this purpose, we focus on clusters that are generated in the same proportion by all source categories *source*. They capture actions that people do in Flickr that can be accounted as common use.

More specifically, we compute the entropy distribution for each cluster k across $p(k \mid source)$ with the following equation:

$$\sum_{source} \left[p(k \mid source) \log_2 p(k \mid source) \right]$$

We then sort the clusters in ascending order and select the 7 clusters with smallest entropy. These clusters represent the behaviors of users accessing Flickr in the same percentage from each source category. In order to understand the characteristics, we draw the heat-map of $p(layout \mid k)$ and the page layouts that constitute them in Figure 4.3.

As Figure 4.3 shows, k_1, k_2 and k_3 contain a large number of *Display all user photos* and *Browse user photos* page views, which indicates browsing through the photos of one or more users. Cluster k_4, on the other hand, contains more cases of users that import and add new contacts (*Add contact* row in Figure 4.3). A very clear case of browsing photo albums is cluster k_5, where we can observe a large value in the *Browse user album* row. A similar behavior is in cluster k_6 where the sessions are more balanced between browsing a specific album (*Browse user album*) and seeing the list of albums (*View user albums*), maybe to explore a different one. Group-oriented navigation is specific of k_7, due to the presence of *Group page* and *Photos of group*. In this case users switch between the main page of the group and its photos.

Although these clusters are useful to understand how users interact with Flickr, we would like to explore the peculiarities of the source URL categories. We therefore manually inspect the clusters and select the ones that show interesting patterns.

4.3.2. Browsing from Different Sources

As stated earlier, many clusters illustrate a very specific browsing behavior. We manually pick a few of them to show how well they describe some navigation patterns in relation with the source categories.

Figure 4.4a shows the distribution of such clusters across source categories whereas Figure 4.4b shows the distribution of the same clusters across page layouts. Due to the large amount of sessions originated from search engines, the *search* source category appears in most of the clusters. Despite this, there are still some clusters in which this is not the case.

Cluster k_7 shows a large contribution of *news* and *search* and the distribution of page layouts for that cluster (first column of Figure 4.4b) is biased towards browsing of groups (*Group page*). This suggests that news editors embed sets of images into the article page. Moreover, photos of the same event are likely to be organized in the same group in Flickr. Cluster k_8, more evenly spread across all source categories, but still showing a predominance of *search*, is similar to k_7 but favors browsing through the photos of a group (*Photos of group*) on the home page of the group (*Group page*). Cluster k_9 contains sessions coming from both *search* and *aggregators* in which users

	Cluster						
Source category	k_7	k_8	k_9	k_{10}	k_{11}	k_{12}	k_{13}
aggregator	0.01	0.05	0.25	0.06	0	0.02	0.82
blog	0.06	0.07	0.03	0.02	0.02	0.01	0.01
mail	0.04	0.03	0.02	0.06	0.01	0.84	0
news	0.23	0.04	0	0	0	0	0
photo	0.01	0.03	0.01	0.01	0.01	0.02	0.01
search	0.57	0.69	0.62	0.77	0.94	0.06	0.13
shop	0.02	0.01	0	0	0	0	0
social	0.04	0.05	0.05	0.07	0.01	0.04	0.01

(a) Heat-map of source URLs, $p(source \mid k)$.

	Cluster						
Page layout	k_7	k_8	k_9	k_{10}	k_{11}	k_{12}	k_{13}
Manage friends	0	0	0	0	0	0.29	0
Browse user fav.	0	0	0	0.41	0	0	0
Photos of group	0.07	0.5	0	0.01	0	0	0
Photo fans	0	0	0	0.02	0	0	0
Search CC photos	0	0	0	0	0.38	0	0
Browse user tags	0	0	0.4	0	0	0	0
Search photos	0	0.01	0.01	0	0.23	0	0
Add friend	0	0	0	0	0	0.33	0
Group page	0.53	0.07	0	0.01	0	0	0
Recent activity	0	0	0	0	0	0	0.7

(b) Heat-map of page layouts, $p(layout \mid k)$.

Figure 4.4: Heat-map of the most interesting clusters. Darker squares indicate higher values for the presence of sessions with that category (row) in the relative cluster (column).

visualize the tag cloud of photo tags used by another user. This visualization gives an aggregated vision of the content posted by her. Cluster k_{10}, mainly originated from *search*, explores the list of favorite photos of a user (*Browse user favorites*).

Cluster k_{11} in Figure 4.4b contains mainly *search* page layouts. It is not surprising that Figure 4.4a shows us that those sessions originate from search engines. One assumption is that in this case users are migrating the search task to Flickr in order to take advantage of the image search features, as for instance filtering photos by Creative Commons license or tags.

Cluster k_{12} shows *mail* as a principal source and is composed of page layouts related to social actions: *a*) *Manage friends* is the set of all pages related to adding, editing or removing information about contacts in Flickr; *b*) *Add friend* is the page in which the user is asked for confirmation when adding a contact. Manual inspection of the sessions suggests that the traffic in this cluster mainly derives from accepted invitation mails sent to mail contacts. We do not examine mail contents, so this hypothesis cannot be verified, and is based solely on the aggregate views of the "add friend" page.

Sessions in cluster k_{13} are mainly originated from *aggregators* and are aimed at checking the recent activity on the Flickr website (*i.e.*, recently added photos, albums, *etc.*). Indeed, aggregators are used by the user to get an overview of recent events in external websites, including Flickr.

The remaining clusters have been inspected but are not listed since they do not show interesting characteristics.

4.4. Discussion

Our analysis shows that users arrive to Flickr from a variety of source domains (*e.g.*, search, social, mail, aggregators, *etc.*) and that the overall length of the sessions depends on the source domain (*e.g.*, users that arrive at Flickr from mail domains tend to spend more time than those arriving from any other sources). While the distribution of visits from different sources gives us interesting insights on the Web as it is today (*e.g.*, social sites have a prominent place), it is possible to make some observations on the behavior in terms of session length (*e.g.*, users that click on mail links may be receiving photos from close social contacts, which might explain longer sessions). At the same time, we found that clear session clusters can be observed in the data (*e.g.*, some sessions are very focused on viewing photos of users, while others focus on viewing photos in groups), and that some of the behavior can be intuitively explained (*e.g.*, sessions that originate in mail domains have a stronger focus on managing and adding friends).

Many similar observations can be made based on the figures presented in this chapter. Sessions that originate from search sites, for instance, cluster around the Flickr search functionality, suggesting that the user's main intent is indeed finding images of some sort. However it is important to keep in mind that such observations constitute hypotheses that need to be further examined.

In this chapter, we analyzed a sample from two months of Flickr user data. Our analysis was performed on user logs. We categorized pages within Flickr into specific categories, and analyzed how the behavior of users in viewing such page categories changes depending on the referrer domain (*i.e.*, the page they come from).

Up to this point, we have seen the birth of a session, *i.e.*, its first step in the website. We will now continue with the analysis of the evolution of browsing sessions after they are born.

Chapter 5

Life of a Session

This chapter analyzes how sessions evolve inside photo-sharing platforms. We focus on two tasks: *searching* and *browsing*.

First, we analyze how people search for images (Section 5.1). We show the differences between queries issued in photo-sharing platforms and the ones done in a general-purpose search engine. The insights provided in this section are intended as a launching point for the design of better interfaces and ranking models for image search.

In Section 5.2 we consider the second task, *i.e.*, browsing. We show that users tend to browse photos in sequences (called *photostreams*). We therefore consider photostreams as content units and build a large photostreams browsing graph. We then apply a clustering algorithm used in community detection to find interesting clusters of photostreams and describe their characteristics.

To conclude, in Section 5.3 we present an interactive visualization of user browsing traffic. The visualization is aimed at discovering interesting content by leveraging past user traffic. It is possible to see that the visualization is coherent with the results of Section 5.2. Browsing does indeed follow photostreams and is characterized by long sequences.

All analysis is performed on the `FlickrBrowsing` dataset (see Section 3.3.2).

The results presented in this chapter were published in [105, 33, 30].

5.1. Life of a Search Session

Photo-sharing platforms such as Flickr and Instagram are increasingly popular and, similarly to online social networks, they support activities such as sharing their photos with friends and forming common-interest *groups* where users can usually join freely to share multimedia content with other members. Such platforms also support image search; in Table 3.4 we showed that over 2% of page-views in Flickr are accounted for by searches, and an effective search performance is arguably important for the long-term success of such platforms.

If the goals of users in general web image search are not well understood, they are even less understood on photo-sharing platforms, where there little work on user search behavior has been published. On the other hand, the server logs of such platforms give us access to *entire* user search sessions, including all post search interactions, not just the *search* and *result click* interactions available in search engine logs. This gives us the opportunity to come to a deeper understanding of what users do after issuing a search.

In this section, we study the search behavior of users of a large online photo-sharing platform, namely Flickr. We study the typical types of search conducted on such platforms, and note some differences from general image search. We look at the entire user session after an initial keyword search, with a view to uncovering behavior patterns that go beyond simple “search and click on result” events.

5.1.1. Search Trees

Since sessions are not strictly linear in nature [34, 155], due to branching (use of browser tabs) and backtracking (the use of the “back” button) behavior, we represent search sessions as *search trees.* The first search action in a session is the root of the tree and, for each subsequent page view in the session, we create a node representing its URL class and add it as a child of the node representing its referrer URL. In the resulting tree, any leaf represents a termination of a browsing branch; this does not necessarily mean the end of a search session, as other branches can occur later. Although in some cases a single session can contain more than one tree, in the remainder of this section, for simplicity, we will use the terms *search sessions* and *search trees* interchangeably, *i.e.*, by *search session* we refer to a subtree within a session corresponding to search activities. To create a more compact representation, we collapse non-branching sequences of nodes of the same class

	Sessions	Trees	Chains
Total	1,071,954	1,017,037	1,622,329
Avg. width	-	1.815	-
Avg. depth	-	1.575	3.129
Unique types	-	109,693	108,255
Trees/session	-	1.053	-
Chains/tree	-	-	1.513

Table 5.1: Events, search trees and search chains in the dataset.

with the same URL parameters into a single node, ignoring differences in URL parameters in the following circumstances: the page number parameter for *search/next* nodes is ignored, and the photo id parameter is ignored for *photostream* nodes when two photos belong to the same photostream, and for *photo* nodes when two photos belong to the same set or group pool (indicating that the user is browsing within the same photostream, set, or pool). In this representation, we identify *search chains* (similar to *search trails* [157]) as the paths in a search tree that start at the root of the tree and end at a leaf.

5.1.2. Statistical Analysis

Table 5.1 summarizes some statistics about the *search trees* and *search chains* in our case. The *search tree* representation gives over $100K$ unique search trees, 95% of which have a depth of at most 3 and width of less than or equal to 4, and 95% of chains also have a length of 4 or less. For the remainder of the section, we will refer to distinct search trees as *tree types*.

In Figure 5.1 we plot the cumulative distribution of repetitions for several URL Classes (*i.e.*, how often a view of a certain page type is followed by a view of the same page type). We can see that in search sessions *photo* views are followed by other *photo* views less than 15% of the time, whereas *photostream* and *userphotos* nodes appear much more often one after the other. This suggests that, when in a *photostream* view, a user is likely to browse photos in this photostream. When a user enters the *photo* view after a search, however, they are unlikely to view other photos within the same set or pool: this is likely to be an artifact of the Flickr user interface at the

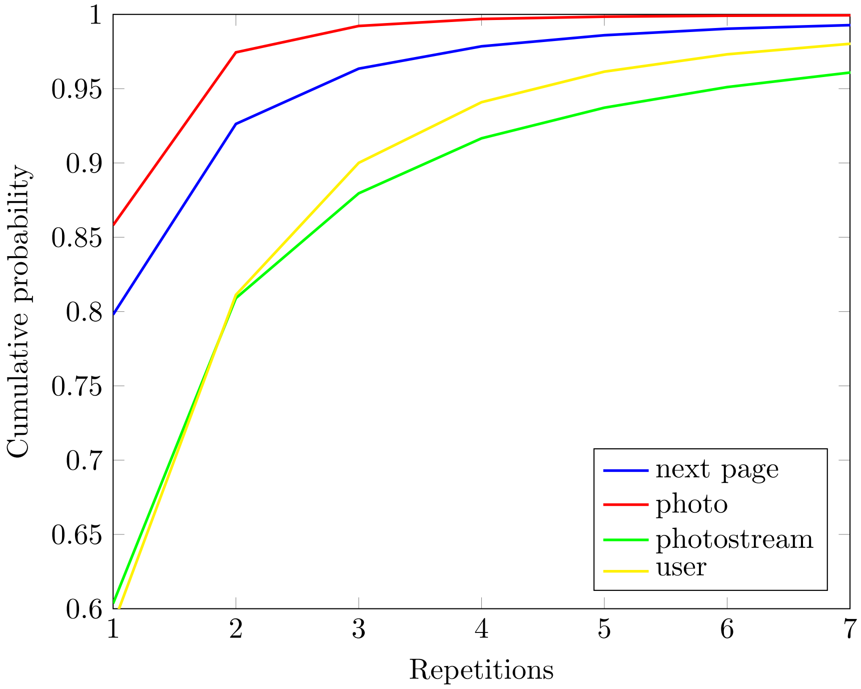
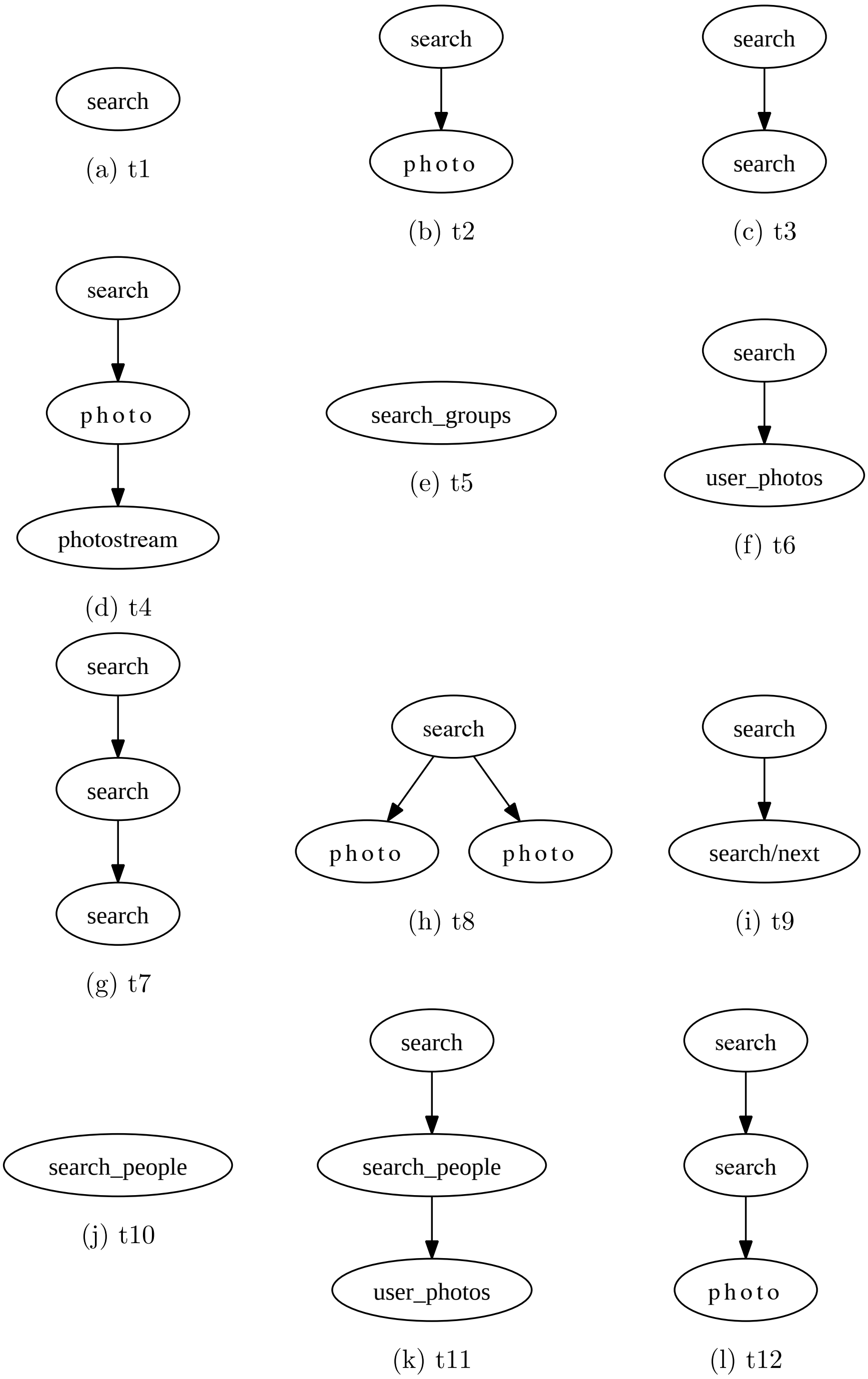


Figure 5.1: CDF plot of repetitions for several URL classes.

time of this study, which defaults to browsing a photostream, with options for browsing related sets or pools receiving less prominence in the interface. These patterns suggest that the user is browsing the results after the search, *e.g.*, the user viewing a sequence of individual photos (*photostream*), or a sequence of thumbnails (*user*).

Figure 5.2 shows the 12 most frequent search tree types, which give a succinct summary of the main user activities following a search. The percentage of sessions belonging to each tree type is shown in Table 5.2. The two most frequent search trees correspond to a search followed by no further action ($t1$), and a search followed by clicking on a single result ($t2$), that between them account for over 43% of the trees. Type $t1$ trees may represent searches where the user is “satisfied” with the first page of thumbnails; alternatively, they could be “failed searches”. Search reformulation is quite frequent ($t3$, $t7$, $t12$), as are browsing photostreams via a single photo view ($t4$) and searching groups ($t5$). Branching is relatively infrequent, as only 1 out of the top 12 (and 10 out of the top-50) cannot be represented as chains.



(a) t1 (b) t2 (c) t3 (d) t4 (e) t5 (f) t6 (g) t7 (h) t8 (i) t9 (j) t10 (k) t11 (l) t12

Figure 5.2: Most frequent search tree types.

Type	Description	%
$t1$	Search only.	36.0
$t2$	Result click.	7.3
$t3$	Query reformulation.	5.4
$t4$	Result + photostream.	2.5
$t5$	Search groups only.	1.8
$t6$	User profile click.	1.5
$t7$	Query reformulation.	1.5
$t8$	Click on 2 photos.	1.4
$t9$	Search + next page.	1.4
$t10$	Search people only.	1.1
$t11$	Search, search people + click.	1.0
$t12$	Search, next page + click.	0.9

Table 5.2: Distribution and description of the top-12 tree types illustrated in Figure 5.2.

5.1.3. Taxonomy of Image Search

Query taxonomies for image search differ from those used for web search. Some work [102] has attempted to adapt Broder's [22] web search taxonomy of intent, while others have classified queries based on the type of objects and concepts the query refers to. Enser [49] distinguishes between unique (*e.g.*, specific people) and non-unique queries, while Westman and Oittinen [154] follow the scheme of Shatford [137], and classify queries as queries for general objects, specific objects, and abstract queries. We broadly follow those taxonomies, and distinguish between general and specific queries, and introduce 2 additional categories that are specific to photo-sharing platforms:

- *General Queries*, which correspond to *non-unique search*, represent searches for items belonging to a certain category. As in Westman and Oittinen [154], we further sub-classify these as either being *objects*, or *concepts*.

Class	%	Subclass	Examples	%
General	47.2	objects	trees, mountains, tiger	27.5
		concepts	fashion, sports	19.7
Specific	35.7	places	san francisco	14.1
		events	burning man,	9.9
		products	iphone 4, geektool	6.1
		people	steve jobs, lady gaga	5.0
		organisation	nypd, lafd, fdny	0.6
Photography	12.8	photo equipment	fuji x100, nikon d7000	6.5
		photo techniques	bokeh, depth of field	5.5
		events	bc33, bc34	0.8
Meta	4.3	user/group names	-	3.4
		other	api key	0.9

Table 5.3: Taxonomy of annotated queries.

- *Specific Queries*, which correspond to *unique search*, represent searches for known-items, subcategorized by type: places, events, people, organizations, and products.

- *Photography Queries* are specific to photo-sharing platforms, and include searches for photo equipment and techniques, and for photography related events.[1]

- *Meta Queries* include searches for specific usernames and groups, and for site-specific Flickr features.

We manually annotated the $1,000$ most frequent queries from our corpus into this taxonomy. Queries that were ambiguous, or that do not clearly belong to this taxonomy, were labeled as "unknown", leaving 974 queries with known categories. From Table 5.3, we can see that 47.2% of queries

[1] Mainly comprised of photography "bootcamps" – events in which photographers meet for training purposes.

are *general*, 35.7% are *specific*, 12.6% are *photography* and 4.3% are *meta* queries. There are less *general* queries than is reported by Jansen [81], although that work focused on all queries, not just the most popular queries. Searches for people are much less important on photo-sharing platforms than has previously been reported, both for general web image search [145, 81] and in a journalistic context [154]. It is also noteworthy that *photography* accounts for 12.8% of popular searches, and that *meta* queries, which may not even be true image searches, account for over 4.3% of popular queries.

Search is an important task but it is not the only one when it comes to sessions in a multimedia platform such as Flickr. We will now move our focus to the life of a session outside search: browsing.

5.2. Life of a Browsing Session

Social media platforms such as Flickr provide a wide range of functionalities and different ways to share and view content. Given the sequential nature of browsing photographs, it is frequent for people to share and view images in sequences, whether the photos are arranged in galleries, slideshows, or in groups. In Flickr, in particular, photos uploaded by a user to his account are placed in a "photostream", which in essence is a sequence of photos. Although there are many ways to reach individual photographs, such sequences constitute a fundamental part of the interaction. In the rest of the section we will refer to such sequences as *photostreams* (or simply streams).

Furthermore, navigation across sequential units of content is present in other fields of social media, *e.g.*, social network, music streaming, and microblogging platforms. In popular social networks photos are organized in albums and can be viewed sequentially. Songs in music streaming services can be listened to one after another usually as a part of an album or a playlist. Posts in microblogging platforms are chronologically organized in independent blogs. Therefore, methods developed for photostreams could be adapted to other social media as well.

A key question for social media platforms, then, is how users navigate inside and between various photostreams. In particular, such photostreams may be considered not just as collections of images, but rather as fundamental units of content. On one hand, understanding how users navigate between specific photostreams is crucial in designing interfaces and algorithms that improve user experience, by providing the right content in the right places. On the other hand, analyzing the semantic categories of such streams can also provide important insights on general topics of interest. In addition, investigating how users transition between photostreams allows us to understand how topics may be related.

In this section, we treat photostreams as content units and analyze a large sample of navigation logs to gain insights into how users navigate between different photostreams. More specifically, we examine user navigation logs containing several millions page views in order to create a photostream transition graph to analyze frequent topic transitions (*e.g.*, users often view "train" followed by "firetruck" photostreams).

To the best of our knowledge, this is the first study that analyzes photostream browsing as opposed to photo browsing.

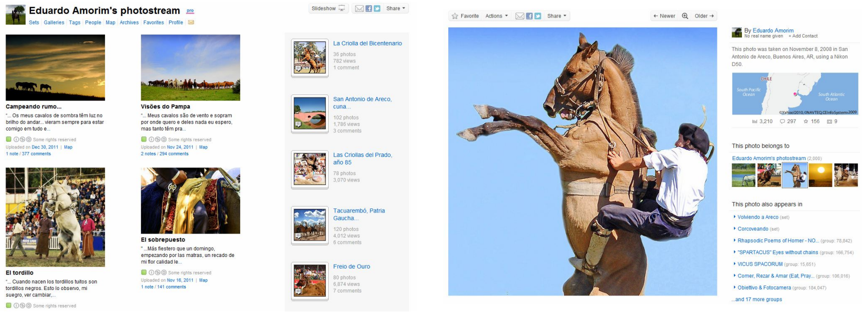
5.2.1. Analysis

In this subsection we define the main concepts of our study, present statistics on how users browse within sessions, and how they transition between photostreams.

Photostream Browsing. Photos in Flickr are organized in photostreams. Each photo in Flickr belongs to a photostream of the owner, but it can belong to other streams of photos as well: groups, sets, galleries, or favorites. Apart from favorites, all these photostreams are either chosen or created by the owner of the photo. Users always view and browse photos in the context of a particular photostream.

There are two main ways of viewing photostreams: *a*) *grid view*, *i.e.*, grid of photos from the stream (see Figure 5.3a), and *b*) *photo-focused view*, *i.e.*, a single zoomed-in photo with a possibility of browsing neighboring photos (see Figure 5.3b). Although Flickr allows different variations of grid views, they share a common feature, namely that they show several pictures from the browsed stream at a glance. The photo-focused view is the same for all the streams: it shows a large selected photo and, on the right side of it, thumbnails of 4 neighboring photos from the stream are presented, which the user can switch to by clicking on them. This way one can change the focus from the current photo to another one from the currently browsed stream. Below the thumbnails a list of all photostreams that the photo belongs to is shown in the form of hyperlinks, as visible in Figure 5.3b.

One can expect that users first enter the grid view of a photostream, and then select one of the photos they like and see it in a photo-focused view.



Eduardo Amorim's photostream

Campeando rumo...
"... Os meus cavalos de sombra têm luz no brilho do andar... vieram sempre para estar comigo em tudo e...
Some rights reserved
Uploaded on Dec 30, 2011 | Map
1 note / 377 comments

Visões do Pampa
"... Meus cavalos são de vento e sopram por onde quero e deles nada eu espero, mas tanto têm pra...
Some rights reserved
Uploaded on Nov 24, 2011 | Map
2 notes / 294 comments

El tordillo
"... Cuando nacen los tordillos tuitos son tordillos negros. Esto lo observo, mi suegro, ver cambiar,...
Some rights reserved

El sobrepuesto
" ...Más fiestero que un domingo, empezando por las matras, un recado de mi flor calidad le...
Some rights reserved
Uploaded on Nov 16, 2011 | Map
1 note / 141 comments

La Criolla del Bicentenario
36 photos
782 views
1 comment

San Antonio de Areco, cuna...
102 photos
1,786 views
3 comments

Las Criollas del Prado, año 85
78 photos
3,070 views

Tacuarembó, Patria Gaucha...
120 photos
4,012 views
6 comments

Freio de Ouro
80 photos
6,874 views
7 comments

By Eduardo Amorim
No real name given + Add Contact

This photo was taken on November 8, 2008 in San Antonio de Areco, Buenos Aires, AR, using a Nikon D50.

This photo belongs to

Eduardo Amorim's photostream (2,000)

This photo also appears in

- Volviendo a Areco (set)
- Corcoveando (set)
- Rhapsodic Poems of Homer - NO... (group: 78,842)
- "SPARTACUS" Eyes without chains (group: 166,754)
- VICUS SPACORUM (group: 15,651)
- Comer, Rezar & Amar (Eat, Pray... (group: 106,016)
- Obiettivo & Fotocamera (group: 184,047)

...and 17 more groups

(a) Grid view.

(b) Photo-focused view.

Figure 5.3: General types of stream views in Flickr.

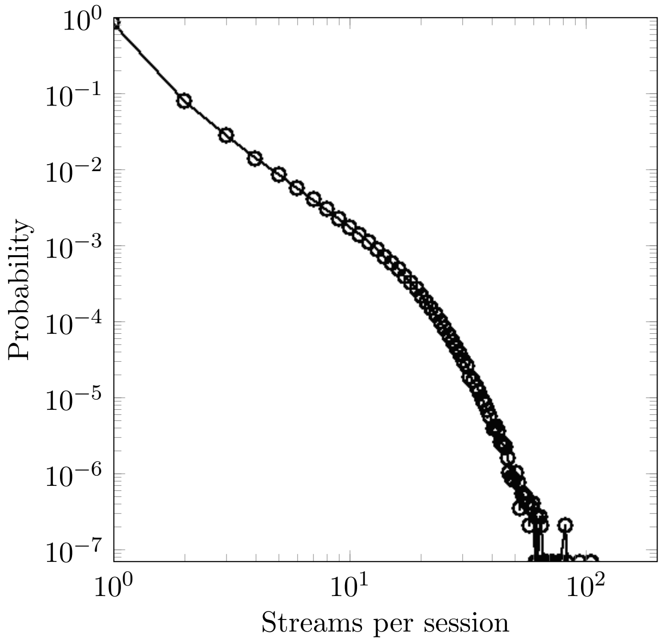
Then, they can continue on browsing other photos from this photostream. The grid view may be used for purposes which seem less involving to the user, *e.g.*, quick browsing many photos from a stream, having an overview of a stream, or seeking interesting content. Photo-focused views give the user options of performing many different actions in reference to the photo, *e.g.*, he or she can comment on the photo, favorite it, download it, see it in different sizes, or in a light-box setting.

For the purpose of the study, we define a *stream-browsing sequence* as an uninterrupted chronological sequence of page views that contains at least one photo-focused view and an indefinite number of grid views of one particular photostream (schematic examples are shown in Figure 5.6). Each browsing session can consist of a number of stream-browsing sequences.

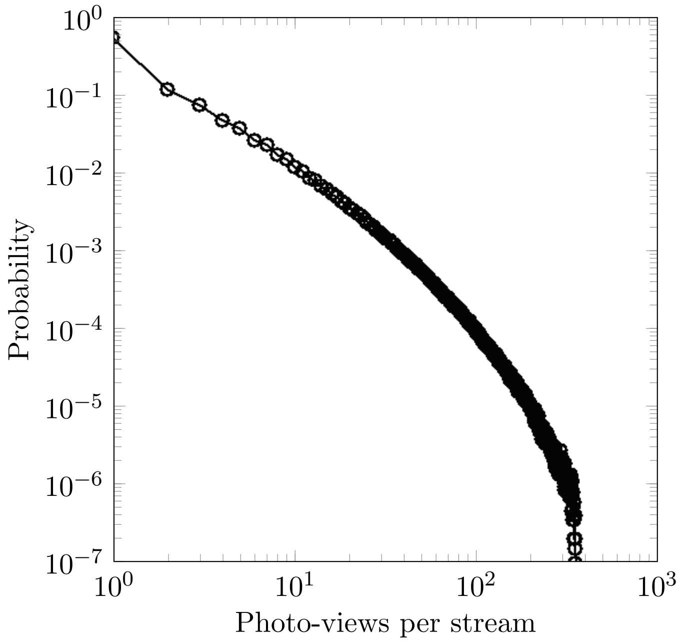
The FlickrBrowsing dataset contains a total of 264 million page views, out of which a considerable part forms stream-browsing sequences. On average, each sequence consists of 8 page views, among which there are photo-focused views and grid views of the photostream. Distributions of both the number of distinct streams viewed per session (Figure 5.4a), and the number of photo-focused views per stream (Figure 5.4b), have a heavy-tail showing high variability in user browsing patterns.

Transitions Between Streams. We have just shown that a large portion of all page views corresponds to sequential browsing of photos inside of photostreams. In this context an interesting question to ask is how users switch between streams.

We distinguish two types of transitions (Figure 5.6): *a*) *direct transitions*,



(a)



(b)

Figure 5.4: Distributions of number of unique streams per session (a) and a number of photo-focused views per each unique stream in a session (b), in log-log scale.

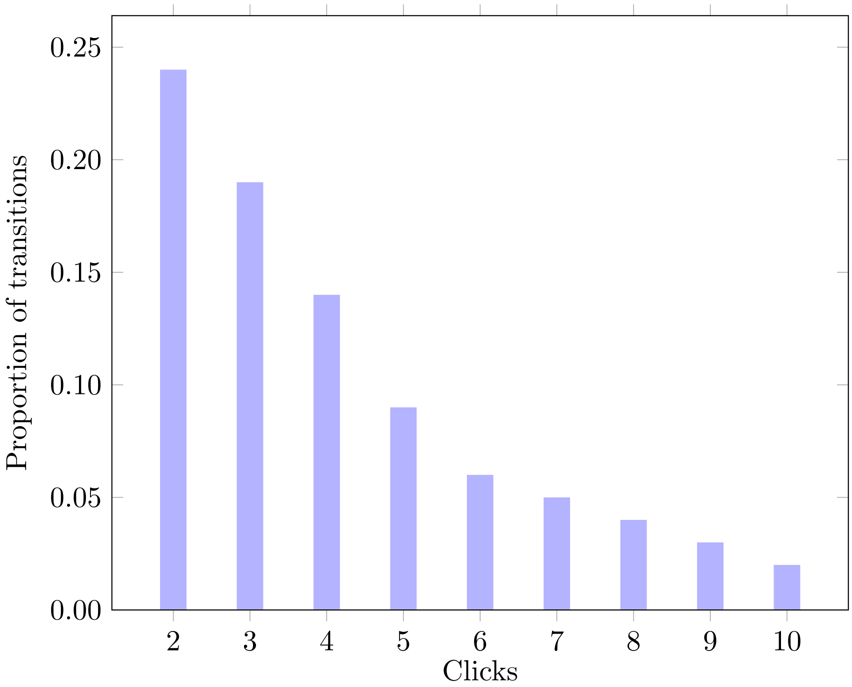


Figure 5.5: The number of clicks between different streams.

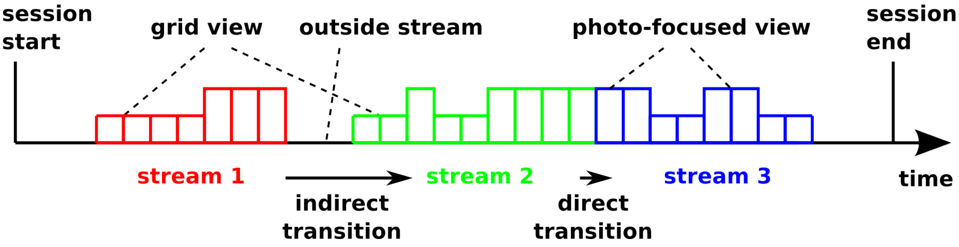


Figure 5.6: Diagram of possible transitions between streams.

which happen when the user is in a photo-focused view of the stream and chooses one of the listed streams to the right of the photo, as in Figure 5.3b,[2] and *b*) *indirect transitions*, in which the user leaves the photo-focused view and enters it again in a different stream after performing a number of clicks (*e.g.*, watching grid views, searching, exploring users' profiles, *etc.*).

We define a transition from photostream i to j as a sequence of non-photo-focused page views from a photo-focused view inside stream i to another photo-focused view inside stream j. This definition implies directionality. One can estimate the number of clicks and actions performed during the transition by counting the number of page views between the photo-focused views of the two streams and summing one. Direct transitions only require one click, whereas indirect transitions require more than one action.

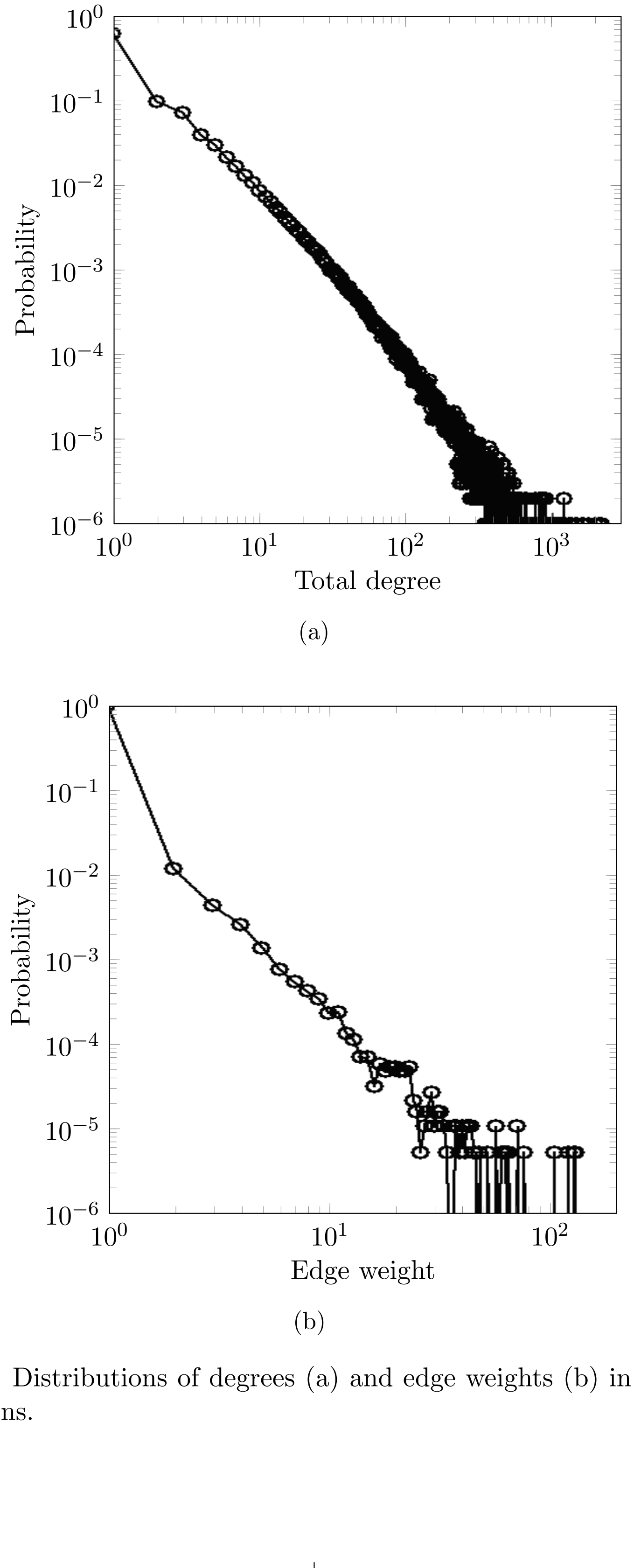
In total we have identified 3.6 million transitions between photostreams. Indirect transitions within 2 clicks cover a large portion of all transitions, as shown in Figure 5.5. However, even more transitions happen after more than 5 clicks, so many users, before reaching another picture in a photostream pass through many non photo-focused page views. Moreover, direct transitions happen much less often than indirect transitions. In the present Flickr interface, photostreams which are reachable from the currently browsed stream with just 1 click are the ones that the displayed photo belongs to. Moreover, in Flickr, only the names of these photostreams are presented, with no thumbnails of pictures shown, which may negatively impact the number of direct transitions between streams.

Discussion. Almost half of all page views in the dataset form stream-browsing sequences. Users tend to see multiple photos of a photostream either in the photo-focused view or in the grid view before leaving the stream. The vast majority of all transitions between photostreams take place over several clicks. These results suggest that a modified photo-focused interface facilitating direct transitions to other streams could be implemented.

5.2.2. Graph of Transitions Between Streams

In this section, we study the transitions between streams in more detail. Our goal is to show that users tend to browse photos of a given topic and sometimes switch to another topic that is related but that is not obvious. Such observation plays an important role in the design of interfaces and recommender systems.

[2] Sample Flickr pages from the user http://www.flickr.com/photos/bombeador/.



(a)

(b)

Figure 5.7: Distributions of degrees (a) and edge weights (b) in the graph of transitions.

Graph	Nodes	Degree	Strength
Full	1,530,875	4.23	4.68
LCC	972,047	5.80	6.46

Table 5.4: General statistics of the stream transition graph and its large connected component (LCC).

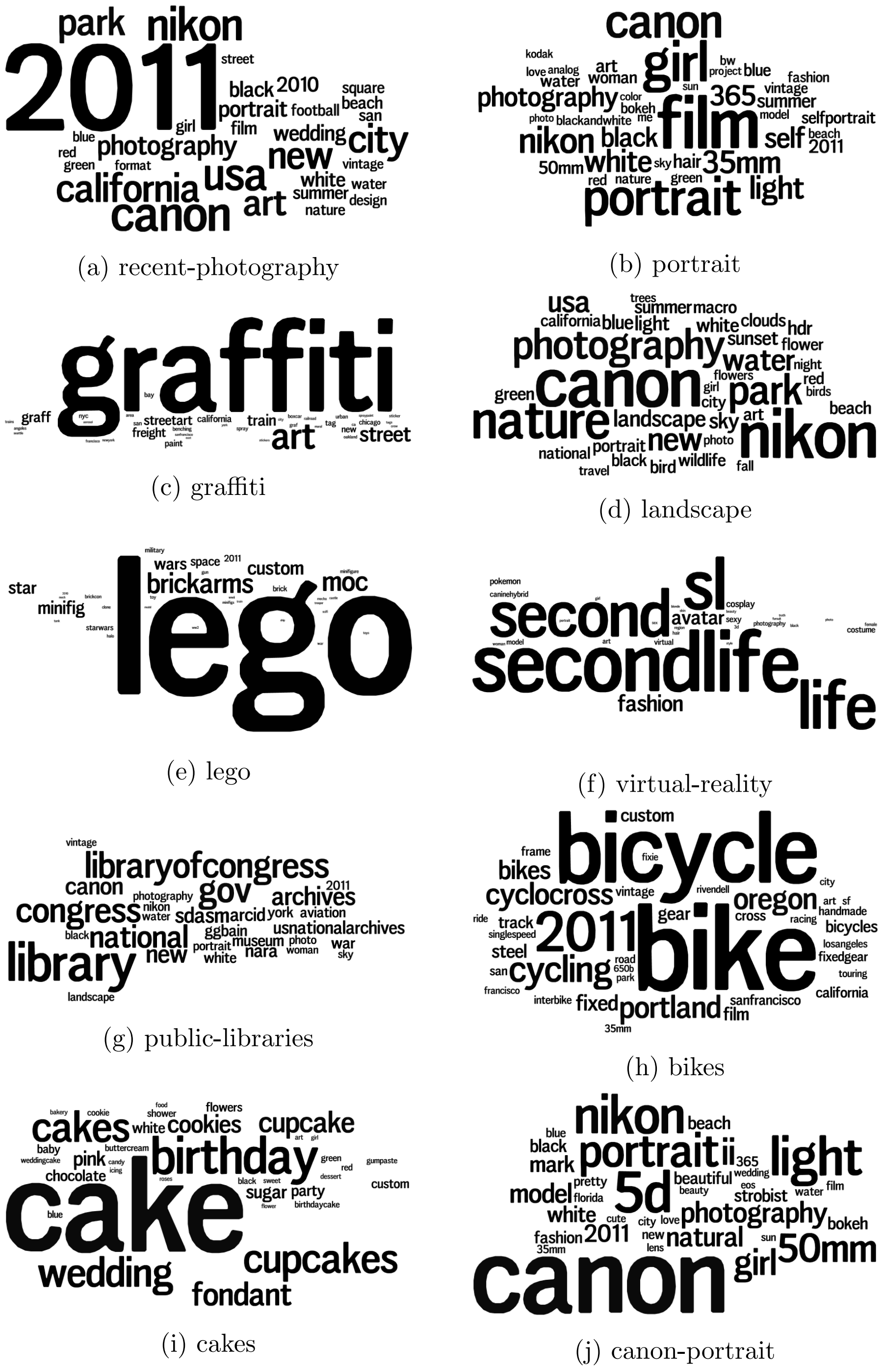
General Description. We define the graph of transitions as follows. Each photostream is treated as a node in a network. Edges in the graph represent transitions between photostreams i and j and their weight is equal to the number of such transitions.

The resulting total number of nodes in the transition graph is over 1.5 million, with an average degree of 4.2, as stated in Table 5.4. The graph is therefore sparse. The average strength of nodes, defined as the sum of the weights of its outgoing and incoming edges, is 5.8. The graph is characterized by typical heavy-tailed distributions of degrees (Figure 5.7a) and weights (Figure 5.7b). Many nodes of the graph belong to the largest connected component, which covers over 60% of all nodes in the network. Further analysis presented in this section is based on it. The largest connected component has similar characteristics to the whole network, with slightly higher average degree and average strength, as presented in Table 5.4.

Clusters of Frequently Co-viewed Streams. In order to investigate if users browse photostreams sharing similar features, we first *cluster these streams* using a community detection algorithm. A priori, detected clusters consist of nodes with dense connections, therefore they consist of photostreams where transitions are frequent. Our goal is to test if clusters of streams share common features.

We used Infomap [126, 128], a state of the art community detection algorithm for weighted and directed networks. This algorithm was found to be one of the best performing methods in a recent review [55]. Infomap detects hierarchical community structure, but for the purpose of this section we analyze only the highest hierarchical level of communities. The number of clusters found by the algorithm at the top level is over 2000.

In order to illustrate the content of clusters covering a considerable portion



(a) recent-photography

(b) portrait

(c) graffiti

(d) landscape

(e) lego

(f) virtual-reality

(g) public-libraries

(h) bikes

(i) cakes

(j) canon-portrait

Figure 5.8: Tag clouds for the large clusters of photostreams.

Cluster Label	Number of photostreams	Escape ratio	Self tag-coherence	Global tag-coherence
recent-photogr.	36,260	0.03	0.003	0.003
portrait	35,689	0.21	0.021	0.008
graffiti	20,518	0.06	0.060	0.006
landscape	12,073	0.21	0.009	0.005
lego	8,015	0.03	0.059	0.006
virtual-reality	6,001	0.07	0.030	0.004
public-libraries	5,044	0.28	0.006	0.004
bikes	3,809	0.14	0.009	0.003
cakes	3,748	0.08	0.056	0.007
canon-portrait	3,456	0.28	0.021	0.008

Table 5.5: Statistics of the large clusters of photostreams.

of the network, we show properties of some of the largest clusters. A possible way to measure the quality of the detected cluster is by calculating the ratio

$$\epsilon_i = \frac{\ell_i^{ext}}{\ell_i^{ext} + \ell_i^{int}}$$

where ℓ_i^{ext} is the number of edges connecting nodes from the cluster i with external nodes from other clusters, and ℓ_i^{int} is the number of edges connecting internal nodes from the cluster i. In this work we call it *escape ratio*, as in our context it measures likeliness of a user browsing inside of a stream from a particular cluster to escape from this cluster by switching to a stream from another cluster. Generally, this ratio should be small for well-defined clusters, however it grows with the number of clusters and their size [63]. Values of the escape ratio for the large clusters found in the largest connected component of the transition graph are shown in Table 5.5. The results vary on clusters but tend to be very low. Given that the largest cluster accounts for less than 4% of all streams, its escape ratio of 0.03 is much lower than the escape ratio of 0.96 that could be expected in a random scenario.

In order to characterize the content of the clusters we aggregate all photo tags that belong to all the streams of each of the clusters. If a photo belongs to several photostreams in one cluster, then we count its tags multiple times. Using this method, we created tag clouds for every cluster in the network. We present them in Figure 5.8, where we plot the 50 most frequent tags for each large cluster. The size of each tag in a tag cloud is proportional to the number of its appearances in the cluster. The labels, stated in the figure underneath each of the tag clouds, are chosen manually.

As one can see in the tag clouds, most of them have quite a narrow focus, and only a few have a wide focus: recent-photography, portrait, landscape, public-libraries, and canon-portrait. As a side note, the narrow focus of clusters could possibly arise from just a few streams with many tags. To test if this is not the case and to quantify narrowness of cluster topics we use a measure of similarity s_{ij} between streams i and j. We define it as the cosine similarity of multidimensional vectors of tag-clouds s_{ij}, where each dimension is a tag and the value is the count in the tag cloud. For every cluster we measure average similarity of its member streams with *a)* other member streams from this cluster, and *b)* all streams. We call these averages, correspondingly, *self tag-similarity* and *global tag-similarity.* The former property measures how coherent are streams within a particular cluster, whereas the latter quantifies how coherent these streams are with respect to all streams. As one can see in Table 5.5, the self tag-similarity is several times higher than the global tag-similarity for most of the clusters, meaning that indeed streams belonging to the same cluster are similar in content. Moreover, the clusters with narrow focus obtain the best scores as their self tag-similarity is up to 10 times higher than their global tag-similarity. Therefore streams in the clusters tend to be of similar topics.

Transitions Between Clusters. Since clusters contain streams of similar topic, an interesting question to ask is between which clusters people switch most often. This can be answered by a creating a node in place of every cluster of streams and aggregating edges of all streams belonging to this cluster. In this manner, we obtain a directed and weighted network of transitions between clusters from the network of transitions between streams. After the conversion we remove self-loops. This network is dominated, however, by the connections between large nodes. To account for the size effect of the nodes and to extract meaningful information about relations between clusters, we take the following approach. In the random case, the expected number of connections from node i to node j, having an out-degree k_i^{out}

and an in-degree k_j^{in}, is equal to

$$\ell_{ij}^{rand} = \frac{k_i^{out} k_j^{in}}{\ell}$$

for a large number of edges in the network ℓ. If connections between clusters were spread randomly between nodes of known degrees then ℓ_{ij}^{rand} would be expected to be the number of edges between particular nodes. To see which connections between clusters are the furthest from a random configuration, we calculate the ratio

$$a_{ij} = \frac{\ell_{ij}}{\ell_{ij}^{rand}}$$

of the actual number of connections ℓ_{ij} and the expected value ℓ_{ij}^{rand}. We call a_{ij} the abundance ratio. If this ratio is larger than 1 then transitions from stream i to j are overrepresented, while if it is lower than 1 then they are underrepresented. We pick the parts of the network formed by edges with abundance ratio a_{ij} higher than 10 and actual weight ℓ_{ij} also higher than 10. We present most of them in Figure 5.9. Here we provide a short description of each of the examples:

(a) Clusters of fans of cars and machinery. From left to right in the figure: the first cluster seems to be on the boundary of cars and photography, while the next one is more narrowly about cars, especially classic ones. Users from this cluster tend to switch between both to see photos of trains and railroads, as well as firetrucks.

(b) Event-oriented clusters. From down to up in the figure: photography of live music shows is related to the cluster of journaling, blogging and fisheye photography.

(c) Household-centered clusters. From left to right: clusters of cakes and vintage style, which incorporate elements from previous eras into modern fashion and style, are related to the cluster of sewing and fabrics, and this is related to dolls. Note that dolls and Disney/Disneyland are also related.

(d) Toys and military. From left to right: photography of lego constructions, mostly of star-wars, is related to army and military photography. This is quite interesting, and shows an interests from toys and plastic soldiers to real ones. The military cluster is related to natural disasters in which often the army and powerful natural forces are involved.

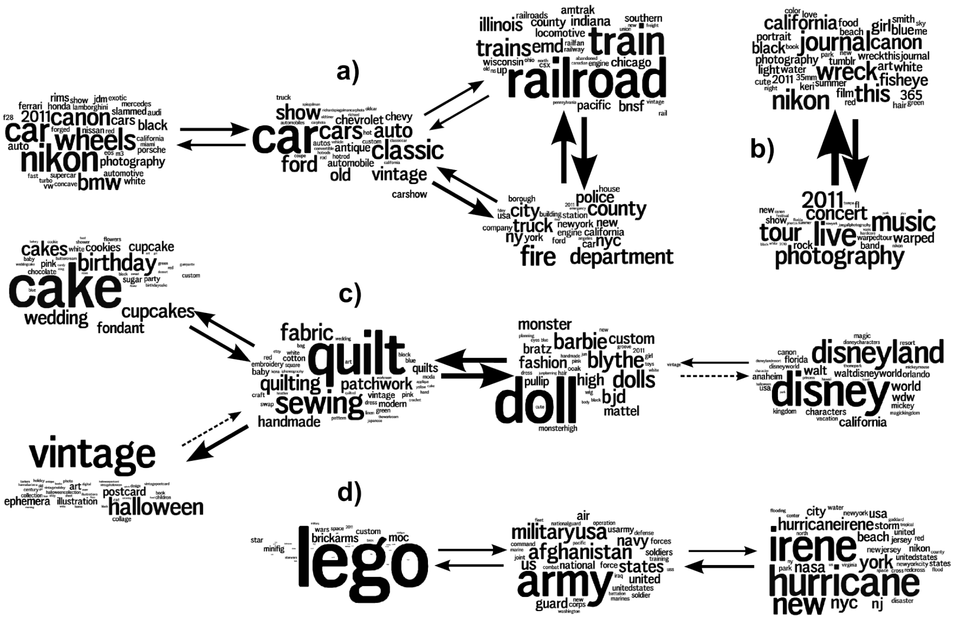

a)
car wheels nikon canon 2011 ferrari honda rims show jdm exotic mercedes slammed audi cars black nissan forged red auto california miami porsche photography bmw supercar automotive white fast turbo vw concave f28
car cars show chevrolet chevy auto classic ford old vintage automobile antique hotrod truck carshow
railroad train trains emd illinois county indiana amtrak railroads southern locomotive wisconsin csx chicago pacific bnsf railfan railway rail
fire department county police truck city newyork new ny york nyc car engine california station building borough usa company house
b)
journal canon california love food girl smith portrait black beach blue photography park tumblr wreck wreckthisjournal art white light water 2011 cute 35mm keri summer fisheye nikon film this 365 red hair green
2011 concert music tour live photography show rock band warped warpedtour new
c)
cake cakes cookies cupcake birthday cupcakes wedding fondant chocolate pink sugar party custom baby flowers
quilt fabric quilting sewing patchwork quilts handmade cotton embroidery baby vintage pink modern green craft swap red white art black blue
doll monster barbie custom bratz fashion blythe dolls pullip high bjd mattel monsterhigh
disneyland disney walt waltdisneyworld orlando anaheim florida world wdw mickey kingdom characters california vacation canon usa
vintage ephemera illustration art postcard halloween collage
d)
lego wars space brickarms custom moc star minifig
military usa air afghanistan army us national navy forces states soldiers united guard corps marines
irene hurricane hurricaneirene storm city water newyork usa beach jersey nikon york nasa park new nyc nj disaster

Figure 5.9: Interesting over-represented transitions between clusters of photostreams. Width of edges corresponds to the abundance ratios.

It is also possible to find underrepresented connections between clusters, and it would certainly be interesting to examine more clusters in detail.

Discussion. On one hand, low escape ratios and high tag-coherence of the clusters of streams show that indeed users browse topically-similar streams. On the other hand, examples of the transitions between the clusters show that the users also switch between streams which are further apart in the topical space, but are still related (*e.g.*, trains and firetrucks, cake and sewing, lego and army). This implies interesting consequences for the design of new interfaces or recommender systems, *e.g.*, the recommended photos should not be topically overspecialized, in order to leave to users the possibility of exploration.

5.3. Visualizing the Life of a Browsing Session

After having presented the analysis of the life of a session, we will show an example of application that leverages data about user browsing sessions to generate interactive visualizations. We present an alternative way of presenting content, by leveraging the aggregated and anonymous navigation patterns of thousands of visitors to a site. The key idea behind our work is that creating dynamic interfaces that leverage previous visitor's patterns can lead to more serendipitous content exploration experiences. In particular, we focus on a graph-based paradigm where nodes represent content items and edges represent the number of visits (page views) that originate in one content item (node) and move to the next one. This paradigm allows users to discover content based on the aggregated browsing patterns of others. It is important to emphasize that our approach fully preserves privacy since navigation patterns of individual users are never visible because all data is anonymized and aggregated: links between two content items exist only if a sufficiently high number of page views has occurred between the content items. Only public photos are included in our analysis and application.

User browsing patters have been used in the past to inform design choices. For example, Paul André *et al.* [5] analyzed the characteristics of user behavior in image search and suggested ways to improve image search engines. Our approach relates to social navigation [57], and to the idea of "Footprints" [156]. Related image-browsing interfaces are described in [132, 131]. The majority of image-browsing interfaces focus on exploiting the similarity between images (*e.g.*, [131]), and while the Footprints framework uses the interaction history for navigation in a complex information space, those tools assume that people know what they want, but may need help finding their way to the information and may need help understanding what they have found. Our purpose, in contrast, is to increase serendipitous discovery.

Websites that host content, such as Flickr, have structured, fixed layouts and sections, which give them consistency and facilitate their use. In many cases, however, the possible navigation paths taken by visitors to such sites can be very rich: in Flickr's case, visitors can view photo streams, profiles of users, their favorite photos, photos of contacts, photos in groups, via search, and other mechanisms. In spite of this wealth of options for viewing content, users are somewhat constrained by the sites' layouts and how the content is organized.

5.3.1. Constructing the Browsing Graph

For the visualization we used the `FlickrBrowsing` dataset (see Section 3.3.2). We concentrated only on the URLs that refer to the *Browse user photos* page layout, *i.e.*, pages aimed at displaying a particular photo. Moreover, since we are interested in representing navigation among photos, we remove the page views in which the referrer and current URLs refer to the same photo (*e.g.*, page reloads, change in photo resolution, *etc.*).

From the filtered list of page views, we create a weighted graph. We create a node i for each photo page and an edge e_{ij} when there are at least n_p page views from i to j. The weight of the arc is the number of page views from i to j.

5.3.2. The Interface

As already seen in Section 5.2, users often view photos in long sequences (*e.g.*, all photos of an album). Therefore the browse graph often consists of long *chains* of nodes. We designed two layout algorithms to compactly display the photos: *a*) *Spiral layout* (Figure 5.10a): photos are displayed on a grid: the first photo of a chain is placed at the center and subsequent photos are placed around it in clockwise order; *b*) *Force-directed layout* (Figure 5.10b): the first photo of a chain is placed at the center and as photos are rendered, they are added radially outwards from the center, but each one is placed using a clockwise angular offset. A force-directed layout algorithm [88] is then applied to the photos.

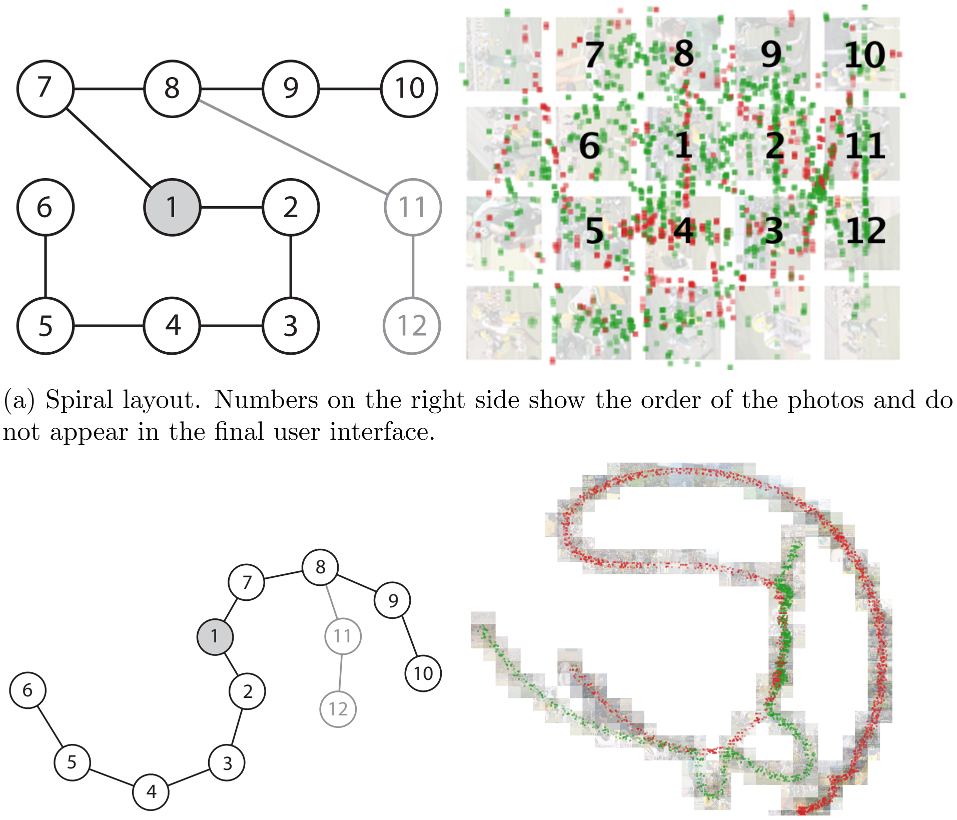
For example, consider a graph containing 12 nodes $\{k\}_{k=1}^{12}$. Suppose past users navigated from node 6 to 5 and down to 1 and then follow to nodes 7 and 8. At this point they either go to nodes 9 and 10 or nodes 11 and 12. The left side of Figure 5.10 shows how each layout would render the resulting subgraph, where photo 1 is in the center.

The front end of our interface is written in HTML/Javascript, the backend is written in PHP and the data is stored in a MySQL database. Initially the interface displays a subgraph containing a predefined number of photos. The user is able to interact with the interface in the following ways:

- *Change layout algorithm*: the user can switch between the two layouts using the same set of photos.

- *Show additional photos*: the user can expand the fringe nodes of the displayed subgraph by clicking on a button. Moreover, she can show the neighbor nodes of a photo by clicking on it.

- *Display traffic*: the user is able to dynamically visualize how past users navigated in the browse graph. This is represented as animated



(a) Spiral layout. Numbers on the right side show the order of the photos and do not appear in the final user interface.

(b) Force-directed layout.

Figure 5.10: The figures show the two layout algorithms (spiral and force-directed). The left part displays an illustration of the layout of a sample graph of 12 photos centered in photo 1 (in grey). On the right side, we see an example using real data. Edges on the left side are represented as animated particles on the right side. Their color indicates the context in which users moved across photos: red indicates that users moved inside the same Flickr group, green that users moved inside the same album. The different paths assumed by the red and green particles show that users tend to navigate photos differently depending on the context.

particles that move from photo i to photo j. Their number is proportional to the number of users (*i.e.*, if many users browsed photo i followed by photo j, a large number of particles will move from i to j).

- *Filter traffic by photo*: by hovering with the mouse on photo i, the particles are filtered and only the ones passing through i are displayed. In the example of Figure 5.10, suppose that people which saw photo 9 browsed to photos 8 or 10, but that no user saw photos 9 and 11 in the same session. Therefore, if the user hovers the mouse on node 9, the interface will display particles going from photo 9 to 8 and 10, but not to 11. This is because users who saw 9 did not navigate to 11. Note that this happens although there is an edge between photos 8 and 11.

- *Show/Filter traffic by type*: the user is able to display the particles according to the context in which past users saw the photos (photos were viewed in the same album, in the profile of a group or are both favorited by the same user, *etc.*). Each type of traffic is displayed in a different color. The right part of Figure 5.10 shows an example.

5.4. Discussion

In this chapter, we analyzed a large sample of sessions from an image sharing platform.

First, we studied user search behavior. Our study uses logs of user behavior during entire search sessions, as opposed to only the search and result click data that are available on standard search logs. Using a taxonomy of image search we describe the main categories of search performed on this platform. We note differences with previous results on general image search and image search in journalism.

We then worked with photostreams as content units for analyzing user browsing behavior in Flickr. In particular, we presented the results of an analysis of a large sample of Flickr navigation logs to gain insights into how users navigate between photostreams. To analyze frequent stream topic transitions, we created a stream transition graph from over 100 million page views. We found interesting patterns in how users navigate between streams and showed that users tend to browse related streams.

Finally, we developed a visualization that suggests that leveraging large-scale aggregated navigation logs can lead to the creation of new and interesting content exploration paradigms. In particular, we conjecture that a dynamic graph-based representation can be effective in serendipitous content discovery.

In the next chapters, we use more complex methods to understand browsing sessions. First of all, we describe a data mining algorithm to extract frequent browsing patterns (which we call summaries) that show a particular behavior of the user. We then move towards probabilistic models of browsing sessions.

Chapter 6

Summarization Based on User Browsing Sessions

6.1. Introduction

We study the novel data-mining problem of extracting summaries from a database recording activity sequences on a graph. As better explained next, ours is a general framework that can be instantiated in different contexts, ranging from information propagation in social networks to user browsing activity over the Web.

In our problem, we are given a database $\mathbb{P}$ of *propagations*, where each propagation represents the trace left by a specific entity ϕ that "flows" over an underlying graph $G = (V, E)$. More precisely, a trace of an entity ϕ is a sequence of observations $\langle v, \phi, t \rangle$ representing the fact that the entity ϕ is observed at node v at time t. The trace of ϕ can naturally be represented as a *directed acyclic graph* (DAG) D_ϕ, whose arcs (u, v) express the fact that an arc between the same nodes exists in the underlying graph G and both u and v activate on ϕ, with u activating strictly before v. In this case, we can think that ϕ flowed from u to v. An example of our input is provided later, in Section 6.2 (and Figure 6.2).

Our goal is to *extract summaries of propagation traces* from $\mathbb{P}$. A summary is a set of propagations. The propagations in a summary should satisfy the requirement of being *structurally similar*. More precisely, we want them to involve (more or less) the same population of nodes. This is however not enough: we also require the propagations to exhibit a well-defined *hierar-*

chical structure when merged together. By hierarchical structure, we mean that one can identify one or more rankings, such that there exist nodes that usually participate early in the propagations inside the summary, and nodes that instead activate later. We mill make these concepts more formal later in the chapter.

Given a summary S, we define the *union graph* $G(S)$ as the directed graph (subgraph of G) obtained by merging the DAGs corresponding to the propagations in S. Moreover we denote as r a ranking of the nodes in $G(S)$. The ranking expresses the hierarchical structure of S, *i.e.*, the order of nodes that entities follow while propagating in $G(S)$. The ranking of nodes makes our summaries informative and useful in a wide and diverse range of applications, like the ones discussed next.

Application Scenario #1. In a social network like Twitter the underlying graph G represents the social connections: the nodes of the graph are the users of the social network and an arc (u, v) exists if u and v are related in some way (*e.g.*, v is a follower of u). Here the entity ϕ is a piece of information (*e.g.*, the URL of an interesting blog, multimedia content, such as photos/videos, *etc.*) propagating in the social networks by means of *re-tweets*. Given a database of propagations $\mathbb{P}$, finding summaries is equivalent to finding groups of entities that propagate in the social network in a similar way. Nodes of these groups may represent different communities of users interested in different topics: politics-related entities flow trough a different set of nodes than entities related to electronic music. Moreover, a summary also comes with a ranking of nodes that reflects, to some extent, the directionality of the information flow in all the propagations in the summary. This is crucial to distinguish users that are "early adopters" (or "opinion leaders", or "trend setters", or "influencers") from users that are instead simple followers.

Application Scenario #2. In another context, the graph G could be a (large) website (*e.g.*, Amazon or Wikipedia), whose nodes and arcs correspond to web pages and hyperlinks between pages, respectively. In this case, the entity ϕ moving in the directed graph corresponds to (a browsing session of) a specific user visiting the website, and multiple users clearly leave different traces. The browsing behavior of a user within the website can be represented as a DAG, where the user moves from the homepage down in the hierarchical structure of the pages of the website. Cycles might arise because of user backtracking or because they are allowed by the website

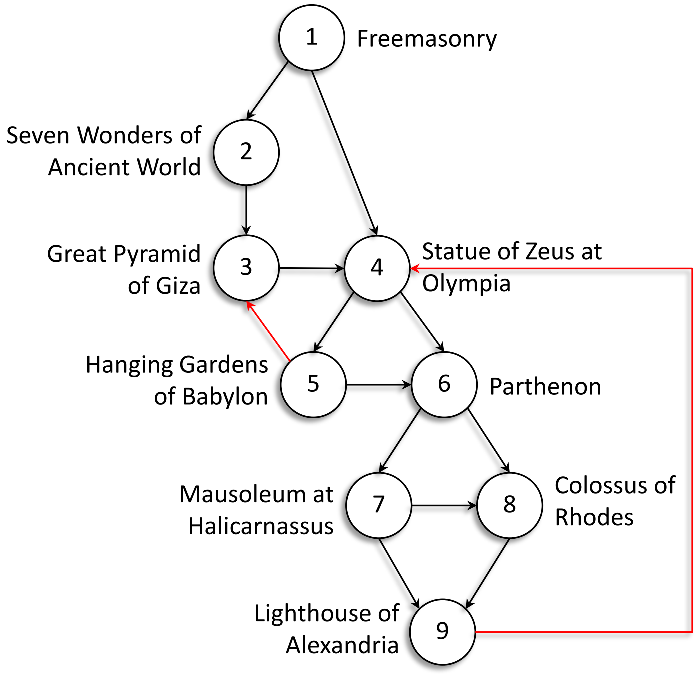

1
Freemasonry
2
Seven Wonders of Ancient World
3
Great Pyramid of Giza
4
Statue of Zeus at Olympia
5
Hanging Gardens of Babylon
6
Parthenon
7
Mausoleum at Halicarnassus
8
Colossus of Rhodes
9
Lighthouse of Alexandria

Figure 6.1: An example of union graph of a summary extracted from a database of browsing sessions in Wikipedia (application scenario #2). The numbers inside the nodes identify one optimal ranking. Links that violate the ranking are depicted in red.

structure. However, as what really matters is the sequence of the pages visited, one can safely ignore cycles and simply consider only the first visit to each page.

Finding summaries, in this case, means finding groups of users (sessions) that navigate the website in a similar fashion. It is essential for website-usage analysis and re-organization [139], as a summary indicates how a specific group of users accesses the website, which are the pages accessed first, and which access page is a good entry point to discover many other pages. The typical browsing activity of a group of users is thus described by its corresponding summary, and different groups of users can be detected and discerned based on their summaries. Moreover, the typical behavior of successful sessions (*e.g.*, the ones ending in a product purchase) can easily be detected and studied in order to improve the website organization.

An example of union graph of a summary extracted from a database of browsing sessions in Wikipedia is provided in Figure 6.1 (more details are given in Section 6.4). The numbers inside the nodes correspond to one optimal ranking, while red links denote ranking violations, *i.e.*, arcs (u, v) for which $r(u) > r(v)$. In the next section we will formalize the concept of violating the ranking, and based on that, we will define the constraint to be satisfied in order to guarantee good hierarchical structures.

Our contributions are as follows:

- We formulate the novel problem of extracting structurally-similar summaries from a set of propagations. We define the requirement about structural similarity as two constraints that the extracted summaries are required to satisfy. One constraint aims to force all the propagations in a summary to involve more or less the same set of nodes, while the second constraint requires for the nodes in the union graph of a summary to have a well-defined hierarchy, *i.e.*, it should be as as close as possible to a DAG structure.
- We show that both constraints satisfy the downward-closure property, thus allowing the definition of Apriori-like methods.
- We devise two algorithms to solve our problem, which differ from each other in the way in which they visit the lattice of all possible sets of propagations. The first algorithm, called BOTTOM-UP, relies on a bottom-up lattice-traversing strategy, which directly exploits the aforementioned closure properties. Motivated by the fact that checking the hierarchy constraint is more time-consuming, we develop a second algorithm, called UP-AND-DOWN, which consists of two separate phases: a bottom-up phase where the hierarchy constraint is discarded, followed by a top-down phase that partially re-visits the lattice starting from those summaries that violate the hierarchy constraint.
- We extensively evaluate our algorithms on four real-world datasets coming from the application scenarios discussed above, *i.e.*, information propagation on social networks and web browsing. Quantitative results show that the UP-AND-DOWN algorithm generally achieves better efficiency, even though BOTTOM-UP can be faster in some settings. Qualitative results provide evidence of the significance of the summaries extracted and how they can be exploited for practical purposes.

The rest of the chapter is organized as follows. In Section 6.2 we define our problem, while in Section 6.3 we describe the proposed algorithms to solve it. Section 6.4 presents the experimental analysis. Section 6.5 concludes the chapter.

The results of this chapter were published in [103].

6.2. Problem Definition

The input to our problem is (*i*) a directed graph $G = (V, A)$ representing a network of interconnected objects, (*ii*) a set $\mathcal{E}$ of *entities*, and (*iii*) a set $\mathbb{O}$ of *observations* involving the objects of the network and the entities in $\mathcal{E}$. As mentioned earlier in the chapter, objects can be, *e.g.*, users in a social network or pages of a website, while entities can be, for example, pieces of information (such as multimedia content) shared by users or web-page visits. Each observation in $\mathbb{O}$ is a triple $\langle v, \phi, t\rangle$, where $v \in V$, $\phi \in \mathcal{E}$, and $t \in \mathbb{N}^+$, denoting that the entity ϕ is observed at node v at time t. We assume that the same entity cannot be observed multiple times at the same node; should this happen, we consider only the first one (in order of time) of such observations.

The set $\mathbb{O}$ of observations can alternatively be viewed as a database $\mathbb{P}$ of *propagation traces* (or simply *propagations*), *i.e.*, traces left by entities that "flow" over G. Formally, a propagation trace of an entity ϕ corresponds to the subset of all observations in $\mathbb{O}$ involving that entity, *i.e.*, $\{\langle v, \phi', t\rangle \in \mathbb{O} \mid \phi' = \phi\}$. Considering the graph G, the database of propagation traces corresponds to a set of *directed acyclic graphs* (DAGs) $\mathbb{D} = \{D_\phi \mid \phi \in \mathcal{E}\}$, where, for each $\phi \in \mathcal{E}$, $D_\phi = (V_\phi, A_\phi)$, $V_\phi = \{v \in V \mid \langle v, \phi, t\rangle \in \mathbb{O}\}$, $A_\phi = \{(u, v) \in A \mid \langle u, \phi, t_u\rangle \in \mathbb{O}, \langle v, \phi, t_v\rangle \in \mathbb{O}, t_u < t_v\}$. Note that each $D_\phi \in \mathbb{D}$ is guaranteed to contain no cycles due to time irreversibility. Moreover, we assume that each propagation is started at time 0 by a dummy node $\Omega \notin V$, representing a source of information external to the network that is implicitly connected to all nodes in V. Thus, each DAG in $\mathbb{D}$ is assumed to contain such a dummy node Ω connected to all its "real" nodes. An example of our input is provided in Figure 6.2.

A *summary* $S \subseteq \mathbb{P}$ is a set of propagations. Given a summary S, we denote by $D(S)$ the set of DAGs corresponding to the propagations in S and by $G(S)$ the union graph of all the DAGs in $D(S)$. The union of two graphs $G_1 = (V_1, A_1)$ and $G_2 = (V_2, A_2)$ is defined as $G_1 \cup G_2 = (V_1 \cup V_2, A_1 \cup A_2)$. An example of graph resulting by the union of two DAGs is given in Figure 6.3.

$\mathbb{P}$

v	ϕ	t
Ω	ϕ_1	0
v_2	ϕ_1	2
v_3	ϕ_1	4
v_4	ϕ_1	5
v_5	ϕ_1	7
Ω	ϕ_2	0
v_2	ϕ_2	1
v_1	ϕ_2	3
v_5	ϕ_2	6
v_7	ϕ_2	7
v_6	ϕ_2	8
v_3	ϕ_2	9
Ω	ϕ_3	0
v_1	ϕ_3	1
v_2	ϕ_3	3
v_6	ϕ_3	5
v_7	ϕ_3	7
v_4	ϕ_3	8

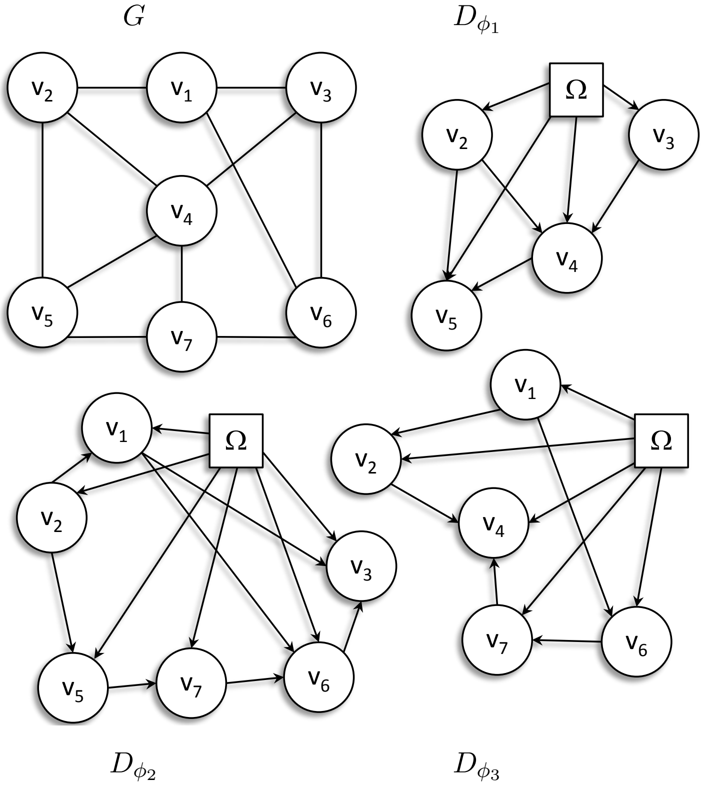


Figure 6.2: An example of the input of our problem: a graph G, and a database of propagation traces $\mathbb{P}$ defined over a set of entities $\mathcal{E} = \{\phi_1, \phi_2, \phi_3\}$. The graph represented here is undirected: each edge corresponds to two directed arcs. Each propagation is started at time 0 by a dummy node $\Omega \notin V$. Given the graph G, the propagation database $\mathbb{P}$ is equivalent to the set of DAGs $\mathbb{D} = \{D_{\phi_1}, D_{\phi_2}, D_{\phi_3}\}$.

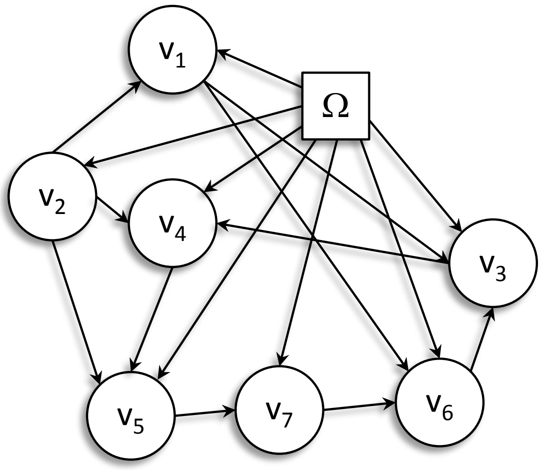


Figure 6.3: The union graph $D_{\phi_1} \cup D_{\phi_2}$ of the DAGs D_{ϕ_1} and D_{ϕ_2} depicted in Figure 6.2.

Note that $G(S)$ is not necessarily a DAG itself, as the union of multiple DAGs can clearly correspond to a cyclic graph.

As informally anticipated earlier, our goal is to extract summaries that (*i*) are homogeneous in terms of the population of nodes involved, and (*ii*) exhibit a good hierarchical structure, *i.e.*, the union graph is as close as possible to a DAG structure. We next formalize these two concepts in two constraints that we require for our summaries to satisfy.

Similar Population of Nodes. To force our summaries to involve a similar population of nodes, a natural choice is to quantify the amount of nodes common to all DAGs of the propagations in a summary and require that it is no less than a certain threshold. In this work, we measure the fraction of nodes shared by a set of DAGs by means of the popular *Jaccard similarity coefficient*, which is one of the most used measures of similarity among sets of objects. Moreover, it has the desirable property of having a fixed-range codomain, $[0, 1]$, which makes the threshold-setting task easier. Formally, given a summary S, we define

$$j(S) = \frac{|\bigcap_{D_\phi \in D(S)} V_\phi|}{|\bigcup_{D_\phi \in D(S)} V_\phi|}.$$

Hierarchical Structure. Here we borrow the concept of "agony" introduced by Gupte *et al.* [66] to define a measure of the hierarchy existing in a directed graph. Given a directed graph $G = (V, A)$, consider a ranking function $r : V \to \mathbb{N}$ for the nodes in G, such that $r(u) < r(v)$ expresses the fact that u is "higher" in the hierarchy than v, *i.e.*, the smaller $r(u)$ is, the more u is an "early-adopter". If $r(u) < r(v)$, then the arc $u \to v$ is expected and does not cause any agony. Instead, if $r(u) \geq r(v)$ the arc $u \to v$ would cause agony because it would mean that u has a follower v (in the social graph terminology) that is higher-ranked than u itself. Therefore, given a graph G and a ranking r, the agony of each arc (u, v) is defined as $\max\{0, r(u) - r(v) + 1\}$, and the agony $a(G, r)$ of the whole graph given the ranking r is just the sum over all arcs:

$$a(G,r) = \sum_{(u,v)\in A} \max\{0, r(u) - r(v) + 1\}.$$

In most cases (as in our problem), the ranking r is not explicitly provided. The objective therefore becomes finding a ranking (they might be multiple) that minimizes the total agony of the graph. In this way, one can compute the agony of any graph G as

$$a(G) = \min_{r} a(G, r).$$

As a DAG implicitly induces a partial order over its nodes, it has always zero agony: the nodes of a DAG form a perfect hierarchy. For instance, in the DAGs $D_{\phi_1}, D_{\phi_2}, D_{\phi_3}$ in Figure 6.2, it is sufficient to take the temporal ordering as ranking, *i.e.*, $r(u) = t_u$ where $\langle u, \phi_i, t_u\rangle \in D_{\phi_i}$, in order to obtain agony equal to zero.

However, as already mentioned above, merging several DAGs to form a summary S leads to a union graph $G(S)$ that is not necessarily a DAG, therefore agony can appear. Consider for instance the union graph $D_{\phi_1} \cup D_{\phi_2}$ reported in Figure 6.3. It is easy to see that the graph $D_{\phi_1} \cup D_{\phi_2}$ is not a DAG as it contains the cycle $v_3 \to v_4 \to v_5 \to v_7 \to v_6 \to v_3$. Due to this cycle, it is impossible to find a ranking r that provides zero agony. In fact, any directed cycle containing k arcs (and not sharing arcs with any other cycle) always incurs agony equal to k [66]. One ranking r providing the minimum agony for $D_{\phi_1} \cup D_{\phi_2}$ is:

$$(\Omega : 0)(v_2 : 1)(v_1 : 2)(v_4 : 2)(v_5 : 3)(v_7 : 4)(v_6 : 5)(v_3 : 6)$$

This ranking yields no agony on all the arcs, except on the arc $v_3 \to v_4$ that incurs agony 6-2+1 = 5, which is indeed the length of that directed cycle.

Although the number of possible rankings of a directed graph is exponential, Gupte *et al.* [66] provide a polynomial-time algorithm for finding a ranking of minimum agony. They provide a linear-programming formulation and show that (*i*) the dual problem has an optimal integral solution, and (*ii*) the optimal value obtained by maximizing the dual problem coincides with the minimum value of the primal. This finding allows to define an algorithm that decomposes the input graph G into a DAG D and a graph H that corresponds to the maximum (in terms of number of arcs) Eulerian subgraph of G (an Eulerian graph is a graph in which the indegree of each node is equal to its outdegree). Let m be the number of arcs and n the number of nodes of G, then the algorithm to compute such a decomposition takes $\mathcal{O}(m^2 n)$ time: it requires at each iteration to find a negative-weight cycle, which can be done by the Bellman-Ford algorithm [13, 54] in $\mathcal{O}(mn)$, while, in the worst case, the number of iterations is m.

Mining Maximal Summaries under Constraints. We have now all the ingredients to define the problem we study in this chapter. Informally, given a database of propagations $\mathbb{P}$, we want to extract sets $S \subseteq \mathbb{P}$ of propagations that have high Jaccard of nodes of the DAGs in $D(S)$ (no less than a threshold β), and small agony of the graph $G(S)$ obtained by merging the DAGs in $D(S)$ (no more than a threshold α). Moreover, we want S to be maximal and have non-trivial size (*i.e.*, $|S| \geq 2$). The formal statement of the problem is stated next.

Problem 6.2.1 (Mining summaries of propagations)**.** Given a set of propagations $\mathbb{P}$, and two thresholds, $\alpha \in \mathbb{N}$ and $\beta \in [0, 1]$, we want to extract

$$\mathbb{S} = \{S \subseteq \mathbb{P} \ \mid |S| \geq 2, a(G(S)) \leq \alpha, j(D(S)) \geq \beta, \nexists T \in \mathbb{S} : S \subset T\}.$$

For each summary $S \in \mathbb{S}$ we can compute one ranking providing minimal agony for $G(S)$, that is $r^* = \operatorname{argmin}_r a(G(S), r)$.

6.3. Algorithms

The search space of Problem 6.2.1 corresponds to the whole lattice $2^{\mathbb{P}}$ of all subsets of $\mathbb{P}$. The two constraints that are part of our problem definition hold the downward-closure property, which enables effective pruning of such large search space, as explained next.

The Jaccard value is monotonically non-increasing as the number of sets to be compared increases. More precisely, given any two sets (of sets) $\mathcal{S}$ and $\mathcal{T}$, with $\mathcal{T} \supset \mathcal{S}$, it holds that $j(\mathcal{S}) \geq j(\mathcal{T})$. For our purposes, this result can easily be translated into the following pruning rule to be exploited during any (bottom-up) traversal of the lattice $2^{\mathbb{P}}$.

Fact 6.3.1. *Given a set of* DAGs $\mathbb{D}$*, a subset* $D \subseteq \mathbb{D}$*, and a threshold* $\beta \in [0, 1]$*, if* $j(D) < \beta$*, then* $j(D') < \beta$ *for all* $D' \in 2^{\mathbb{D}} : D' \supset D$*.*

The second property we show is that the agony is monotonically non-decreasing when the size of the graph increases. We formally state such a result in the following theorem.

Theorem 6.3.1 (Agony is monotone)**.** *Given a directed graph* $G = (V, A)$ *and an arc* $(u, v) \notin A$*, let* $G_{uv} = (V \cup \{u, v\}, A \cup \{(u, v)\})$ *denote the graph derived from adding the arc* (u, v) *to* G*. It holds that* $a(G) \leq a(G_{uv})$*.*

Proof. As shown in [66], the problem of minimizing the agony of a graph G can be formulated as a linear program whose dual problem corresponds to finding an Eulerian subgraph of G having the maximum number of arcs. Let $E(G)$ denote the maximum Eulerian subgraph of G and let $|E(G)|$ denote the number of arcs in $E(G)$. Another result reported in [66] is that $|E(G)| = a(G)$. Now, it is easy to see that, although not necessarily optimal, the subgraph $E(G)$ represents at least an admissible solution (thus a lower bound) of the maximum-Eulerian-subgraph problem when the graph in input is G_{uv}. Thus, combining all such results:

$$a(G) = |E(G)| \leq |E(G_{uv})| = a(G_{uv}),$$

which proves the theorem. □

An immediate corollary of the above theorem to exploit in our context is the following: if for any subset of propagations $S \subseteq \mathbb{P}$ it holds that the agony $a(G(S))$ of the corresponding union graph exceeds a certain threshold α, then the agony $a(G(T))$ of (the union graph of) every superset $T \supset S$ is guaranteed to be greater than α as well. An opposite result clearly holds for the subsets of a summary S whose agony is no less than α.

Corollary 6.3.2. Given a set of propagations $\mathbb{P}$, a subset $S \subseteq \mathbb{P}$, and a threshold $\alpha \in \mathbb{N}$, it holds that:

1. if $a(G(S)) > \alpha$, then $a(G(T)) > \alpha$ for all $T \in 2^{\mathbb{P}} : T \supset S$;
2. if $a(G(S)) \leq \alpha$, then $a(G(T)) \leq \alpha$ for all $T \in 2^{\mathbb{P}} : T \subset S$.

Based on these properties we devise two algorithms, namely Bottom-up and Up-and-down, which explore the lattice $2^{\mathbb{P}}$ of all subsets of $\mathbb{P}$ in two different ways: the Bottom-up algorithm performs a bottom-up, breadth-first search, while the Up-and-down algorithm consists of two phases: a first phase roughly similar to the Bottom-up algorithm where only the Jaccard constraint is considered, followed by a top-down phase where the lattice is partially re-visited to check the agony constraint.

In the remainder of the section, we provide the details of the two algorithms, while also discussing the advantages and disadvantages of both, especially in terms of time/space requirements.

6.3.1. The Bottom-up Algorithm

We describe here the first algorithm we devise to solve Problem 6.2.1. The pseudocode of the proposed algorithm, called Bottom-up, is summarized in Algorithm 1.

Bottom-up resembles the classic Apriori algorithm for frequent-itemset mining [1]. Given a set of DAGs $\mathbb{D}$ and two thresholds α, β, the proposed algorithm visits the lattice of all possible subsets of $\mathbb{D}$ in a bottom-up fashion. Particularly, the algorithm performs a breadth-first search, starting from the subsets of $\mathbb{D}$ of size 2 (level 2), and increasing the level one-by-one until no summaries satisfying the requirements can be extracted. This way, the pruning rules stated in Fact 6.3.1 and (the first statement of) Corollary 6.3.2 can easily be exploited at each level: whenever a candidate C (*i.e.*, a subset of $\mathbb{D}$) does not satisfy the constraints about either the Jaccard threshold β or the agony threshold α, it is removed from the candidate set so to skip the visit of all its supersets. Particularly, as computing the agony of a candidate C is much more time-consuming than computing Jaccard (*i.e.*, $\mathcal{O}(m^2 n)$ vs. $\mathcal{O}(n)$, where n and m are the number of nodes and arcs in the union graph $G(C)$, respectively), the first constraint checked by the algorithm is the one concerning Jaccard (Line 6). Only if this constraint

Algorithm 1 Bottom-up

```
Require: set of DAGs $\mathbb{D}$, thresholds $\alpha \in \mathbb{N}$, $\beta \in [0,1]$
Ensure: set $\mathbb{S}$ of all maximal subsets of $\mathbb{D}$ such that $|S| \geq 2$, $a(G(S)) \leq \alpha$,
    and $j(S) \geq \beta$, for all $S \in \mathbb{S}$
 1: $\mathbb{S} \leftarrow \emptyset$,  $\mathbb{C} \leftarrow \{\{D\} \mid D \in \mathbb{D}\}$
 2: while $\mathbb{C} \neq \emptyset$ do
 3:   $\mathbb{C}' \leftarrow generateCandidates(\mathbb{C})$
 4:   $\mathbb{C} \leftarrow \emptyset$
 5:   for all $C \in \mathbb{C}'$ do
 6:     if $j(C) \geq \beta$ then
 7:       if $a(G(C)) \leq \alpha$ then
 8:         $\mathbb{C} \leftarrow \mathbb{C} \cup \{C\}$
 9:         $\mathbb{S} \leftarrow \mathbb{S} \setminus \{S \in \mathbb{S} \mid S \subset C\} \cup \{C\}$
10:       end if
11:     end if
12:   end for
13: end while
```

is not violated, then the algorithm proceeds to compute the agony and check whether its value is within the corresponding threshold (Line 7).

The *generateCandidates* procedure invoked in Line 3 derives the set of candidates to be processed in the next level $i+1$ from the set of potential candidates $\mathbb{C}'$ that have passed the tests about the threshold α and β at level i. The procedure essentially performs a classic Apriori-like join step. Each pair of potential candidates C_1, C_2 in $\mathbb{C}'$ sharing a common prefix of length $i-1$ (*i.e.*, for which $|C_1 \cap C_2| = i-1$) is merged to form the set $C_{12} = C_1 \cup C_2$ of size $i+1$; such a set C_{12} will be then included in the candidate set $\mathbb{C}'$ only if all its subsets of size i are present in $\mathbb{C}$.

In order to further speed-up the execution, the *generateCandidates* procedure exploits the following simple result about Jaccard: given any two sets S_1, S_2, with $|S_1| \leq |S_2|$, the Jaccard value $j(\{S_1, S_2\})$ between such sets is upper-bounded by the size of the smaller-sized set divided by the size of the larger-sized one, *i.e.*, $j(\{S_1, S_2\}) \leq \frac{|S_1|}{|S_2|}$. Hence, a further test is performed on a candidate $C_{12} = C_1 \cup C_2$ that has passed all the tests required by the Apriori-like join phase: C_{12} is actually included in the final set $\mathbb{C}'$ only if the above constant-time-computable upper bound on Jaccard is no less than the threshold β.

A key advantage of the Bottom-up algorithm is the capability of exploiting the pruning rules stated in Fact 6.3.1 and Corollary 6.3.2, which allow a smart yet efficient search of the lattice. Moreover, due to its closeness to the classic frequent-itemset-mining Apriori algorithm, it is rather simple and easy-to-implement. Nevertheless, Bottom-up still suffers from a couple of major limitations:

1. The efficiency bottleneck of the algorithm is the computation of the agony, which, as said before, takes $\mathcal{O}(m^2 n)$ time. Even though the Bottom-up algorithm tries to minimize the number of agony computations by first checking the less expensive Jaccard constraint, unneeded agony computations may still arise. Indeed, each maximal summary S satisfies the property that all its subsets do not exceed the agony threshold (Corollary 6.3.2, Statement 2); this means that, for each of such maximal summaries S, all subsets of S have necessarily been involved into an agony computation in previous iterations that resulted in no pruning. For a bottom-up visit there is no way to avoid that. An idea could be to compute agony incrementally: however this is not even a viable solution, as even adding a single new arc might force the algorithm for agony computation to perform a number of operations comparable to re-doing the whole computation from scratch [66].

2. The space complexity of Bottom-up can be high, as the algorithm needs to keep in memory the union graph of all candidates of the current level to allow agony computations. On the other hand, the solution of loading at runtime the union graph of the various candidates would slow down the algorithm too much.

Hence, we devise a second algorithm to overcome the above issues.

6.3.2. The Up-and-down Algorithm

The Up-and-down performs a two-step traversal of the lattice $2^{\mathbb{P}}$ in which a bottom-up phase is followed by a top-down phase. The outline of Up-and-down is reported as Algorithm 2.

The first step of Up-and-down (Lines 2-11) is a bottom-up visit similar to the Bottom-up algorithm, but here performed considering only the Jaccard constraint (thus completely discarding agony). The aim of this step is to find the set $\mathbb{S}_j$ of all maximal summaries that satisfy Jaccard. Among

Algorithm 2 Up-and-down

```
Require: set of DAGs $\mathbb{D}$, thresholds $\alpha \in \mathbb{N}$, $\beta \in [0,1]$
Ensure: set $\mathbb{S}$ of all maximal subsets of $\mathbb{D}$ such that $|S| \geq 2$, $a(G(S)) \leq \alpha$,
    and $j(S) \geq \beta$, for all $S \in \mathbb{S}$
 1: $\mathbb{S}_j \leftarrow \emptyset$,  $\mathbb{C} \leftarrow \{\{D\} \mid D \in \mathbb{D}\}$
 2: while $\mathbb{C} \neq \emptyset$ do
 3:     $\mathbb{C}' \leftarrow generateCandidates(\mathbb{C})$
 4:     $\mathbb{C} \leftarrow \emptyset$
 5:     for all $C \in \mathbb{C}'$ do
 6:         if $j(C) \geq \beta$ then
 7:             $\mathbb{C} \leftarrow \mathbb{C} \cup \{C\}$
 8:             $\mathbb{S}_j \leftarrow \mathbb{S}_j \setminus \{S \in \mathbb{S}_j \mid S \subset C\} \cup \{C\}$
 9:         end if
10:     end for
11: end while
12: $\mathbb{S}_{\neg a} \leftarrow \{S \in \mathbb{S}_j \mid a(G(S)) > \alpha\}$
13: $\mathbb{S} \leftarrow \mathbb{S}_j \setminus \mathbb{S}_{\neg a}$
14: $M \leftarrow \max_{S \in \mathbb{S}_{\neg a}} |S|$
15: for $i = M, M-1, \ldots, 2$ do
16:     $\mathbb{S}_{\neg a}^{(i)} \leftarrow \{C \in \mathbb{S}_{\neg a} \mid |C| = i\}$
17:     for all $C \in \mathbb{S}_{\neg a}^{(i)}$ do
18:         if $a(G(C)) \leq \alpha$ then
19:             $\mathbb{S} \leftarrow \mathbb{S} \cup \{C\}$
20:         else
21:             $\mathbb{S}_{\neg a} \leftarrow \mathbb{S}_{\neg a} \cup \{C' \in \mathbb{S}_j \mid C' \subset C, |C'| = i-1, \nexists S \in \mathbb{S} : S \supset C'\}$
22:         end if
23:     end for
24: end for
```

these summaries, there will clearly be some that do not satisfy agony; such summaries are collected into the set $\mathbb{S}_{\neg a}$ (Line 12). The second step of the algorithm (Lines 15-24) restarts from these summaries that violate the agony constraint and performs a top-down visit of the lattice aimed at discovering the summaries whose agony is instead within the threshold α. The top-down visit is performed in a breadth-first fashion (like the bottom-up counterpart), starting from level $i = M$, where M denotes the maximum size of a summary in $\mathbb{S}_{\neg a}$ (in general $\mathbb{S}_{\neg a}$ may in fact contain summaries of different size). For each level i, the algorithm computes the agony of all candidate summaries in $\mathbb{S}_{\neg a}^{(i)}$, which denotes the set of all summaries in $\mathbb{S}_{\neg a}$ of size i (Line 18). Each candidate $C \in \mathbb{S}_{\neg a}^{(i)}$ satisfying the agony constraint is added to the solution set $\mathbb{S}$ (Line 19): indeed, according to (the second statement of) Corollary 6.3.2, all subsets of C are guaranteed to satisfy agony as well, then no backtracking is further needed from C. If the agony constraint is violated, the candidate C in $\mathbb{S}_{\neg a}^{(i)}$ is processed so to add to the candidate set to be considered in the next iteration all $(i-1)$-sized subsets of C that do not have any superset in the current solution set $\mathbb{S}$ (Line 21).

The main advantage of the two-step lattice traversal of the Up-and-down algorithm is that, in most cases, it significantly reduces both the total number of agony computations and the space complexity, thus offering a valid solution to the issues of the Bottom-up algorithm discussed at the end of Section 6.3.1. Indeed, the bottom-up phase of Up-and-down completely ignores the agony constraint, whose computation, as said, constitutes a bottleneck in terms of both time and space. The agony constraint is considered only in the subsequent top-down phase, where, however, the number of agony computations is expected to be less than the agony computations performed by a purely bottom-up strategy. For a better explanation, let us consider the following example.

Example *Let* $\mathbb{D} = \{A, B, C, D, E, F, G, H, I, J, K, L\}$ *and assume that the bottom-up phase of the* Up-and-down *algorithm produces the set* $\mathbb{S}_j = \{ABCD, EFGH, IJKL\}$ *(where* $ABCD$ *is a shorthand for* $\{A, B, C, D\}$*). Assume also that, among the elements of* $\mathbb{S}_j$*,* $IJKL$ *violates the agony constraint, while* $ABCD$ *and* $EFGH$ *do not,* i.e.*, assume that* $\mathbb{S}_{\neg a} = \{IJKL\}$*.*

For the elements of $\mathbb{S}_j \setminus \mathbb{S}_{\neg a}$ *(*i.e.*,* $ABCD$ *and* $EFGH$*), the speed-up achieved by the* Up-and-down *algorithm with respect to* Bottom-up *is guaranteed and quite evident: for each of such elements,* Up-and-down *computes agony only once, while* Bottom-up *would perform a number of agony computations proportional to the number of subsets of that element (*i.e.*, 11 agony computations for a 4-sized summary). Also the space requirements of* Up-and-down *are significantly less, as, for the bottom-up phase,* Up-and-down *needs to keep in memory only the set of nodes of each candidate (because only the node set is needed to compute Jaccard), unlike the* Bottom-up *algorithm that requires in memory the entire graph* $G(C)$ *for each candidate* C *(needed for computing agony).*

Concerning the elements of $\mathbb{S}_{\neg a}$*, instead, the speed-up and the memory savings achieved by* Up-and-down *depend on how further the top-down phase*

	$\lvert\mathbb{D}\rvert$	n_{min}	n_{max}	n_{avg}	m_{min}	m_{max}	m_{avg}
`Twitter`	8,888	12	13,547	66	11	240,153	347
`Last.fm`	51,495	6	472	24	5	2,704	39
`Flixster`	11,659	14	16,129	561	13	85,165	1,561
`WikipediaBrowsing`	5,477	4	125	9	4	131	9

Table 6.1: Characteristics of the datasets used in the experiments: number of propagations/DAGs ($\lvert\mathbb{D}\rvert$); minimum, maximum, and average number of nodes in a DAG in $\mathbb{D}$ (n_{min}, n_{max}, n_{avg}); and minimum, maximum, and average number of arcs in a DAG in $\mathbb{D}$ (m_{min}, m_{max}, m_{avg}).

needs to go. For instance, assume that the actual candidates that satisfy agony are $\{JKL, IKL, IJK\}$. This way, the top-down phase of UP-AND-DOWN *would last only one level, thus still guaranteeing a speed-up and memory saving with respect to* BOTTOM-UP*. But if, as another example, the actual set of candidates satisfying agony is instead the singleton $\{IJ\}$, the* UP-AND-DOWN *algorithm would need to visit a large portion of the lattice under $IJKL$ before encountering the set IJ, whereas* BOTTOM-UP *would have processed IJ quite soon.*

According to the above reasoning, the main conclusion that may be drawn here is that UP-AND-DOWN guarantees better efficiency and smaller space complexity than BOTTOM-UP in most of the cases. Nevertheless, the BOTTOM-UP algorithm still remains preferable in cases where the time (and space) spent by UP-AND-DOWN in the top-down phase is predominant. This may happen, *e.g.*, when small agony thresholds α and/or large Jaccard thresholds β are involved. As we will show in Section 6.4, this conclusion is also confirmed by experimental evidence.

6.4. Experimental Evaluation

We provide here experimental evidence of the performance of our BOTTOM-UP and UP-AND-DOWN algorithms. We experiment with four real-world datasets, whose main characteristics are summarized in Table 6.1. For a description of the datasets, see Section 3.3. Three datasets (*i.e.*, `Twitter`, `Last.fm`, and `Flixster`) come from the domain of information propagation in a social network (application scenario #1 in the beginning of the

	α					
β	10	20	30	40	50	60
0.7	515	646	537	416	410	411
0.8	121	128	128	120	116	116
0.9	44	51	51	47	45	45

(a) `Twitter`

	α			
β	3	5	7	10
0.7	12,208	12,292	12,306	12,293
0.8	5,126	5,136	5,128	5,132
0.9	1,895	1,903	1,900	1,902

(b) `Last.fm`

Table 6.2: Number of maximal summaries extracted from `Twitter` (a) and `Last.fm` (b).

chapter), while the remaining one (*i.e.*, `WikipediaBrowsing`) concerns the web-browsing domain (application scenario #2 in the beginning of the chapter).

In the following sections, we discuss the results achieved by the proposed algorithms from both a quantitative (Section 6.4.1) and a qualitative (Section 6.4.2) viewpoint. We use `Twitter` and `Last.fm` mainly for quantitative evaluation, while we resort to `WikipediaBrowsing` and `Flixster` for qualitative evaluation.

All algorithms are implemented in JAVA and all experiments are performed on a single machine with Intel Xeon CPU at 2.20GHz and 48GB RAM.

6.4.1. Quantitative Evaluation

We report here quantitative results achieved by our Bottom-up and Up-and-down algorithms on `Twitter` and `Last.fm`.

	α					
β	10	20	30	40	50	60
0.7	8 (3.3)	9 (4.4)	11 (4.6)	12 (3.9)	14 (4.1)	14 (4.1)
0.8	5 (2.4)	9 (2.7)	10 (2.8)	12 (2.9)	13 (2.7)	13 (2.7)
0.9	5 (2.4)	8 (2.7)	8 (2.8)	10 (2.7)	11 (2.7)	11 (2.7)

(a) `Twitter`

	α			
β	3	5	7	10
0.7	13 (3.3)	13 (3.3)	13 (3.3)	13 (3.3)
0.8	13 (2.9)	13 (2.9)	13 (3.4)	13 (2.9)
0.9	11 (2.5)	11 (2.5)	11 (2.5)	11 (2.5)

(b) `Last.fm`

Table 6.3: Maximum and average size (*i.e.*, number of propagations) of the maximal summaries extracted from `Twitter` (a) and `Last.fm` (b).

We test our algorithms with Jaccard thresholds $\beta \in [0.7, 0.9]$ and agony thresholds $\alpha \in [10, 60]$ (`Twitter`) or $\alpha \in [3, 10]$ (`Last.fm`).

General Characterization. Tables 6.2, 6.3, and 6.4 show general statistics about the summaries extracted. In particular, Table 6.2 reports on the number of maximal summaries, while the remaining tables show statistics about the size of the summaries: maximum and average number of propagations in a summary (Table 6.3) and maximum and average number of nodes and arcs in the union graph of a summary (Table 6.4).

As expected, the size of the summaries increases as the agony threshold α increases and/or the Jaccard threshold β decreases (Tables 6.3–6.4), because this corresponds to less selective constraints. As far as the number of summaries (Table 6.2), this is not necessarily true, because it may happen that, for a less restrictive constraint, the number of maximal summaries is less but the summaries include a larger number of DAGs.

	nodes					
β	$\alpha = 10$	$\alpha = 20$	$\alpha = 30$	$\alpha = 40$	$\alpha = 50$	$\alpha = 60$
0.7	2,252(34)	2,252(33)	2,252(35)	2,252(37)	2,252(37)	2,252(37)
0.8	2,252(55)	2,252(52)	2,252(53)	2,252(55)	2,252(57)	2,252(57)
0.9	2,185(73)	2,185(65)	2,185(65)	2,185(70)	2,185(72)	2,185(72)

	arcs					
β	$\alpha = 10$	$\alpha = 20$	$\alpha = 30$	$\alpha = 40$	$\alpha = 50$	$\alpha = 60$
0.7	13,003(129)	13,003(137)	13,003(141)	13,003(142)	13,003(142)	13,003(142)
0.8	13,003(221)	13,003(220)	13,003(224)	13,003(233)	13,003(237)	13,003(237)
0.9	6,771(214)	6,771(201)	6,771(199)	6,771(208)	6,771(212)	6,771(212)

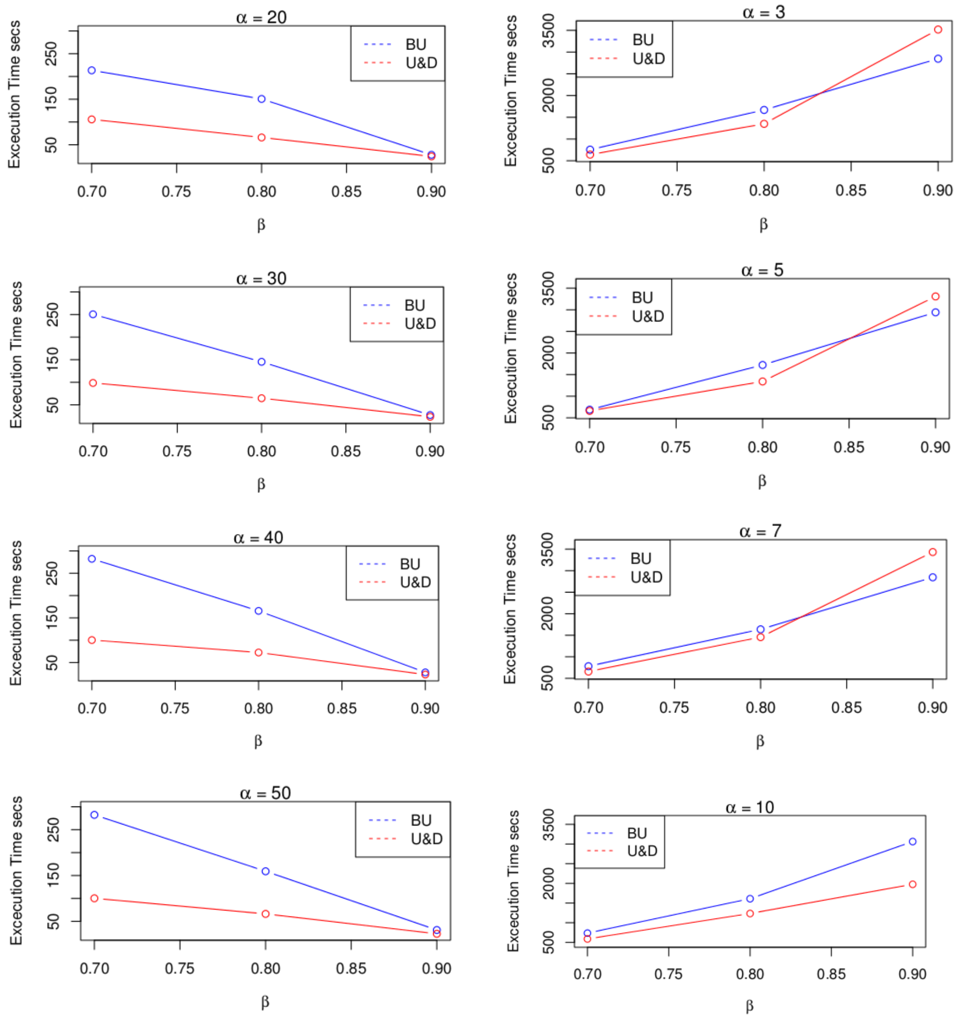
(a) `Twitter`

	nodes			
β	$\alpha = 3$	$\alpha = 5$	$\alpha = 7$	$\alpha = 10$
0.7	483 (22)	493 (22)	493 (22)	496 (22)
0.8	376 (20)	376 (20)	394 (19)	407 (19)
0.9	333 (17)	333 (17)	333 (17)	333 (17)

	arcs			
β	$\alpha = 3$	$\alpha = 5$	$\alpha = 7$	$\alpha = 10$
0.7	2,904 (35)	2,960 (36)	2,990 (37)	3,019 (38)
0.8	1,998 (32)	1,998 (32)	2,126 (31)	2,221 (30)
0.9	1,621 (22)	1,621 (22)	1,621 (22)	1,621 (22)

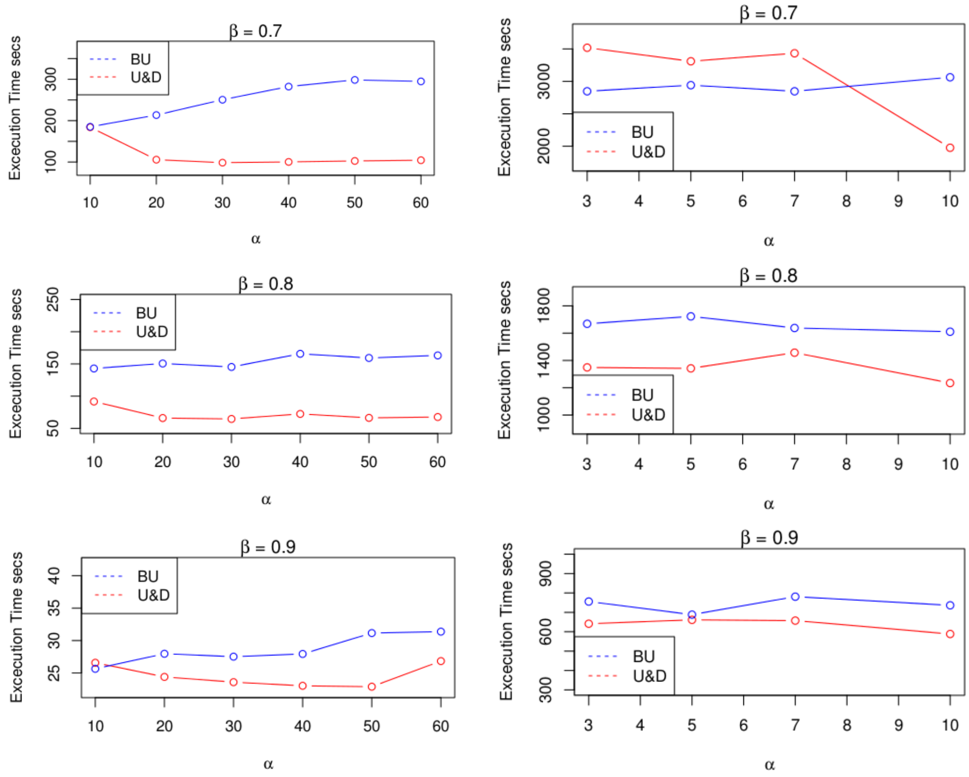
(b) `Last.fm`

Table 6.4: Maximum and average number of nodes/arcs of the union graph of the maximal summaries extracted from `Twitter` (a) and `Last.fm` (b).



(a) `Twitter` (b) `Last.fm`

Figure 6.4: Running times (seconds) of the proposed Bottom-up (bu) and Up-and-down (u&d) algorithms on `Twitter` (left) and `Last.fm` (right) with varying Jaccard threshold: $\beta \in [0.7, 0.9]$.



(a) `Twitter` (b) `Last.fm`

Figure 6.5: Running times (seconds) of the proposed Bottom-up (bu) and Up-and-down (u&d) algorithms on `Twitter` (left) and `Last.fm` (right) with varying the agony threshold: $\alpha \in [10, 60]$ (`Twitter`) and $\alpha \in [3, 10]$ (`Last.fm`).

Efficiency Evaluation. The running times of our algorithms are shown in Figures 6.4 and 6.5. Particularly, in Figure 6.4 we show the results with varying the Jaccard threshold β while keeping fixed the agony threshold α; in Figure 6.5, instead, we keep β fixed and show the times with varying α.

Figure 6.4 clearly shows that the running times of both Bottom-up and Up-and-down are decreasing as the Jaccard threshold β increases: this is expected as the larger β, the smaller the number of candidates to be processed (larger β denotes a more restrictive constraint). The two algorithms instead differ from each other when looking at the behavior when varying the agony threshold α (while keeping β fixed). Indeed, as Figure 6.5 shows, the Bottom-up times are increasing as α increases, while the opposite happens for Up-and-down (there are some fluctuations, but mainly due to bookkeeping). This is again expected, because larger values of α imply more candidates to be processed at each level of the Bottom-up algorithm. On the other hand, the larger α is, the more likely is that the summaries found at the end of the first phase of the Up-and-down algorithm satisfy the agony constraint, thus leading to a more lightweight (and faster) top-down phase.

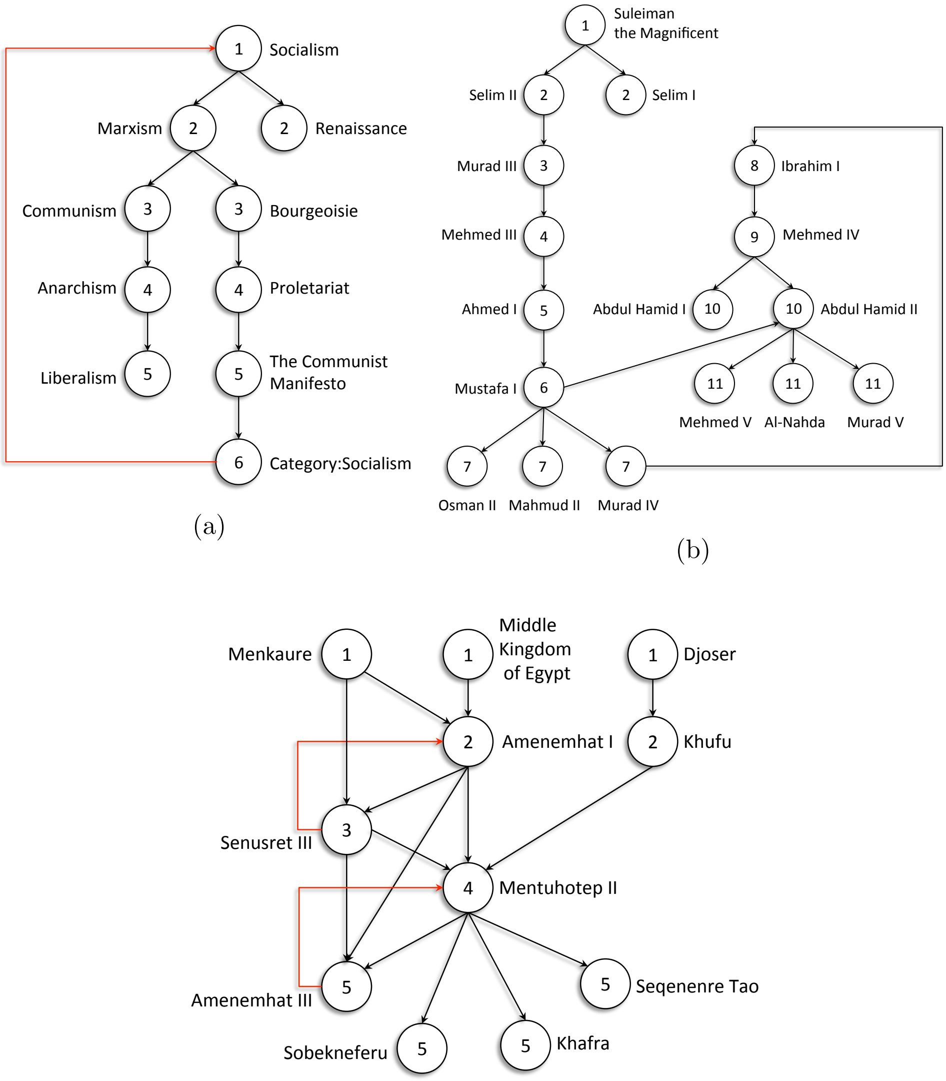
As far as the comparison of the two algorithms with each other, the results confirm what discussed in Section 6.3.2. Indeed, we can observe here that the Up-and-down algorithm is more efficient than Bottom-up in most settings, with gains up to 65% (`Twitter`) and 35% (`Last.fm`). Nevertheless, in some cases Bottom-up outperforms Up-and-down. This happens, for instance, for small agony thresholds α and/or large Jaccard thresholds β (indeed, the gain achieved by Up-and-down over Bottom-up is overall decreasing as α decreases and/or β increases). The reason is that smaller α and/or larger β imply a smaller number of candidates to be processed in the bottom-up lattice-traversing phase, which is an advantage for Bottom-up (it reduces the number of agony computations), but a disadvantage for Up-and-down, as it increases the time spent in the top-down phase.

In conclusion, hence, although in most cases the fastest algorithm is Up-and-down, the efficiency of the two algorithms depends on the selectivity of the constraints used, thus leading to cases where Bottom-up is instead preferable.

6.4.2. Qualitative Evaluation

We analyze here the summaries extracted by our algorithms from a qualitative viewpoint.

Wikipedia. Figure 6.6 shows examples of union graphs of summaries from `WikipediaBrowsing`. Graph (a) consists of two main parts: the first part is a loop going through *Socialism* and *Proletariat*; the second is a branch that ends up in *Liberalism*. One could imagine that the first part is due to users who would like to explore the topic of socialism, while the second one is browsing to a different one. Graph (b) is a chronological sequence of the sultans from the Ottoman Empire. This summary has no agony. The nodes of Graph (c) are pharaons of Ancient Egypt. The ranking in the summary is almost the same as the chronological order. The exceptions are



(a)

(b)

(c)

Figure 6.6: Three examples of graphs of summaries extracted from Wikipedia. The numbers inside the nodes correspond to one optimal ranking. Links that violate the ranking are represented in red.

Mentuhotep II, who reigned before *Amenemhat I*, and *Khafra*, who reigned between *Khufu* and *Menkaure*.

We can observe that each summary is related to a train of thought. The hierarchy of the summaries defines a temporal sequence of concepts that are presented one after another. For example, Graph (a) consists in the description of the socialism, passing through Karl Marx, and the connections to other political philosophies, while Graph (b) goes through the history of the Ottoman Empire. Applications might exploit this to organize knowledge and present it to people. A summary could also inspire a lecture since it condenses the way in which many people move through concepts. Moreover, as we can see in the examples, nodes in a summary are topically similar to each other. Summaries could hence help categorizing the knowledge space. Finally, summaries can be used for online contextual recommendation of sequences of pages. Given the current user session, one can fit a summary and predict not only the next page the user will visit but also the whole future browsing path. This recommendation is contextualized since it leverages the browsing history of the user.

Flixster. In Table 6.5 we report two examples of summaries extracted from `Flixster`. For each of such summaries, we show some information (*i.e.*, title, genre, and director) of the movies corresponding to the DAGs of the propagations in that summary. We can observe that movies in the same summary exhibit some homogeneity of genre and type of audience. In this context, one might exploit summaries to discover early adopters for a topic (group of movies), by looking at the ranking that is coupled with each summary. Applications may leverage this information to target recommendations of new movies sharing similar topics with the summary, so to guarantee the desired spread over the network.

6.5. Discussion

In this chapter we studied for the first time the problem of extracting informative summaries from a database of propagations. We defined the summaries of interest using two constraints: the first constraint is defined based on the Jaccard coefficient, while the definition of the second one relies on the graph-theoretic concept of “agony” of a graph. We showed that both constraints satisfy the downward-closure property, thus enabling Apriori-like algorithms. We developed two algorithms that visit the search space in different ways and apply to various real-world datasets.

Title	Genre	Director
Man of the Year	comedy/drama/romance	B. Levinson
Conversations With Other Women	comedy/drama/romance	H. Canosa
Step Up 2 the Streets	drama/music/romance	J. M. Chu
Brubaker	drama	S. Rosenberg

Title	Genre	Director
Prick Up Your Ears	biography/drama	S. Frears
Crooklyn	comedy/drama	S. Lee
Boy Culture	drama/comedy	Q. A. Brocka
One Eyed Jacks	western/drama/action adventure	M. Brando
Cop and a Half	family/comedy	H. Winkler
Operation Pacific	drama/war/action adventure/comedy	G. Waggner

Table 6.5: Two summaries extracted from `Flixster`: title, genre, and director of the movies corresponding to the DAGs of the propagations of each of the two summaries.

The algorithms presented in this chapter require the α and β parameters to be set manually. More sophisticated algorithms may automatically detect the best parameters by performing a search in the α and β space. Alternatively, they can avoid setting rigid thresholds by making the constraints soft [19].

In the next chapters we propose models for browsing. The first model is aimed at understanding which are the features of browsing that best predict user actions. This will provide guidelines for the creation of more complex models.

Chapter 7

Models of Browsing

In this chapter, we take the first steps towards modeling user browsing behavior. A problem when building models for large datasets is to understand which pieces of information are useful to predict the future actions of the user and which are not. This problem is known as *feature selection* [101, 108].

Automatically understanding which feature is useful and which is not is important for a number of reasons. First of all, it helps improving the performance by removing uninformative features. This is essential, particularly in the context of news, where stories have a very short lifespan and are constantly being created and changed. Models should react to these events and update constantly. Additionally, understanding which feature can better predict what the user will do next gives insight about the behavior of the user while browsing, giving a quantitative evaluation of the observations of Chapters 4 and 5.

In this chapter, we aim at modeling how users browse news portals. Using a sample from click logs of Yahoo News (see Section 3.3.4), we focus on features of user browsing and model sessions using a simple yet reliable model that captures a wide variety of features (*e.g.*, geographical location, user interests, context of the session, *etc.*). We evaluate the model on the dataset, learning the right balance of the features.

Our contributions are:

- We propose a simple model of news article browsing. The model takes as input the state of the news article at the time when the user is accessing it (*e.g.*, freshness, time since last comment, context). The model is able to learn the mixture of the features, *i.e.*, weight the features based on their contribution to the model.

- We show the result of the learned model, in terms of the weights of the features. We show that the behavior of users accessing Yahoo News from different websites is different from the one of the people browsing inside it. More specifically, we show that people entering Yahoo News target fresh news and highly commented ones. On the contrary, users already in Yahoo News are not influenced by the comments.

The rest of the chapter is organized as follows. Section 7.1 presents the model of browsing sessions. We perform the experiments on the dataset and report the results in Section 7.2.

7.1. Modeling

In this section we model user sessions using a hybrid approach. With the term "hybrid" we mean that the model is composed by a set of features that capture different aspects of the browsing behavior. We present the features of the model and finally join them into a single structure.

We use the `YahooNewsBrowsing-USA` dataset (Section 3.3.4). The dataset contains the web pages seen by users (page views), grouped into sessions. In addition, it contains the timestamps of the actions people perform on news articles, namely commenting on them and sharing them on social media.

In the following sections we refer to the current page accessed by the user in the browser as X, and to its category as cat_X. Moreover, we use the term *external* to refer to page views whose referrer URL is not Yahoo News, else we refer to them as *internal*.

7.1.1. Referrer Domain

As seen in Chapter 4, there is a precious piece of information that can be collected by the web server: the *referrerURL* (that we denote as r), *i.e.*, the last web page seen by the user before the first page is requested to the web server. This information gives access to at least a partial context from where the session originates. For example, many sessions are originated from search or social media and end up in a news article.

	shine.*	facebook.com	finance.*	search.*	<other>
shine.yahoo.com	0	0.13	0.5	0.29	0.39
facebook.com	0.14	0	0.5	0.22	0.25
finance.yahoo.com	0.42	0.53	0	0.34	0.59
search.yahoo.com	0.28	0.19	0.33	0	0.31
<other>	0.39	0.24	0.48	0.31	0
<internal>	1.82	1.74	1.44	0.89	0.65

(a) Referrer domain $p(X \mid r)$. Column headers have been abbreviated due to space constraints.

	Male	Female	Unknown
Male	0	0.07	0.02
Female	0.08	0	0.03
Unknown	0.02	0.03	0

(b) Genders $p(X \mid g)$.

	California	Florida	Illinois	New York	Ohio	Texas	%
California	0	0.01	0.01	0	1.44	0.01	15.63
Florida	0.01	0	0	0.01	0	0	9.34
Illinois	0.01	0	0	0.01	0	0	5.47
New York	0	0	0.01	0	1.42	0.01	5.29
Ohio	1.43	0	0	1.46	0	0	4.74
Texas	0.01	0	0	1.05	0	0	3.80

(c) Locations $p(X \mid \ell)$.

Table 7.1: Comparison of the page view distributions for different values of the features. For each feature, we take the top-K values and compute the Kullback-Leibler divergence among the distributions of article categories cat_X. The higher the value of the KL-divergence, the more different are the distributions.

The first component of the model is therefore the referrer domain,

$$p(X \mid r) \propto n(r \rightarrow cat_X) + \varepsilon \tag{7.1}$$

where $n(r \rightarrow cat_X)$ is the number of page views originated from r which end up in category cat_X. We apply a smoothing parameter ε in order to avoid zero probabilities.

To give an initial proof of the importance of the referrer domain, we compute the distribution of sessions coming from each referrer domain over the categories of articles in Yahoo News. Manual inspection shows that distributions are different across referrer domains. To quantify which are the most diverse referrer domains, we compute the Kullback-Leibler divergence [93] among the distributions $p(X \mid r)$ of the referrer domains. Table 7.1a shows the KL-divergence for the top 4 referrer domains. In addition, the table also shows the divergence w.r.t. all the remaining domains (`<other>`), and w.r.t. the navigation inside Yahoo News (the `<internal>` row).

From the table we can see that different referrer domains show indeed different distributions over the categories of news articles. For example, `shine.yahoo.com` and `facebook.com` are similar one to the other but are both different from `finance.yahoo.com`. Moreover, we can observe that distributions of the referrer domains are different from the browsing *inside* Yahoo News, *i.e.*, $p(X \mid$ `<internal>`$)$ (see last row of the table).

7.1.2. Markov Chain

Sessions are sequences of page views. After entering from a referrer domain, the user visits a series of pages in Yahoo News. It is natural to consider in the model the order of the pages. We will use a simple Markov chain to capture the information about transitions.

We measure the probability of transitioning from an item to another one, $p(X_i \mid X_{i-1})$, based on the past observations in the server logs. The probability is defined on the level of news article categories. Similarly to Equation 7.1, we smooth the distribution using a parameter ε.

$$p(X_i \mid X_{i-1}) \propto n(cat_{X_{i-1}} \rightarrow cat_{X_i}) + \varepsilon$$

where $n(cat_{X_{i-1}} \rightarrow cat_{X_i})$ is the number of transitions from the category of X_{i-1} to the category of X_i.

7.1.3. Personalization

We now focus on the user. Indeed, the browsing behavior of users may depend on the personal taste or on demographic factors. For this reason, we model the dependency of the browsed content from the profile of the user.

The most specific information about the user that can be extracted from server logs is the anonymized identifier of logged-in users, u. Secondly, we use the geographical location ℓ. This information can be extracted from the IP address that originated the request to the server. Additional information about the user is the *gender* and *age*, as stated by the user in the profile. The information is in decreasing order of accuracy. Indeed the location may be inaccurate due to nodes of the communication network (*e.g.*, proxy servers or Internet Service Provider gateways) hiding the true origin of the request. The gender and the age are stated by the user with no guarantee of correctness. However, all information has been used in other works (*e.g.*, [152]) with good results.

Similarly to Section 7.1.1, we compute the distributions over the category of news articles for different user segments and compute the KL-divergences. Table 7.1b shows the KL-divergence across genders and Table 7.1c shows the same across locations.

We can see that the divergence is not high in the case of gender and location. However, there are still some differences in some cases, as for example between males and females and between some states of the United States (*e.g.*, California and New York against Ohio).

7.1.4. Freshness

Content *freshness* f, *i.e.*, the recency of the news article, is an important factor influencing content consumption. In our dataset, we observe that, after one day, news articles receive around 70% of their total number of visits and that 90% of the users read news articles before the fifth day since the publication.

To estimate the speed at which people consume content, we plot the cumulative distribution of visits depending on the time since the publication of the article (Figure 7.1). We can see that data follows a power law, therefore, we can write:

$$p(X \mid f) = power(\Delta t_f) = a \times (\Delta t_f)^b$$

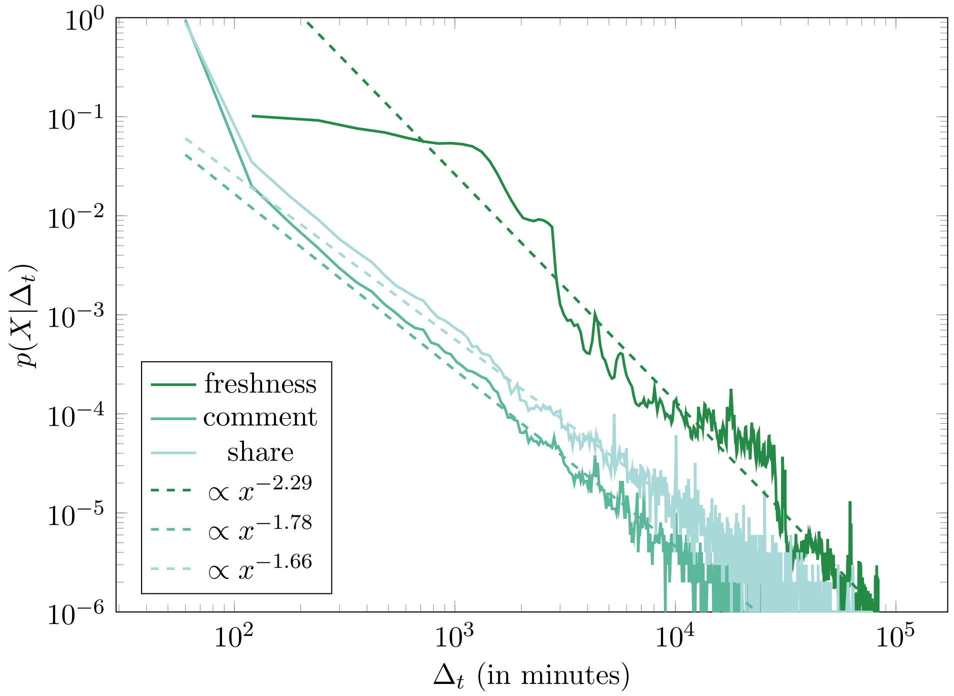


Figure 7.1: Distribution of time since publication of the article (dark green), since last comment action (medium green), and since last share action (light green). The dashed lines are the Maximum Likelihood Estimator of the distributions.

where Δt_f is the time since the publication of the article.

The parameters of the power law a and b are estimated using the Maximum Likelihood Estimator [37]. We plot the approximating curve in Figure 7.1 (dark green dashed line).

7.1.5. Social Signal

The last component of the model is the *social signal*, *i.e.*, the information extracted from the user contribution to social networking platforms, such as Facebook, Twitter, or Tumblr. News are constantly shared by users on social media and it is commonly understood that the explicit actions of users in social media give a good indicator of the popularity of news articles. Here, we use two features about user social actions: comments c

and shares s. Comments are very frequent actions done by users in Yahoo News, and long discussions are born around trending articles. Shares are used to spread the news outside Yahoo News, *e.g.*, on social media or to email recipients. It is natural to consider both comments and shares in our model since they are explicit actions, which have an impact internally (comments) and externally (shares) in Yahoo News.

Both comments and shares have a short lifespan, rarely going beyond the day of the publication. The time distribution of comments is therefore biased towards the publication of the article. However, after the first burst short after the publication of articles, there are other peaks of commenting activity. This may be because users tend to visit and comment "old" articles, possibly to reply to the comments of other users. Thus, the probability of commenting may be formulated not based on the time since the publication of the article, but based on the time since the last comment. This formulation accounts for both "fresh" comments, *i.e.*, those appearing soon after article's publication, and "old" comments, *i.e.*, those appearing in the peaks.

By manually inspecting the distribution of clicks as a function of the time since last comment, we see that it resembles a power law distribution (Figure 7.1, medium green). As a result, we model the probability of a particular item as:

$$p(X \mid c) = power(\Delta t_c)$$

where Δt_c is the time since last comment.

The same is valid for shares s (light green in Figure 7.1).

Comparison between Freshness and Social Signal. Since freshness f (Section 7.1.4), comments c, and shares s are all based on time, we investigate how related the features are. To do so, we compute the Pearson product-moment correlation coefficient ρ of the features in the train set. We observe that freshness is correlated neither to comments ($\rho_{f,c} \simeq 0.131$) nor to shares ($\rho_{f,s} \simeq 0.124$). However there is a higher correlation between comments and shares ($\rho_{c,s} \simeq 0.565$). As a result we keep all the features in the model.

7.1.6. The Complete Mixture Model

We build the session model by mixing all features. Each feature is weighted by a factor λ_k that represents its importance.

Let $C = \{u, g, a, \ell, f, c, s\}$ be the set of features, containing the user u, the gender g, the age a, the geographical location ℓ, the freshness f, and the social signal coming from comments c and shares s. The likelihood of a complete session is given in Equation 7.2, where we split the contribution of the first (Equation 7.3) and the next page views (Equation 7.4).

$$p(\{X_1, \ldots, X_N\} \,|\, \Theta) = p(X_1 \,|\, \Theta) \prod_{i=2}^{N} p(X_i \,|\, X_{i-1}, \Theta) \tag{7.2}$$

$$p(X_1 \,|\, \Theta) = \lambda_r \, p(X_1 \,|\, r) + \sum_{c \in C} \lambda_{c,e} \, p(X_1 \,|\, c) \tag{7.3}$$

$$p(X_i \,|\, X_{i-1}, \Theta) = \lambda_m \, p_t(X_i \,|\, X_{i-1}) + \sum_{c \in C} \lambda_{c,i} \, p(X_1 \,|\, c) \tag{7.4}$$

The features in C are shared among all page views, while the referrer domain r is only used from page views coming from other domains, and the Markov chain m is only used for page views originated internally in Yahoo News. The referrer and the Markov chain models both take into account the influence of the previous page view on the current one. The other features in C take into account only the current page view.

The basic idea behind the model is the following. The first page view of the session is modeled as a mixture of the referrer and of the other features in C. For the next actions, we perform a mixture between the Markov chain and the other features in C.

Parameters Θ contain the mixing coefficients λ_c as well as the parameters of the power law used for freshness, comments, and shares.

7.1.7. Cluster Model

The model can be naturally extended to cluster the data by adding a cluster variable k for each page view. Each cluster will have its own set of parameters Θ_k.

Figure 7.2 shows the complete cluster model. We introduce two hidden variables, whose values will be learned. The first variable, z_i, is a 1-of-K variable that encodes to which cluster the page view X_i belongs. The second variable, γ_k, encodes which feature is responsible for generating the page views in cluster k.

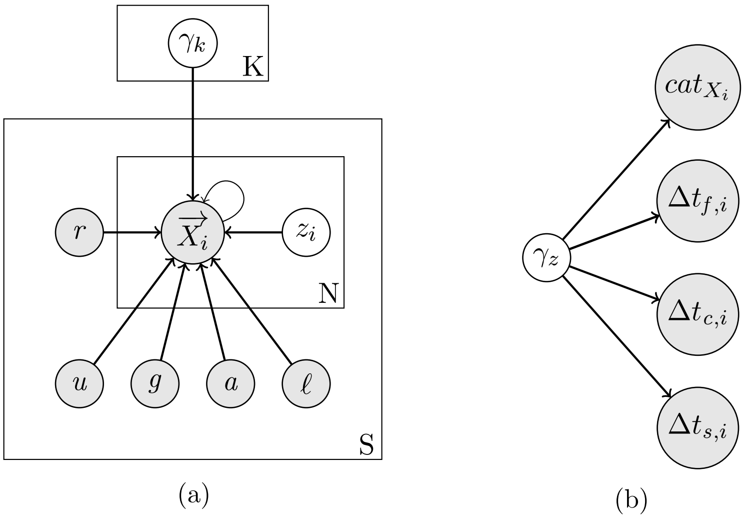


Figure 7.2: Plate model for the complete model (a) and for the page view $\overrightarrow{X_i}$ (b). Shaded nodes are observed.

7.1.8. Complexity

The space complexity of the model consists in the space needed to store the distributions. Given the set of news article categories C, the set of external referrers R, and the set of users U, then the complexity is $O(|C| \cdot (|R| + |U| + |C|))$, where $|R|$ is the number of referrer domains, $|C|$ is the number of categories of news articles, and $|U|$ is the number of users. In the equation we only considered the most space consuming probability distributions, namely r, u, and m.

7.1.9. Parameter Estimation

The parameters of the model Θ are estimated using the Expectation Maximization (EM) algorithm.

Expectation Step. First we compute the joint probability of the page view belonging to the cluster k and being generated by it using the feature

c applying the Bayes' theorem:

$$\begin{aligned} p(z_i = k, \gamma_k = c \,|\, X_i) &= \frac{p(X_i \,|\, z_i = k, \gamma_k = c)\, p(z_i = k, \gamma_k = c)}{p(X_i)} \\ &\propto \underbrace{p(X_i \,|\, c)}_{\text{observed}}\, p(c \,|\, k)\, p(k) \end{aligned}$$

The distributions $p(X_i \,|\, c)$ are computed offline using the training set. We also compute the parameters of the power law distributions using the Maximum Likelihood estimator [37].

The Expectation step can be easily parallelized, since each page view is independent from the other ones.

Maximization Step. We accumulate over all page views and compute the parameters of the model.

$$p(k, c) \propto \sum_{X_i} p(z_i = k, \gamma_k = c \,|\, X_i) \tag{7.5}$$

From the result of Equation 7.5 we can compute $p(k)$, the prior of clusters, and $p(c \,|\, k)$, the probability of the features for each cluster.

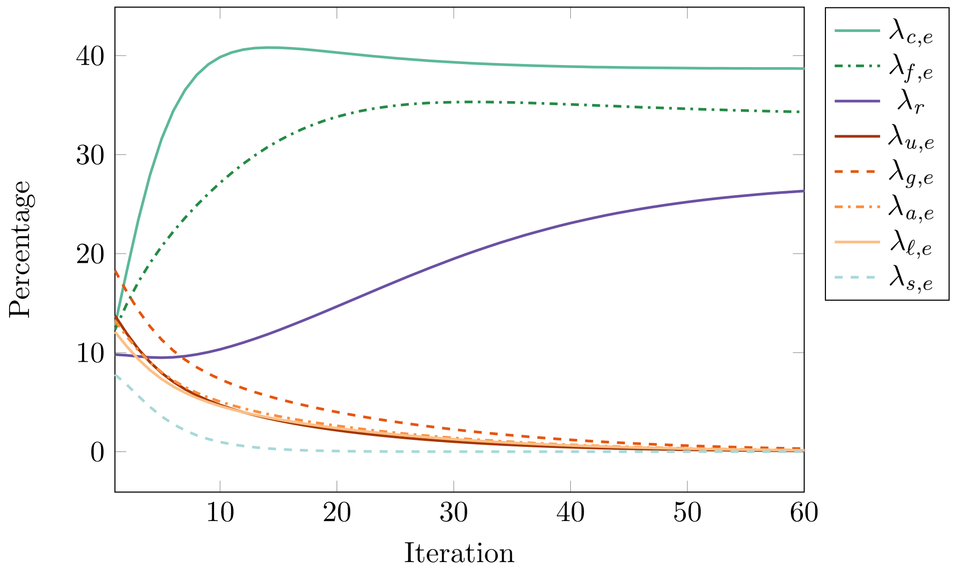
7.2. Experiments

In this section, we evaluate the performance of the model under different points of view. For the evaluation, we split the dataset in two sets, one used for training (first 80% of sessions in order of time), and one for testing (remaining 20%).

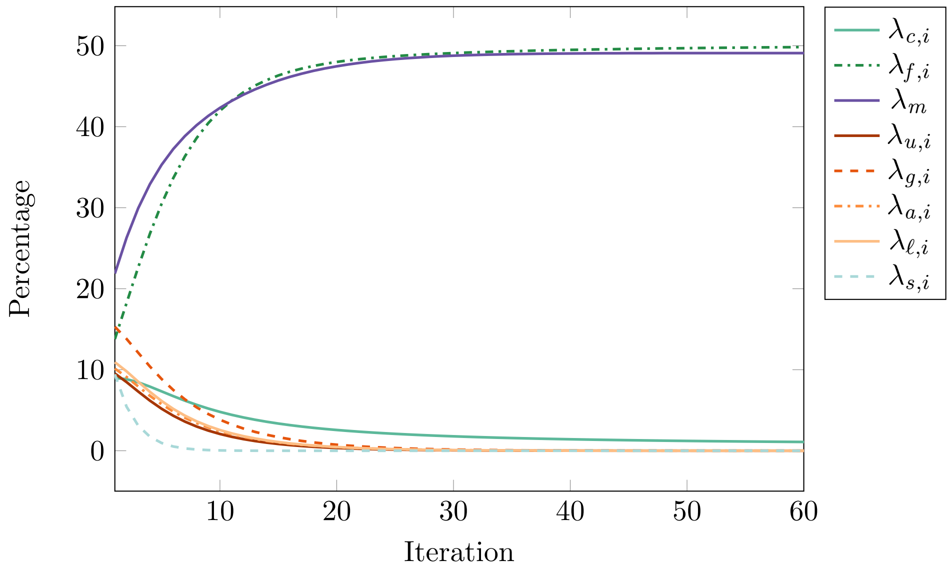
7.2.1. Model Training

We train the model described in Section 7.1.7. Figure 7.3 shows the values of the parameters for the 1-cluster model $\mathcal{M}_1$. We can see immediately that the contributions of the components to the model is different depending on whether the user is coming from a different domain (Figure 7.3a) or is browsing internally in Yahoo News (Figure 7.3b).

When the user is entering from a different domain, the predominant component is comments c, followed by freshness f, and referrer domain r. This means that most of the traffic coming from outside could be predicted just



(a) Parameters for external page views.



(b) Parameters for internal page views.

Figure 7.3: Values of the parameters of the 1-cluster model $\mathcal{M}_1$ at various iterations of the EM algorithm.

considering the time of the visit of the user to Yahoo News (*i.e.*, using comments and freshness). There is however a contribution of the referrer domain r. This means that, as introduced qualitatively in Chapter 4 and Section 7.1.1, the traffic coming from particular referrer domains ends up in different categories of news articles.

Concerning the internal traffic, most of the contribution is due to the freshness f and to the Markov chain m. Surprisingly, comments c, which had such a great importance for external traffic, is almost insignificant in this case. The contribution of freshness could be explained by the fact that news sites promote fresh news articles by putting them on the top of the page.

The 1-cluster model already shows its potential to capture differences and peculiarities of specific cases. We will now increase the number of clusters K and describe the results of a model with more clusters.

7.2.2. Cluster Model Training

In this section we present the results of the training of the K-cluster model for $K = 4$ and for $K = 10$. Figure 7.4 shows the value of the parameters Θ_k for the two cases. We only considered the four parameters that have the highest importance: r, m, f, and c.

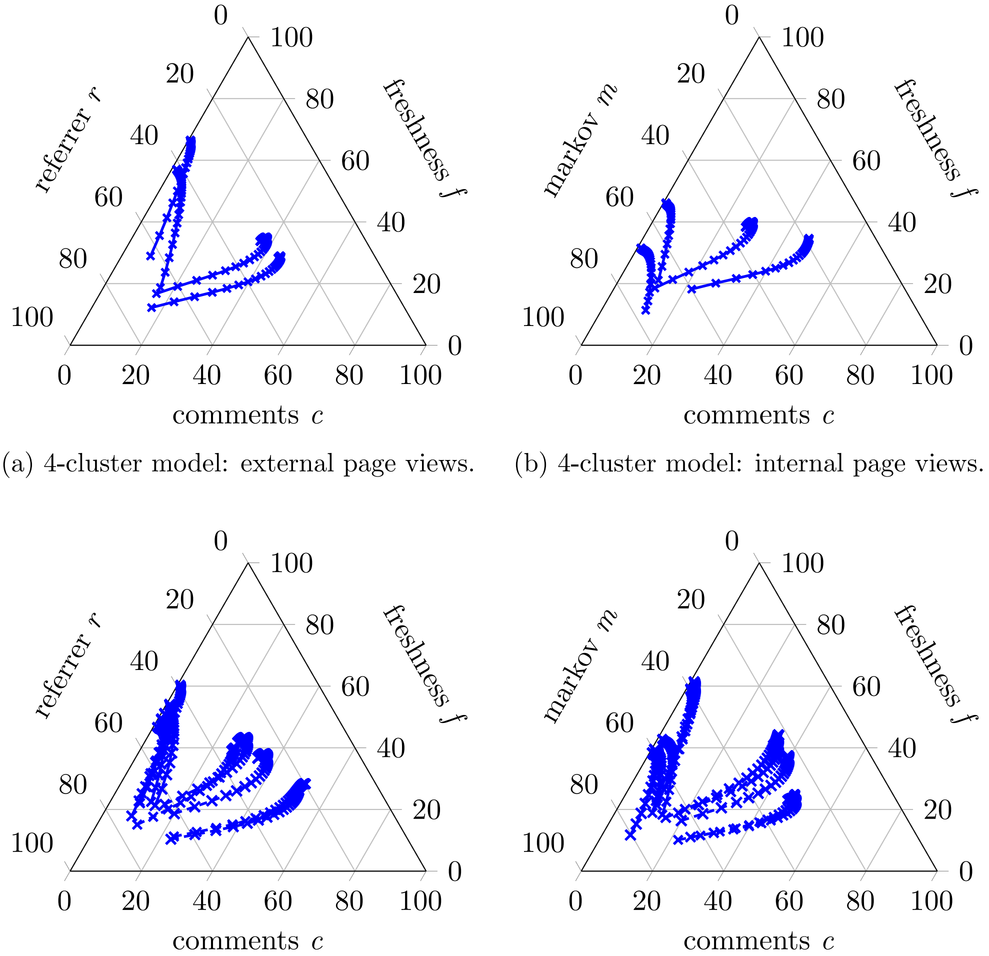
We can see that the characteristics of the 1-cluster model appear in the multi-cluster ones. For example, the importance of the previous action is higher on average for internal actions. Indeed, r in Figure 7.4a has lower values than m in Figure 7.4b.

We can see that the behavior is independent from the number of clusters. Increasing the number of clusters to 10, we can see that they more or less converge to the same areas. The first area is around $p(X \mid c) = 0$ (left edge of the triangle), where the contributions come mainly from the referrer and the freshness. The second area is a mixture of the three components (in the center of the triangle).

7.2.3. Contribution of the Features

From the complete 1-cluster model presented in Section 7.2.1, which we call $\mathcal{M}_1$, we derive simplified models, considering only a subset of information about the actions:

- $\mathcal{M}_{r,m}$: only considers the information about the external context (referrer domain r and Markov chain m);



(a) 4-cluster model: external page views.

(b) 4-cluster model: internal page views.

(c) 10-cluster model: external page views.

(d) 10-cluster model: internal page views.

Figure 7.4: Value of the parameters of the cluster model for $K = 4$ and $K = 10$, at various iterations of the EM algorithm. We use ternary plots to represent the contribution of the most significant features: f, r, m, and c. Each line represents the evolution of the parameters of a cluster at various iterations of the learning (starting close to the bottom left corner).

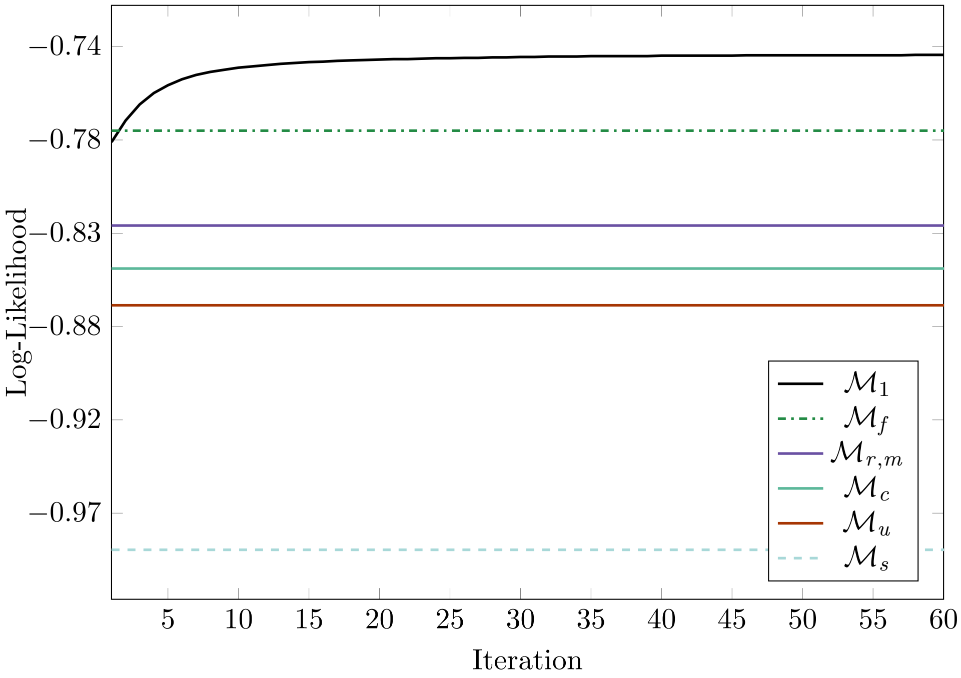


Figure 7.5: Log-likelihoods of the baseline models.

- $\mathcal{M}_u$: only considers the user u;
- $\mathcal{M}_f$: only considers the freshness of the articles f;
- $\mathcal{M}_c$: only considers the contribution of the comments c;
- $\mathcal{M}_s$: only considers the contribution of the shares s.

Figure 7.5 shows the performance of the baselines in terms of log-likelihood in the test set. Since the full model $\mathcal{M}_1$ increases the log-likelihood by learning the parameters, we plot the evolution of its log-likelihood depending on the iterations of the algorithm.

We can see that the best performing baseline, $\mathcal{M}_f$, outperforms the 1-cluster model at the first iteration, but it is rapidly reached by it in the next iterations. The second best model is $\mathcal{M}_{r,m}$, as one could expect from Section 7.2.1. The worst-performing model is $\mathcal{M}_s$, while comments c perform better.

7.3. Discussion

This chapter presented a model for user browsing. The model combines the contribution of an heterogeneous set of features. The features capture aspects such as the freshness of news articles, the context of the session, the user, and the social signal. We formalized the generative model and described the results of the learning. According to the learned model, the best features are the freshness of news articles, the referrer domain, the Markov chain, and the time since last comment.

The next chapter will build on the results of this one. We will describe a more complex model of user browsing that is focused on the features that gave the greatest contribution, namely the context of the session, the referrer URL, and the Markov chain.

Chapter **8**

Clustering Based on User Browsing Sessions

Modeling user behavior has become critical on the web, but particularly for large-scale websites that openly offer content or services without requiring user registration. Such websites often rely on repeated user visits, so their success depends highly on how well they are able to anticipate a user's information needs by providing the right content, at the right time, in the right places. Yet, it is not unusual for the "owners" and editors of these sites to rely on simple click-through rate heuristics to make important decisions that clearly impact on whether visitors will return to the site or not. In the particular case of news, this includes deciding the different layouts of news sections (*e.g.*, should the business section display a link to a technology article on the *top part* of an article page or on the *right or left* panel?), the links to include (*e.g.*, should the sports section have a link to *entertainment*?), and the type of content to promote.

Such decisions, however, are often complex because all the variables that determine the look and feel of a page and the content provided, must also take into account user behavioral patterns, which often depend on context. News consumption patterns differ depending on how the user arrives at the site, whether by clicking on links shared through social media, e-mail, or through comments on the news sites themselves (see [120]). In addition, users search for news, subscribe to RSS feeds, and visit news pages directly. Added to this is the fact that users don't consume news the same way at different times of the day or different days of the week. Given this complexity, there are important needs for news content providers in at least two areas: (1) gaining insights into how users behave when they visit the site depending on the context; and (2) using models that can be leveraged to predict behavior and automatically link content or set layout parameters.

In this chapter, we address these two areas. In particular, we present a probabilistic framework for session modeling that creates clusters of similar sessions, and uses contextual session information (time, referrer domain, link locations, page categories) to probabilistically assign a session to multiple clusters. We use a generative probabilistic model whose core is formed by a Markov process to capture the sequential nature of augmented sessions, and which naturally extends to a clustering of the data that can be computed by means of a nested Expectation Maximization algorithm. Moreover, the fully probabilistic nature of our approach allows us to turn the model into a predictor by marginalizing out latent variables and conditioning on the desired input observables. Exploiting the flexibility of the inference machinery allows us for instance to compute predictions for the next category, for the location of the next click, or for identifying keywords in link texts given a category, respectively. Visualizing the posterior estimates of the respective parameters provides insights on where to place links and which words to use for the anchor texts.

Our main technical contribution is the extension of Markov process-based clustering models to dynamically include context. We develop a nested mixture model for distributions over session timestamps that is able to capture periodic behavior and derive a nested EM-algorithm that *simultaneously* infers the mixture weights of the time distribution and the cluster parameters for the distributions over categories and other context. Our framework does not limit the type or number of context variables, but we validate our approach by using timestamps, referrers, and click metadata as contextual variables.

We empirically evaluate our approach using the `YahooNewsBrowsing-UK` dataset (see Section 3.3.3) and observe that the session-based clustering model outperforms usage-based and personalized models by a large margin. We provide exemplary interpretations of the produced clusters along various dimensions and discuss their impact on user understanding.

The results of this chapter were published in [68].

8.1. The Clustering Model

In this section we will describe the contextual generative model used for clustering browsing sessions.

In Section 3.3.3 we formalized a session x of length M as a 5-tuple $x = (t, r, \vec{v}, \vec{s}, \vec{w})$, where t is the timestamp of the session, r is the referrer domain, $\vec{v} = v_1, \dots, v_M$ and $\vec{s} = s_1, \dots, s_{M-1}$ are sequences of page view categories and click locations, and $\vec{w} = w_1, \dots, w_{M-1}$ are the clicked anchor texts in bag-of-words representation, respectively.

The basic idea behind our model is as follows. In the first step, a cluster k is drawn according to a multinomial distribution parameterized by π. Then the session is drawn according to the parameters θ_k of the selected cluster by drawing timestamp t, referrer r, and the first page view v_1 and using the Markov process to generate subsequent clicks with page views v_j, locations s_j, and word distributions w_j until the *exit* state which terminates the generation process is reached. The probability of a session x can be factorized as follows:

$$P(x|\theta_k) = P(t|\beta_k)P(r|\rho_k)P(\vec{v}|r, \tau_k)P(\vec{s}|\vec{v}, \sigma_k)P(\vec{w}|\mu_k),$$

where $\theta_k = \{\beta_k, \rho_k, \tau_k, \sigma_k, \mu_k\}$ denotes the set of parameters of the k-th component so that

$$P(x|\Theta) = \sum_{k=1}^{K} \pi_k P(x|\theta_k)$$

with $\Theta = \{(\theta_k, \pi_k)\}_{k=1}^{K}$ denotes the complete generative model.

Figure 8.1 shows plate models visualizing the generative process. Observed variables are shaded, while unshaded nodes correspond to model parameters; arrows denote dependencies and boxes indicate repetitive draws. The node labeled $\vec{e}$ denotes the sequence of events, whose generating model is detailed in the bottom of the figure.The remainder of this section explains the model as well as the inference and parameter optimization processes in greater detail.

8.1.1. Timestamp

The distribution for the timestamps $P(t|\beta)$ is designed to capture regular behavior across days of the week: a week is modeled as a mixture model

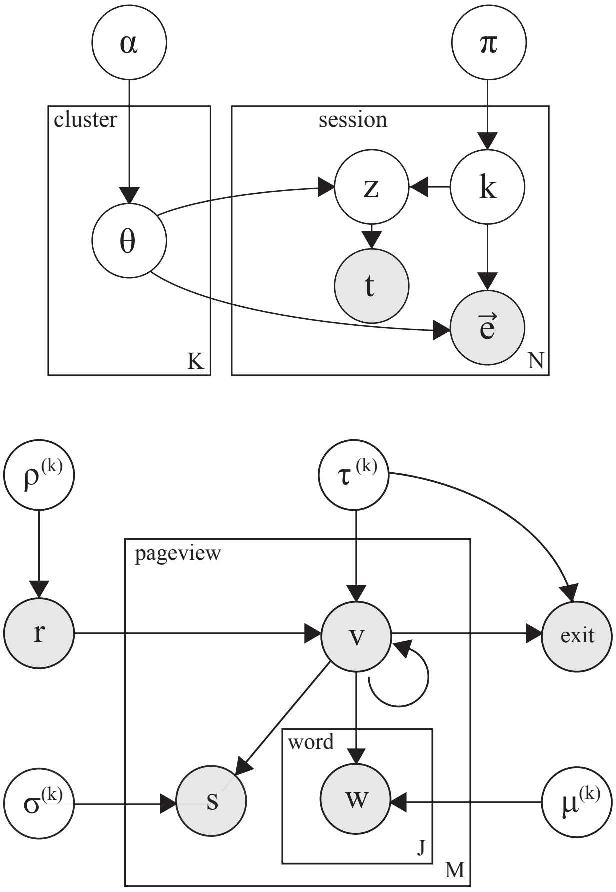


Figure 8.1: Graphical model for the generative process. Shaded nodes encode observables, and unshaded nodes model parameters. The navigation sequence is subsumed in node $\vec{e}$ in the left hand diagram, and detailed in the right hand diagram.

of periodic Gauss-like distributions whose peaks are repeated in one week intervals. In addition to these weekly repeating patterns, we capture regularities within workdays by including components whose peaks repeat from Monday to Friday. Note that repeating components alone does not favor periodic patterns since a repeating component is only a mixture of non-repeating components and does not change the space of overall mixture distributions. We therefore introduce a bias towards periodic and smooth distributions by interpolating the components with a uniform distribution to various amounts. Smoother and more periodic components are interpolated less than peaked components.

In general, mixtures using an infinite number of components do not scale well. Thus, we restrict our mixture to only a finite number of components that can be estimated efficiently; that is, instead of introducing components centered at every possible point within a week, we use components spaced in 10 and 30 minute intervals, respectively. We end up with 1,536 components organized in four groups:

- The first group consists of 48 working-day periodic components spaced 30 minutes apart, with a standard deviation of four hours;
- The 144 components in the second group are also periodic over the working days; their time lag is 10 minutes, and their standard deviation is one hour;
- The components of the third group are non-periodic (apart from repeating weekly) to capture patterns that differ between days of the week. We deploy 336 density functions centered in 30 minute intervals with a standard deviation of four hours;
- The fourth group contains 1,008 non-periodic components spaced in 10 minute intervals with a standard deviation of one hour.

Each element in these groups is referred to as a mixture component g_j. For every cluster k, the influence of each component is parameterized by a 1536-dimensional vector β_k with $\sum_{j=1}^{1536} \beta_{k,j} = 1$. Every session has a latent indicator variable z that selects one of the mixture components, such that the overall distribution over timestamps can be written as $P(t|\beta) = \sum_j P(z = j|\beta)P(t|g_j)$. Figure 8.2 shows an exemplary time distribution for a solution with three clusters together with the actual observed distribution in the training set.The described mixture model pro-

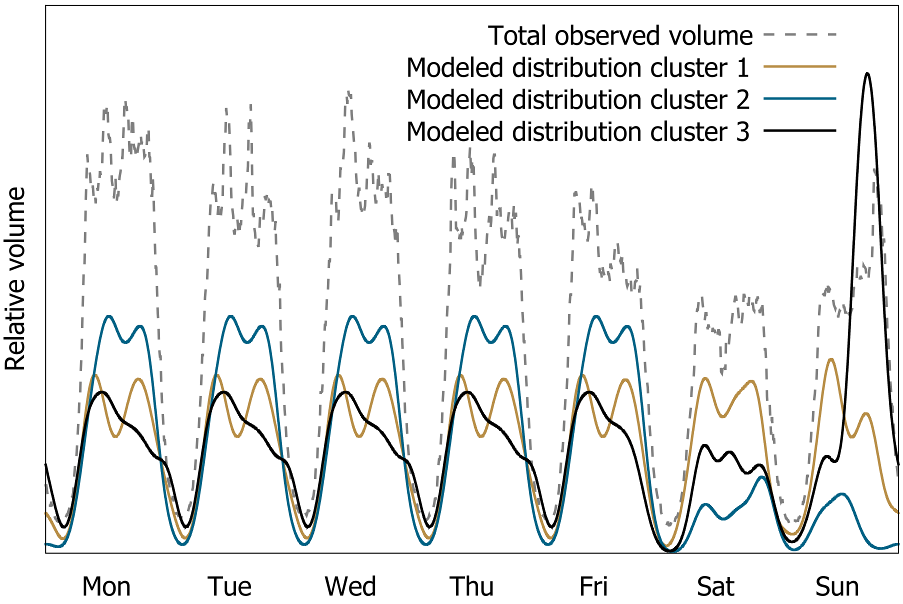


Figure 8.2: Observed and modeled time distributions.

duces smooth and periodic distributions without overfitting the data, *i.e.*, without reproducing the noise of the actual observed distribution.

8.1.2. Referrer Domain and Page views

As shown in Figure 8.1, the referrer domain r and the page views $v_1, \ldots, v_M$ form a Markov chain together with a distinguished *exit*-symbol. We use a first-order Markov assumption, which reflects the intuition that clicks only depend on the viewed page and are thus independent of previous page views and/or clicks. The resulting Markov process consists of two components, a multinomial distribution parameterized by a vector ρ over the set of all referrer domains $P(r|\rho)$ and transition probability parameters τ for the sequence of page views $P(\vec{v}|r, \tau)$. The latter decomposes into the matrices $\tau = \{\tau^0, \tau^+\}$ where τ^0 specifies the distribution of the topic of the first page view given the referrer, and τ^+ specifies the probability of transitioning between the topic v_m and topic v_{m+1} or the end of the session, respectively. Hence, the probability of the Markov chain is given by $P(r, \vec{v}|\rho, \tau) = P(r|\rho)P(v_1, \ldots, v_n|r, \tau)$, where $P(r|\rho) = \rho_r$ and matrices τ^0

and τ^+ such that

$$\begin{aligned} P(\vec{v}|r,\tau) &= P(v_1|r,\tau^0)\left[\prod_{m=1}^{M-1} P(v_{m+1}|v_m,\tau^+)\right] P(exit|v_M,\tau^+) \\ &= \tau^0_{r,v_1}\left[\prod_{m=1}^{M-1} \tau^+_{v_{m-1},v_m}\right] \tau^+_{v_M,exit}. \end{aligned}$$

8.1.3. Anchor Texts and Location of Clicks

The distribution of the anchor texts of the clicked links $P(\vec{w}|\mu)$ could give insights into the static information needs of the users. The words of the anchor texts are drawn from multinomial distributions over a dictionary with cluster-specific vector parameter μ. Similarly, the location of the clicked links is also modeled by a multinomial distribution $P(\vec{s}|\vec{v},\sigma)$ which is however conditioned on the category of the following page view. The latter multinomial is governed by matrix parameter σ. Using the independence of link text and location leads to $P(\vec{w},\vec{s}|\vec{v},\sigma,\mu) = P(\vec{w}|\mu)P(\vec{s}|\vec{v},\sigma)$ with

$$P(\vec{w}|\mu) = \prod_{m=1}^{M-1} P(w_m|\mu) = \prod_{m=1}^{M-1}\prod_{i=1}^{|w_m|} \mu_{w_{m,i}}$$

where $|w_m|$ denotes the number of anchor text words of the m-th clicked link, and

$$P(\vec{s}|\vec{v},\sigma) = \prod_{m=1}^{M-1} P(s_m|\sigma,v_{m+1}) = \prod_{m=1}^{M-1} \sigma_{v_{m+1},s_m}.$$

8.2. Parameter Estimation

Given a set of N sessions $X = \{x_1,\ldots,x_N\}$ and a number of clusters K, the task is to estimate the parameters $\Theta = \{(\theta_k,\pi_k)\}_{k=1}^K$ of the generative model. We aim at finding the *maximum-a-posteriori* (MAP) solution by solving

$$\operatorname*{argmax}_{\Theta} P(\Theta|X) = \operatorname*{argmax}_{\Theta} P(\Theta)\prod_{i=1}^{N}\sum_{k=1}^{K} \pi_k P(x_i|\theta_k),$$

where $P(\Theta)$ is modeled by a symmetric Dirichlet prior with concentration factor α.

The main difficulty in the optimization is the presence of two different types of latent variables, the first is encoding the cluster memberships of the sessions k and the second encodes the distribution over time components z that generate the timestamp. Since the latter is required for inferring the former, we now present a nested Expectation Maximization strategy to optimize both simultaneously.

Let us assume for a moment that the time component indicator variables z_i were known. In that case we could use a standard Expectation-Maximization (EM) clustering algorithm [44] for the parameter estimation. The EM algorithm computes, in every E-step, estimates $\gamma_{i,k}$ of the cluster membership variables, with $\sum_k \gamma_{i,k} = 1$, which indicate the posterior probabilities of an example x_i belonging to cluster k. In the M-step, the MAP-estimates for every set of the cluster parameters θ_k are computed as follows:

$$\hat{\theta}_k = \arg\max_{\theta_k} P(\theta_k|X, y) = \arg\max_{\theta_k} \log P(\theta_k) + \sum_i \gamma_{i,k} \log P(x_i|\theta_k).$$

Due to the conjugacy of the Dirichlet prior to the multinomial distribution, the maximization simplifies to counting the occurrences of a particular component transition. For example the time distribution component weights β_k are computed as

$$\hat{\beta}_{k,\ell} = \frac{\alpha - 1 + \sum_i \gamma_{i,k} [\![z_i = \ell]\!]}{\sum_{\ell'} \alpha - 1 + \sum_i \gamma_{i,k} [\![z_i = \ell']\!]},$$

with the indicator function $[\![z_i = \ell]\!] = 1$ if $z_i = \ell$ is true or 0 otherwise. All other parameters are calculated analogously.

However, since the z_i are actually unknown, we have to marginalize over them. Thus the optimal parameter vector β for a cluster k, given the current cluster membership estimates γ, is optimized using

$$\hat{\beta} = \arg\max_{\beta} (\alpha - 1) \sum_j \log \beta_j + \sum_i \log \left[\gamma_{i,k} \sum_{\ell=1}^{1536} \beta_\ell P(t_i|g_\ell) \right],$$

under the constraint $\sum_j \beta_j = 1$ where g_ℓ denotes the generating components of the timestamp. This is a concave optimization problem under the condition that $\alpha \geq 1$, since the terms $P(t_i|g_\ell)$ are constant. Having no closed-form solution, a straight-forward approach would be to solve it using gradient descent or a variant of Newton's method. However the estimates γ change in every iteration of the EM-algorithm, and thus a costly optimization would have to be performed in every iteration.

A more efficient method is to *intertwine* the optimization of β with the EM-algorithm, performing only one closed-form update of β in every M-step. We derive this update analogously to the M-step update for the cluster prior π (*cf.* [18]), by introducing additional variables $\zeta_{k,i,\ell}$, with $\sum_l \zeta_{k,i,\ell} = 1$, which encode our posterior belief that the timestamp of session x_i is generated by component ℓ, conditioned on x_i belonging to cluster k. These can be computed in the E-step as

$$\zeta_{k,i,\ell} = \frac{\beta_{k,\ell} P(t_i|g_\ell)}{\sum_{\ell'} \beta_{k,\ell'} P(t_i|g_{\ell'})}. \tag{8.1}$$

Using these estimates, we can compute the component weights of each cluster in the M-step as

$$\hat{\beta}_{k,\ell} = \frac{\alpha - 1 + \sum_i \gamma_{i,k} \zeta_{k,i,\ell}}{\sum_{\ell'} \alpha - 1 + \sum_i \gamma_{i,k} \zeta_{k,i,\ell'}}.$$

This *nested* EM-algorithm is guaranteed to increase the data likelihood in every iteration until convergence to a local optimum, analogously to the standard EM-algorithm.

8.3. Inference

Our generative model $P(x|\Theta)$ can be easily turned into a prediction model by marginalizing out latent variables and conditioning on the desired input observables. Recall that, at the m-th page view of a session, we already observed the previously visited categories $v_1, \ldots, v_m$ and the previously clicked locations $s_1, \ldots, s_{m-1}$ and link texts $w_1, \ldots, w_{m-1}$, as well as the session's timestamp t and referrer domain r.

For instance, we can predict the category of the next page view a user will navigate to by conditioning on the context and history while marginalizing over the latent cluster variable. Conditioning again on this prediction, we can furthermore predict which location within the page she will click on next. Let $\vec{e}_{[m]}$ denote the events of a session up to the m-th page view, that is $\vec{e}_{[m]} = \{(v_1, \ldots, v_m), (s_1, \ldots, s_{m-1}), (w_1, \ldots, w_{m-1})\}$, then the predictive distribution for the next category (including the end of the session) is given by $P(v_{m+1}|\vec{e}_{[m]}, t, r)$ and can be computed by marginalizing over the cluster variables,

$$P(v_{m+1}|\vec{e}_{[m]}, t, r) \propto \sum_k P(v_{m+1}|v_m, \theta_k) P(\vec{e}_{[m]}, t, r|\theta_k) P(k). \tag{8.2}$$

However, our model also contains traditional models as special cases that are solely based on the observed sequence of categories [23] by an additional marginalization over the context variables,

$$\begin{aligned} P(v_{m+1}|v_1, \dots, v_m) &\propto \sum_k \sum_{\vec{s}, \vec{w}, t, r} P(v_{m+1}, k|\vec{e}_{[m]}, t, r) \\ &\propto \sum_k P(v_{m+1}|v_m, \theta_k) P(v_1, \dots, v_m|\theta_k) P(k). \end{aligned} \tag{8.3}$$

The comparison of Equations (8.2) and (8.3) shows that the context variables provide additional information on how to weight the influences of the different clusters. In the following section, we evaluate the context variables in terms of their contribution to the predictive performance.

Our model can contribute to optimize the layout of web pages by providing insights on where to place links and likely-clicked word distributions. We therefore infer the location of the next click by conditioning on the linked category

$$P(s_m|v_{m+1}, \vec{e}_{[m]}, t, r) \propto \sum_k P(s_m|v_{m+1}, \theta^{(k)}) P(\vec{e}_{[m]}, t, r, v_{m+1}|\theta^{(k)}) P(k).$$

The predictive distribution for clicking on a link with anchor text w, $P(w|\vec{e}_{[m]}, t, r)$, can be computed similarly and is proportional to $\sum_k P(w|\theta^{(k)}) P(\vec{e}_{[m]}, t, r|\theta^{(k)}) P(k)$.

8.4. Incremental and Distributed Parameter Estimation

For practical applications, the batch style of the nested EM-algorithm hinders deployment because every retraining needs to be performed on all data. In this section, we briefly sketch the parameter estimation in real-time using incremental updates, similar to the algorithm proposed by Neal and Hinton [112].

Once the clusters are determined by running the nested EM-algorithm until convergence, new sessions can be incorporated by performing a single partial iteration. For every new session, we have to compute the estimates γ, then update the counters of all components and normalize the cluster parameters using $\mathcal{O}(K)$ operations. Let $count_T(\cdot)$ denote the counts of all weighted entity occurrences after having processed T examples, *e.g.*, $count_T(\rho^{(k)}, \ell) = \alpha - 1 + \sum_{i=1}^{T} \gamma_{i,k} [\![r_i = \ell]\!]$ and $count_T(\rho^{(k)}) = \sum_\ell count_T(\rho^{(k)}, \ell)$. Then a new example x_* can be incorporated into the model by first estimating its cluster membership using the current model parameters $\tilde{\Theta}, \tilde{\pi}$ as

$$\gamma_{*,k} = \frac{\tilde{\pi}_k P(x_* | \tilde{\theta}^{(k)})}{\sum_{k'} \tilde{\pi}_k P(x_* | \tilde{\theta}^{(k')})}.$$

The counts are updated according to $count_{T+1}(\rho^{(k)}, \ell) = count_T(\rho^{(k)}, \ell) + \gamma_{*,k} [\![r_* = \ell]\!]$ and $count_{T+1}(\rho^{(k)}) = count_T(\rho^{(k)}) + \gamma_{*,k}$. The new MAP-parameters are

$$\hat{\rho}_\ell^{(k)} = \frac{count_{T+1}(\rho^{(k)}, \ell)}{count_{T+1}(\rho^{(k)})}.$$

The remaining parameters, π, β, τ, σ, and μ, are updated analogously.

That way, an up-to-date, approximate model can be maintained efficiently and full retraining is only necessary occasionally. The benefit of an online variant is that novel topics can be taken into account and recommended to users faster. Our model already has an advantage over user-centric, personalized models, because every user benefits from the information gained about sessions in the cluster she is currently in. Having an always up-to-date model entails that estimates for the click probability of a new topic are available as soon as a few peers have clicked on it.

Furthermore, the training of our model can easily be distributed on several machines using the MapReduce framework. EM-like algorithms process training instances one after another and store tables with counts for every instance in the E-step. The counting can be performed on several machines in parallel during the map-phase, independently for every training example, generating cluster membership estimates and fractional counters. In the reduce phase, the fractional counters are aggregated, and finally the M-step is performed, namely computing the MAP-parameters from the total counts. After the M-step, the current model is distributed to all machines for the next iteration of mapping and reducing respectively expectation and maximization (*cf.* [41]). The distributed computation schema can in principle also be applied to the online variant, for processing multiple new examples in parallel.

8.5. Empirical Evaluation

In this section we evaluate the generative model under several aspects using the `YahooNewsBrowsing-UK` dataset (see Section 3.3.3).

The next section reports on the predictive performance of the probabilistic model and compares the outcomes with appropriate baseline methods. Section 8.5.2 addresses insights gained by applying our model to the news domain and discusses the findings in terms of user understanding.

8.5.1. Predictive Performance

We measure predictive performance in terms of the predicted log-likelihood of the next page view and the location of the next click conditioned on the session's history and context. That is, we average $\log P(s_m, v_{m+1}|\vec{e}_{[m]}, t, r)$ over all events of all test sessions. The higher the session log-likelihood of a model, the better it reflects the characteristics of the data. This is a more natural evaluation measure than for instance measuring the accuracy of the most probable page view and location, (*e.g.*, $\arg\max_{s_m, v_{m+1}} P(s_m, v_{m+1}|\vec{e}_{[m]}, t, r)$), since there are no negative examples. Note that if a user clicks on a link ℓ, it does not mean that she is not interested in other articles but that she is at that point *more* interested in ℓ.

Our evaluation comprises several aspects of the probabilistic model. We first introduce the baseline methods. We then compare the accuracy of the next click with appropriate baseline methods, and finally evaluate the impact of the context by marginalizing over the respective variables.

Baselines. We compare our model to two user-centric baselines. Instead of using the nested EM-algorithm, the two baselines use a fixed assignment of user sessions to clusters. They are formally defined as follows.

The *usage-based* baseline simply groups the users into three groups according to their number of page views in June 2011. We define the group sizes so that they reflect heuristics used in commercial systems to provide a basic level of personalization and/or monetization. The first group contains *tourists* who rarely visit the site, the second group covers *regular users*, and the third group contains the *power users*. We estimate a probabilistic model for every group.

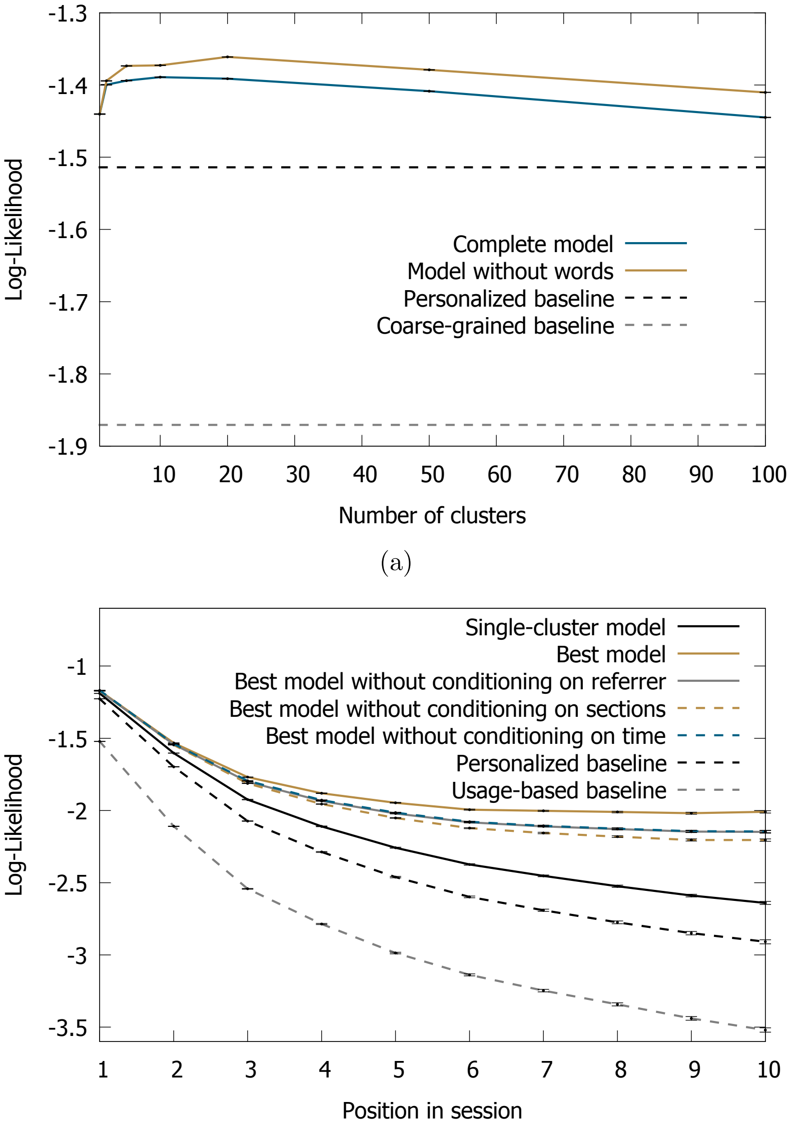
The *personalized* baseline reflects a personalized approach and estimates a single probabilistic model for every user, by assigning her respective sessions in June 2011 to a cluster. However, initial experiments showed that the data was too sparse for users with only a few page views. We thus split users according to their usage in two groups by using a threshold η. For users whose page views exceed η in June 2011, a personalized model is estimated as described, while users who generate fewer page views than η are grouped into a single cluster. If users cannot be disambiguated and uniquely assigned to a cluster in the evaluation data from July we also resort to the model that is estimated on the shared cluster. The trade-off η is adjusted by model selection where the chronologically last 25% of June 2011 are used as holdout data.

Predicting Categories Using Context. We evaluate the performance of predicting the next clicked category and link location, conditioned on the session history, using the model in Equation 8.2. We compare the baselines with the full generative model of Section 8.1 and a model where we omitted the words of the anchor texts. Preliminary experiments have shown that the latter improves over the full model. By contrast, marginalizing over the other context variables reduces the performance of the full model. We refer to the next section for detailed analysis of the impact of the different context variables.

Figure 8.3a shows the average log-likelihood of the prediction for different numbers of clusters. The two baselines use a fixed clustering and are therefore independent of the number of clusters. The full probabilistic model and its counterpart without the anchor texts outperform the baselines significantly, even for only a few clusters. Additionally, the models without words consistently outperform the full model, indicating that the distribution over the bag-of-words is too noisy to contribute positively.

The predictive performance initially increases with the number of clusters and then decreases again for more than 20 clusters; generally, solutions with too many clusters tend to overfit the data. In the remaining experiments we therefore focus on models with 20 clusters and always marginalize out the anchor texts of the clicked links.

Evaluating the Impact of Context. We now evaluate the importance of the incorporated context. We begin with the best model obtained that consists of 20 clusters and does not depend on the anchor texts of the links. Using this model, we selectively discard parts of the remaining context,



(a)

(b)

Figure 8.3: Prediction performance and standard error depending on number of clusters (a) and length of session history (b).

that is the referrer, the timestamps, and the locations of the clicks, to measure their respective impact. For comparison, we include the *usage-based* and *personalized* baselines and an additional single-cluster solution that has only a single generating component and does not take context into account. Additionally, we break down prediction accuracy by the position of the clicked category and link within the session, in order to gain insight into how accumulating various amounts of context information impacts accuracy.

Figure 8.3b shows the resulting prediction accuracies for the baselines, the best model, and various sub-models thereof. Accuracies clearly drop as sessions progress. Except for the usage-based baseline, all methods predict the first click after the first page view equally well. As the number of page views increases, the performance of the approaches becomes more distinguishable. A possible explanation for the performance drop is that users are presented a variety of related news articles and may be distracted by interesting news articles while browsing the site, making prediction more difficult as a session progresses.

Interestingly, the single-cluster solution performs significantly better than the other two baselines. Apparently, the fixed clusterings are inappropriate approaches to the data and thus lead to poorer performance. The contextual models always perform significantly better than the baselines, which do not take advantage of context. More importantly, instead of deteriorating as the baselines, the context saturates the performance of the contextual models, which remain constant for sessions with more than 5 page views. Discarding context significantly drops the performance; the differences in performance clearly show the importance of the different types of context.

8.5.2. Applications of Our Model

In this section we discuss the suitability of the cluster mapping, visualization, and how our model may be applied to modifying page layouts.

Mapping Users to Clusters. One of the key questions is how confident the cluster assignments obtained from the probabilistic model are. We measure confidence using the information theoretic measure *perplexity*. In our case, the maximum possible perplexity value for 20 clusters is 20, which indicates a uniform distribution over the 20 clusters, while a perplexity of 1 implies a point distribution for a single cluster.

Figure 8.4 shows the perplexity of the distribution over the 20 clusters, conditioned on different sets of variables. The leftmost point denotes the *a*

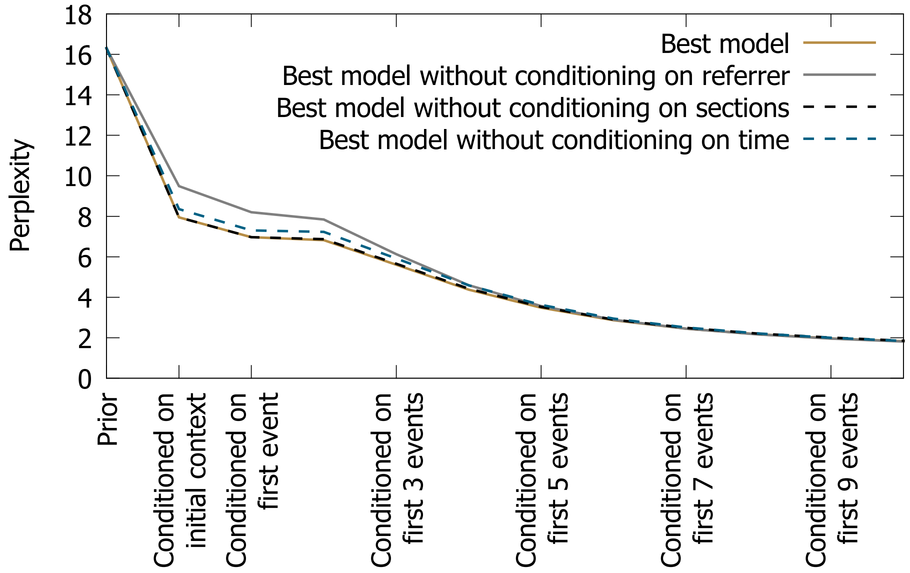


Figure 8.4: Perplexity of distribution over clusters.

priori perplexity that is solely based on the priors π_k of the clusters. The second point is the perplexity of the distribution conditioned on the initial context given by the referrer domain and the timestamp. The figure shows that context significantly reduces the uncertainty by about 50%. Every click of the user further reduces the perplexity which drops rapidly until it reaches 2 which corresponds to the same uncertainty as that of a coin flip.

The figure shows that we obtain significant reductions in perplexity and therefore higher confidence about the cluster membership by conditioning the model on context.

Time-based Visualizations. Previous work on clustering user sessions (*e.g.*, [23]), focuses on visualizing the resulting clusters only in terms of the sequences of visited categories. By contrast, one of the main advantages of using dynamic contextual models is that the resulting clusters can be interpreted along the context dimensions. The corresponding visualizations thus highlight context-specific aspects of the data and allow for meaningful projections. For instance, Figure 8.5 shows the observed category distribu-

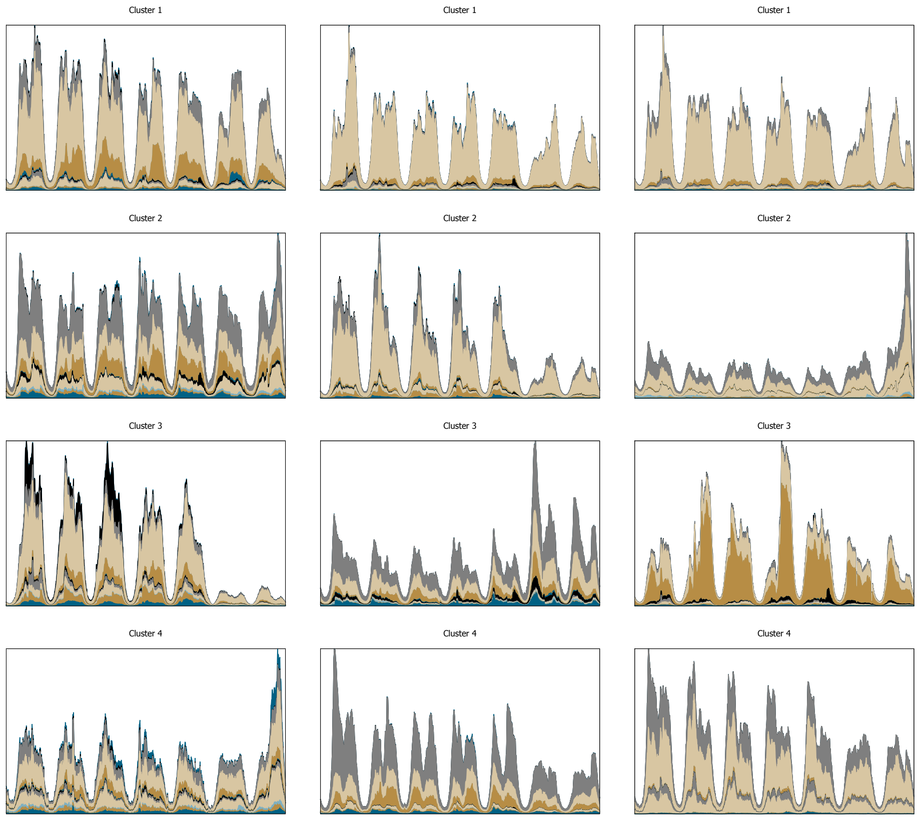


Figure 8.5: Distribution of categories over time in the largest 4 clusters for models with 4 (left column), 20 (middle column), and 50 clusters (right column).

tion for the four clusters with the highest prior probabilities, projected on the days of the week according to the timestamps of the contained sessions. The rows depict a model with four clusters (left column), the already discussed solution with 20 clusters (middle), and a large model with 50 clusters (right).

The figure shows strong correlations between the relative volume of the categories and time. Some clusters are specialized on reoccurring patterns for business days, while others focus on capturing weekends. The respective clusters also possess different topic distributions, indicating that one captures work-related browsing sessions while others cover more recreationally-

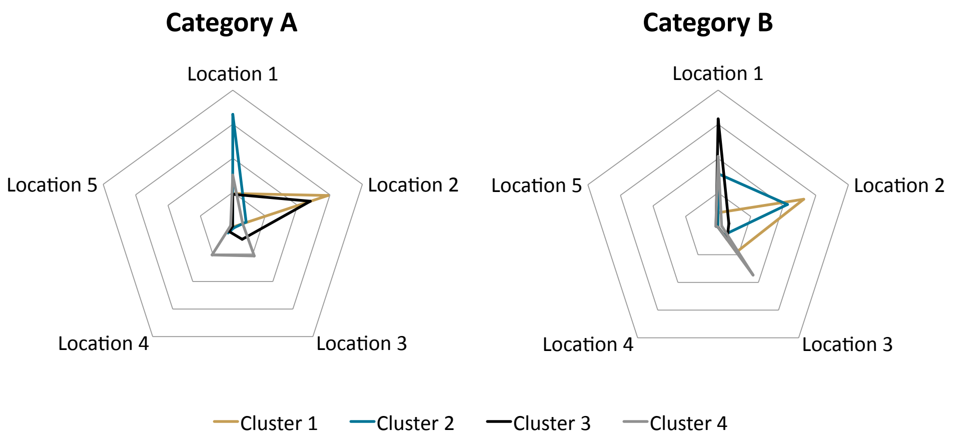


Figure 8.6: Distributions over five link locations for four clusters and two exemplary categories.

oriented information needs. Naturally, solutions with more clusters tend to cover fewer categories.

Compared to traditional approaches the contextualization leads to intuitive and interpretable results. Without context, the range of visualization possibilities is limited and more or less restricted to displaying transition matrices or cluster distributions.

Improving Web Page Layout and Content. Our model can be used to improve web page layout (*e.g.*, where to place "modules", sections, or links), and content (*e.g.*, words to use for link anchor texts). For example, Figure 8.6 shows the five most frequent locations for two categories A and B. The colored lines correspond to the four clusters and show the probability of a click on one of the locations given the category.

The visualization shows that some locations, such as 4 and 5, play only a minor role in the layout and are rarely clicked on. By contrast, locations 1 and 2 receive a high number of clicks and exhibit interesting behavior. Sessions in the blue cluster interested in category A mainly use location 1, while members of the black cluster focus on location 2 for performing the same action. Vice versa, location 2 is preferred by the blue cluster for category B, while the black cluster "prefers" location 1. Once detected, this behavior can be exploited by cluster-dependent layouts of the page to guide the user through the site, and link locations that are ignored by groups of users could be dynamically replaced by more appropriate pointers.

8.6. Discussion

The dynamic nature of user behavior in news consumption along with the complexities of the news cycle makes modeling and prediction extremely difficult. Our framework is able to consider context dynamically, and can be applied for prediction, as well as to obtain insights that could be used to make decisions on content and layout. On one hand, the interpretability of the clusters can provide significant insights (*e.g.*, a content provider examining Figure 8.5 could easily determine the most suitable content categories for weekdays vs. weekends), and on the other hand, its prediction capabilities could be used to automatically adjust content locations and links.

We presented a generative model for user navigation on the Web. Our approach models sessions as sequences of contextualized page views where context is incorporated in terms of timestamps, click metadata, and referrer domains. The model naturally leads to a clustering of the sessions that can be projected on context variables for interpretable visualizations. We empirically showed, on a large sample from Yahoo News, that our probabilistic approach is more accurate than baseline models. We exploited several features and discussed applications of our model in adjusting content locations and links.

The next chapter is an exploration of new models for browsing sessions. We try to go beyond Markov-based models and investigate models in which sessions are not linear but rather composed by parallel threads.

Chapter 9

Beyond Traditional Sessions

In this chapter we present two applications that go beyond the traditional session model. Until now, sessions have been modeled as sequences of actions or at most trees. This model is due to tabbed browsing in modern web browsers in which users open pages in a specific tab or spawn a new one from an already opened one.

The first application named *PRiSMA* (Section 9.1) is aimed at investigating the potential of *parallel browsing*, *i.e.*, the possibility of having multiple browsing threads opened at the same time.

The second application named *Metro* (Section 9.2) allows the user to smoothly move in the information space in order to explore people's relationships. Since the user is browsing through different dimensions at the same time (*i.e.*, people. photos, events), modeling a session by means of a list of pages becomes more limiting. A session becomes therefore a continuous path inside the information space.

The content of the next sections were published in [146, 32].

9.1. Searching Images in Parallel

We present *PRiSMA*, an image search application primarily designed for tablet devices, which allow users to explicitly perform multiple queries in parallel on large image collections. *PRiSMA* provides an intuitive and novel graphical user interface, which facilitates branching an initial query to simultaneously explore two or more result sets. As depicted in Figure 9.2, the results of each query are presented in an horizontal sliding strip. The inter-

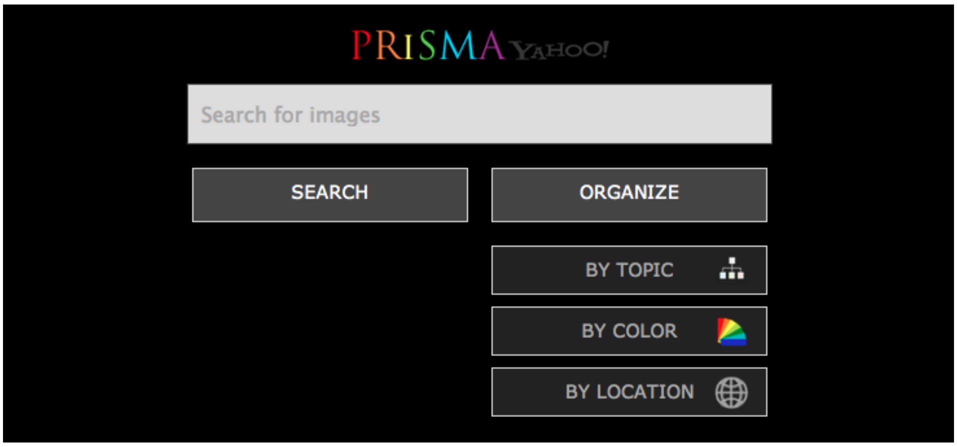


Figure 9.1: A screenshot of the search panel. Users can perform a simple search or branch the query by *topic*, *color* or *location*.

face allows users to easily create new strips, merge them, remove them and edit their associated query to modify the results of each query independently (see Figure 9.3). These functionalities, combined with traditional faceted search, allow users to automatically split the results by colors, geolocation or topic (*e.g.*, *Sports*, *Politics*, or *Nature*). Furthermore, *PRiSMA* also supports searching by image similarity, hence, users can create a new strip on the fly containing images, which are similar to a user-selected picture. Any such action can be done and undone without loosing the previous search stage. In this way, users are encouraged to explore the image collection in diverse and complementary ways with little effort.

By facilitating the exploration of multiple queries in an orderly fashion *PRiSMA* can help users to: *a*) have a better account of the result space; *b*) diversify the results; and *c*) conveniently broad or narrow the search space, first by exploring alternative search paths to latter focus only on those queries that provide better results. We envision three uses in which *PRiSMA* may be particularly useful for. In creative tasks, where users may require to explore diverse images within a given context while keeping a global picture of the results. As an educational tool, where users may use parallel search as a visual dictionary, where, with a single click, a term is shown in different contexts (*e.g.*, organizing the term “folk art” by location). Journalist and editors may benefit from the diversity in the result in order to identify potential stories. Thus, comparing images of concurrent events (*e.g.*, search for different images of riots in the different countries of the

Middle East) or comparing images from similar events in different periods of time.

PRiSMA may also be useful when users need to compare search results, for which, given no better option, users may place two windows side-by-side [153].

PRiSMA makes use of faceted search [165], image clustering using the image metadata [12] and it allows for searching images using similarity based on the images associated tags [121]. While each of these features is relevant for the implementation of *PRiSMA*, they are not essential to the application.

The main contributions of this section are:

- the proposal of a novel approach for image search browsing;
- the development of an image search application intended to facilitate and promote searching images in parallel;
- a user study with very positive results regarding the user's acceptance of this novel image search approach.

9.1.1. An Application for Parallel Browsing

For most common image searches, traditional search applications like Yahoo,[1] Bing,[2] or Google,[3] provide very good results. In many occasion however, users must modify their queries several times, not always with a satisfactory result [35]. With a greater or lesser success, several heuristics, techniques and interfaces have been proposed in order to improve these situations. To the best of our knowledge, all image search applications, whichever underlying technique they use, assume a single stream of exploration. Acknowledging that some image searches will always require, o benefit from, several modification of the initial query, we propose a novel approach by which users can simultaneously explore the result of multiple queries in a structured and orderly fashion.

PRiSMA is an image search application for tablet computers that facilitates and promotes the exploration of image collection from different perspectives. In other words, *PRiSMA* allows users to branch an initial query into two

[1] http://images.search.yahoo.com/

[2] http://www.bing.com/images/

[3] https://www.google.com/imghp/

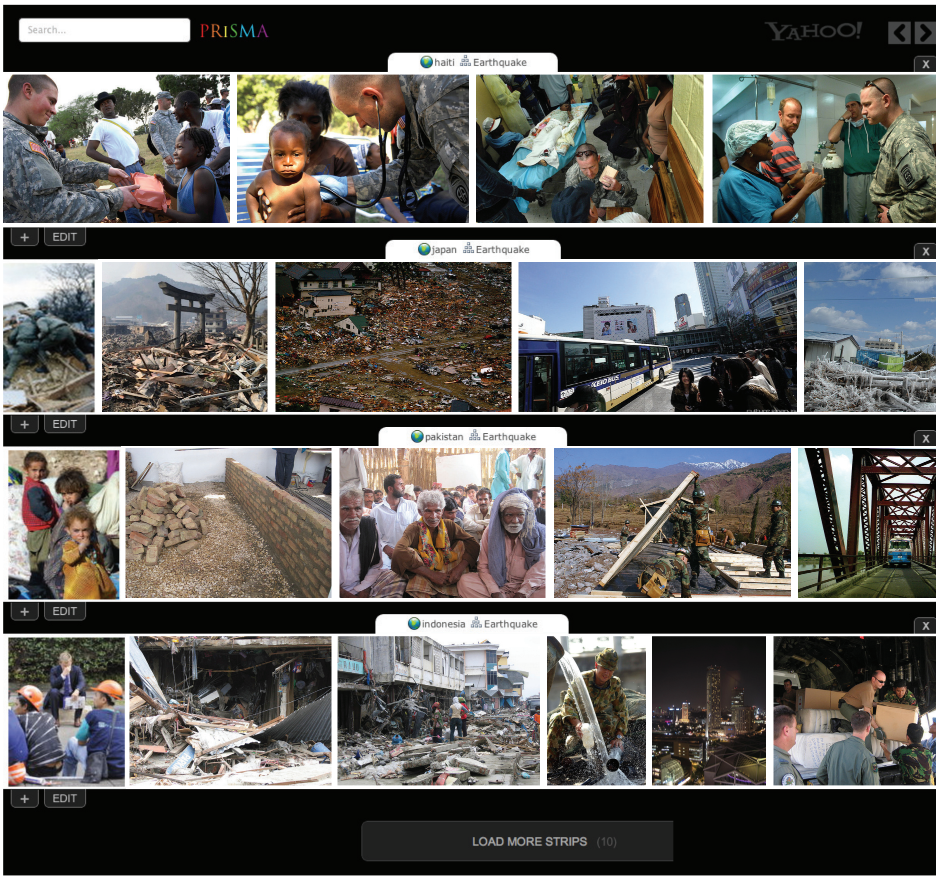


Figure 9.2: A screenshot for the results of the search for the keyword Earthquake organized by location. When automatically branching a query, the application opens up to 4 new strips. A button allows users to load more strips, or conclude the expansion.

or more queries and follow their results on the same screen. As depicted in Figure 9.2, *PRiSMA* places the results of each query in a horizontal strip. Users can browse the results in each strip by sliding their finger over the touch screen. Strips can be scrolled simultaneously or independently. This allows for two modes of exploration: a more general exploration, possibly more convenient on an early stage of the search; and a more detailed search, where users can focus on a single result set.

Strips can easily be created, reordered, edited (as depicted in Figure 9.3), removed and merged. This respectively allows users to initiate a new concurrent search, facilitates the results comparison, enables modifying the queries, discard non relevant search path and combine two or more queries.[4]

Currently *PRiSMA* allows searching the image collection by *tags*, *colors*, geographic *location*, and *topic*, where the topic of an image is given by a taxonomy of over 140 terms including categories like *Nature*, *People*, *Celebrities*, *etc.* To illustrate a simple use case of parallel image search, let us suppose a user, who is preparing some slides for a presentation, searches for the term "information overload", after some browsing, she may think that adding the keyword "funny" might render more interesting results. Now, rather than initiating a completely new search, the user can branch the query adding the term "funny" and continue browsing the two results in parallel.

As depicted in Figure 9.3, *colors*, *location*, and *topic* can be used for traditional faceted search on a single query. More interestingly, however, is that *PRiSMA* allows branching automatically any query into the multiple values of the selected facet type. Thus, for example, a user searching for images of "bus" may branch the query with a single click (see Figure 9.1), to obtain multiple strips organized by *topic*, *color* or *location.* Figure 9.2 depicts the result of branching the query "earthquake" by *location*, while Figure 9.3 shows the results of branching the query "tiger" by *topic.* This functionality is particularly useful in an initial stage of an exploration where users may want to diversify the results to help them clarify the scope of their search. An important side effect of diversity is the promotion of creative and serendipitous search. Thus returning to the "information overload" example, if the query is branched by *topic*, the user will obtain images of "information overload" in the context of *Technology* or *Business*, but also under less obvious topics such as *Cheerful*, *Travel* or *Food & Drinks*, which may help finding more original images.

It is worth noting however that while *PRiSMA* may promote diversity, it does so in an organized fashion. Strips with diverse but uninteresting images are easily removed. In other words, *PRiSMA* provides a principled way to broaden or narrow the search space, recognizing the benefits of expanding the exploration in an initial phase, but also the need to focus the search on a later stage.

In addition to the above-mentioned features, *PRiSMA* support searching images by similarity, based on the images' associated tags. Thus at any moment, by tapping on the bottom-left corner of an image, a new strip

[4] For the time being, this involves combining the two queries with an OR operator.

is created containing images similar to that selected by the user. In the current stage of development, this feature has shown to have an uneven performance. Because image similarity is explored as a secondary query (on a new strip), obtaining weak results is not critical, since the user can easily remove the created strip and continue exploring the image collection by other means.

While *PRiSMA* is mainly intended for tablet computers, it also runs on desktop computers. The application is fully implemented using HTML5, with Javascript and PHP. The image collection used by *PRiSMA* is the `GettyImages` dataset (see Section 3.3.9). Being stock images, most images contain rich and clean metadata. In particular, features like location and colors are already given by the images' tags. The search by image similarity is currently based only on the images' tags. Given the quality of the images metadata a straight query over the selected image tags renders fair results.

PRiSMA may be used for a variety of use cases, including simple entertainment (*e.g.*, searching for celebrities in different contexts or location) to more creative uses which may involve searching for conceptual terms, as in the case of "information overload", or searching more graphically oriented images, where organizing the search by colors may be of particular help. Suppose a user wants to explore images for the term "forest" for creative/inspirational purposes. In order to expand the search the user may select to view "forest" branched by location. The user may now have strips with images of forests from USA, Japan, UK, and Canada. The user may now decide to further branch by colors in Japan. With only three taps on the tablet computer, the user may obtain images for Japanese forests in the colors green, white, red, and pink.

Journalist and editors may also benefit from *PRiSMA*. By visualizing the results of different queries in the same screen, users can easily compare the results in order to construct a narrative. Searching, for instance, for concurrent events in different locations (as in Figure 9.2), may help users identify similarities and differences useful for building a story.

PRiSMA is in early stage of development and it firstly intended to help us explore the pros and cons of searching images in parallel from a user perspective. Our initial concern when initiating this exploration was related to the inherent complexity of searching in parallel multiple queries. In the following section we present our study intended to clarify this concern.

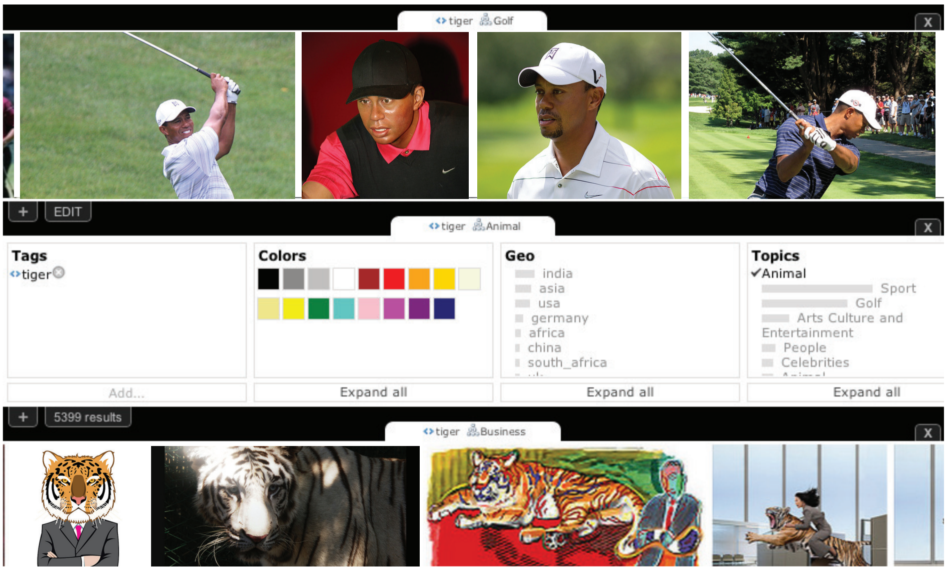


Figure 9.3: A screenshot of the Edit panel of a strip. Each strip can be edited individually. Users can add or remove tags, select one or many of the available facets or request to branch the query by all the values of the facets.

9.1.2. Evaluation

For the study we conducted individual interviews with eleven participants. We began each interview with a five minutes presentation of *PRiSMA*, both on a tablet and a desktop computer. We then gave turn to participants to ask questions and use the application with no specific instructions and for as long as the participants wanted. We concluded the interview with a questionnaire. The interview took between 20 and 30 minutes, depending on the time participants spent with the application and providing feedback. The participants included 10 computer scientists and a professional graphic designer. The age of the participants ranged between 25 and 40 years old, with a majority of male users and only one female. All users were familiar with image search and use it regularly: 3 of the participants searched for images on a daily basis, 4 on a weekly basis, and 4 users a few times per month.

For this study we were particularly interested in learning three basic things from the users: *a*) whether the inherent complexity of parallel search overshadows its benefits; *b*) what images would users search in this kind of environment; and *c*) to what extent users perceive this approach as both useful and novel.

Overall the results of this study were very good, in fact much better than expected. Unless otherwise specified, questions required participants to give a score between 0 and 10, being 10 the best option. From our study, users had a very clear understanding of the general principles of searching in parallel with *PRiSMA*, with an average score of 9. The users found the overall interaction very intuitive (with an average score of 7.5) and found no particular difficulty in navigating simultaneously through the results of the different queries (with an average score of 2.18, 10 was "very difficult"). In summary all participants were willing to use this application (with an average score of 8.5) and believed it was worth sharing (with an average score of 8.4).

Participants reported numerous use cases in which they envision themselves using *PRiSMA*. However, a recurrent theme was the search for images that illustrate abstract concepts, generally to be used for presentation slides. Participants particularly valued the diversity in the results obtained when automatically branching the query over one of the available facets. Participants equally valued the structured presentation of the results, which allowed to keep the search focused. One of the users was particularly interested in the use of *PRiSMA* for news related images, in order to browse through multiple events in different locations, inline with the search depicted in Section 9.2. Other suggested uses included entertainment (*e.g.*, "*search for rare and ambiguous terms*" to see how *PRiSMA* organizes the results). Three users, including the graphic designer, thought *PRiSMA* was very good for inspirational purposes.

Participants were then asked to think of other applications that would allow them for similar results. Only two participants responded, one participant mentioned Flickr[5] while the other mentioned both Google Images[6] and Cooliris.[7] Nonetheless, they both expressed a clear preference for the parallel search approach as it allowed a "*more focused search*".

To further ensure that participants perceived *PRiSMA* as a novel and useful image search application, we wanted participants to compare *PRiSMA* with the use of the browser's tabs as a way to perform parallel browsing. All participants were not only familiar with the use of tabs, but recognized

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

[6] https://www.google.com/imghp/

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

themselves as users who make an intensive use of the browser's tab for searching and browsing in parallel (with an average score of 9.1). While they understood what makes the two options comparable, they recognized the two as very different (with an average score of 7.64) with a clear preference for searching images using *PRiSMA* over using multiple tabs (with an average score of 8.2). Users valued the possibility of seeing all results at once, in one single page. They also valued the fact that a query can be branched automatically by any given facet (*e.g.*, colors, geographical location or topic).

To conclude, we have presented *PRiSMA*, an image search application intended to facilitate and promote searching images in parallel. We have illustrated how, by facilitating the exploration of multiple queries in an orderly fashion, *PRiSMA* can help users have a better account of the result space, diversify the results and conveniently broad or narrow the search space. We have illustrated use cases of *PRiSMA* for creative, educational and editorial uses. Of course, these examples should be further developed and eventually tested. In this early stage of development, our main concern was to learn from users whether the inherent complexity of parallel search overshadows its potential benefits. To clarify this concern, we conducted a study on 11 users with surprisingly positive results. Users unequivocally deemed that the complexity was well justified. Of course, future work should include a more rigorous and extensive study not only on the application's principles but also to evaluate the actual performance of the application.

9.2. Exploring Participation in Public Events

The structure of a social network is time-dependent, as relationships between entities change in time. In large networks, static or animated visualizations are often insufficient to capture all the information about the interactions between people over time, which could be captured better by interactive interfaces. We propose a novel system for exploring the interactions of entities over time, and support it with an application that displays interactions of public figures at events.

In the context of image search, people often query and browse photos of celebrities and public figures [75]. Automated query analysis allows search engines to identify the queried person and display structured information on the result screen (*e.g.* biographical information, birth year, related people, *etc.*). The information is usually an aggregated summary of a person's life and does not allow the user to further explore important events, nor the social interactions of a celebrity.

Representing events in a person's life, and especially social interactions, can help us to gain a better understanding of a person, not just as a standalone entity but also as an individual in a social environment. A person at a particular instant of time is not just a set of properties (*e.g.* hair color, job, birth date, *etc.*), but is also defined by the connections to other people. Displaying the interactions of entities over time is a challenging task because of the conflation of the temporal and relational dimensions.

In this section we present *Metro*, a system for exploring social interactions over time, leveraging information about participation in public events, using the paradigm of crossing *life lines* over time. *Metro* provides functions to explore online content in an unconventional yet practical way, allowing complex exploration of the information space by querying, pivoting over people and events, and inspecting the context information around the visualized interactions (photos of events and related people). *Metro* not only allows the user to explore existing social interactions and visualize their temporal characteristics, *i.e.*, are they sporadic, periodic or clustered around a particular date, but it also helps the user to discover tempo-structural holes in the network, *i.e.*, moments in time when links between people are missing. Through interactive search, *Metro* can retrieve people that are connected to a particular person and *not* to another. To the best of our knowledge, this work is the first to present such feature automatically.

Example *We present an example to motivate the* Metro *approach. Consider the case of a person living in different geographical locations. His or her social network could be composed of separate components. This is often due to the geographical distance between the people one knows. If we represent the social network as a graph, we would see the person acting as a hub across communities. Without any further information, we are not able to reconstruct the life of this person, nor the reason why he or she is connecting such heterogeneous communities. By exploiting time and structure jointly, we can understand if the person was interacting simultaneously with multiple communities or if he or she was interacting with a community at a time. Moreover we may be able to observe frequent and time-independent interaction,* i.e.*, people with whom he or she interacts across locations (*e.g. *family, long-lasting friends,* etc.*). Displaying participation to events instead of explicit connections allows us to distinguish between currently active and non-active relationships.*

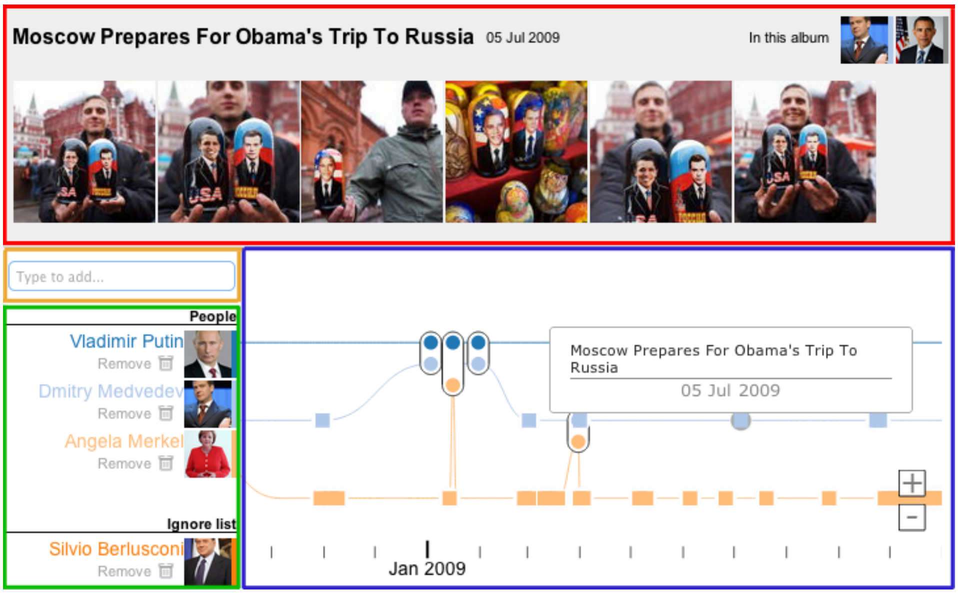


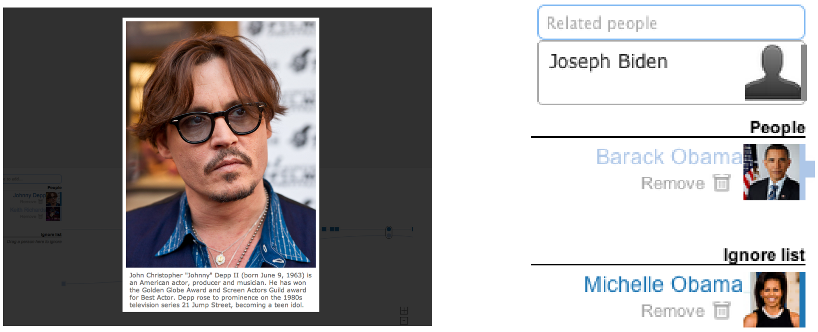
Figure 9.4: The interface containing the event bar (red), the search field (yellow), the people and ignore lists (green), and the timeline (blue).

The methodology we present could naturally adapt to many exploratory tasks, such as co-appearance in online social networks, or authorship of scientific publications. Now we present an application to search and browse photos of public figures, which is a frequent task in image search [81].

9.2.1. Exploring People and Events

Metro is a system to explore interactions among people over time. In the application we present, people are public figures and their interactions are the co-participations in events. Figure 9.4 shows a snapshot of the working system. We can see how three politicians interact during 2009. Each person is represented by a horizontal line of the same color as his or her name on the left. Lines join when the two people appear together in photo. We can see, for example, that Vladimir Putin visits Angela Merkel during mid-January. Along the blue line there is a point surrounded by a gray circle. This is the currently selected point. It refers to the preparation of United States president Barack Obama to visit Russia. The photos of the event are shown in the upper part.

The front-end of this demo has been fully developed in HTML 5 and the



John Christopher "Johnny" Depp II (born June 9, 1963) is an American actor, producer and musician. He has won the Golden Globe Award and Screen Actors Guild award for Best Actor. Depp rose to prominence on the 1980s television series 21 Jump Street, becoming a teen idol.

(a) The profile of a celebrity.

(b) Recommendation of people.

Figure 9.5: The profile of a user and the recommendation box as they appear in the interface.

back-end uses PHP 5[8] and MySQL.[9] The application uses the Getty Dataset described in Section 3.3.9.

9.2.2. Interface Structure

In this section we describe the structure of the interface. Figure 9.4 shows the interface of *Metro*; components of the interface are highlighted by different colors to ease the description.

Interactions between people are displayed with intersecting *life lines* on the top of a timeline (bottom right module, highlighted in blue). The horizontal axis represents time and the vertical one the social relations. Participation in events is represented as points on the person's life line. When multiple people attend the same event, the points are grouped together, placed on the topmost free row and enclosed in a black border. To minimize line crossings when many lines are present, we use a greedy algorithm to order them so that lines of people who appear together often are drawn close to each other. The timeline can be explored by horizontally zooming or scrolling.

Hovering on a point opens a text-box with a short event description, while clicking on it loads related pictures in the *event bar* (red box in the figure), together with the full event description and the list of other people attending the event.

[8] http://php.net/

[9] http://www.mysql.com/

On the left in the green box is the *people list*, which contains the list of currently displayed people. Clicking on a person's photo displays the biographical information, extracted from Wikipedia (see Figure 9.5a). People in the *people list* can be dragged down to the *ignore list*. The timeline only shows events in which at least a person in the people list appears and *no* person in the ignore list appears.

9.2.3. Recommendation Algorithm

The people list can be expanded by searching for a person's name in the search field on top (yellow). If no query is typed, clicking on the search field opens a box with a *recommendation list* of the people who mostly co-occurred with the people already present in the list (see Figure 9.5b). The recommendation algorithm takes as input two disjoint sets of people, P_+ (the current people list) and P_- (the ignore list), and the current time frame on the timeline $t_1 \leq t_2$. The suggested people should be tightly connected to the ones in P_+ but not to the ones in P_- in the given time frame.

The score assigned to people for ranking is computed as follows. First, let $c : \mathbb{R} \times \|P\| \times \|P\| \mapsto \mathbb{R}$ be the co-occurrence function, where $c(t, p_k, p_h)$ returns the number of photos taken at time t in which both p_k and p_h appear. Second, the time-constrained co-occurrence function $\bar{c}(p_1, p_2) = \int_{t_1}^{t_2} c(t, p_1, p_2)\ \mathrm{d}t$ is created to quantify how often people co-occur in a specified time interval. Finally, the person score p, which is used to rank people for the recommendation list, is computed as:

$$score(p) = \frac{\prod_{p_+ \in P_+} \bar{c}(p, p_+)}{\prod_{p_- \in P_-} \bar{c}(p, p_-)}.$$

An example of recommendation is shown in Figure 9.5b. In this case $p_+ = \{$ *Barack Obama* $\}$ and $p_- = \{$ *Michelle Obama* $\}$. The recommended person, the vice-president of the United States *Joseph Biden*, appears often with Barack Obama but not with Michelle Obama.

9.2.4. Functionality

Unlike previous work, *Metro* allows users to explore the history interactions over time at different granularities and across several dimensions. This is done by means of the following functions.

Search people. The social space is explored by searching for a person's name. When adding new life lines in the interface, the intersections in common events are dynamically adapted.

Slice over a social or a time dimension. Display all events for a person in the timeline or all the attendees at an event in the event bar. The slicing works also for group of people, when more than one are selected.

Context exploration. The interaction between public figures during events is contextualized by the content displayed in the event bar. Pictures related to the events are shown, together with the full set of attendees.

Pivoting. People related to the ones displayed and to the current time frame are recommended with the algorithm described above, allowing the user to pivot from one person to another based on their past co-appearances. Moreover, event attendees shown in the event bar can be added to the people list. The iteration of this process allows to move smoothly through the space and find related entities [48].

In summary, *Metro* is a system to explore people's participation in public events. The interface jointly represents social interactions and the temporal dimension, and allows the user to browse through people using either. This rich set of features enables an effective way to explore the information space that could be adapted to different domains.

Chapter **10**

Conclusions

The path of understanding of user browsing behavior started with a general analysis of browsing behavior in social media platforms. First, we studied the entry points of web sessions, *i.e.*, the URL from where people access the website (Chapter 4). We showed that this feature has an impact on the next actions the user will take. Then (Chapter 5), we studied the evolution of both search and non-search sessions. We presented an interactive application that leverages the linear structure of browsing sessions to enhance content discovery. In Chapter 6, we extracted frequent browsing patterns from the sessions using data mining techniques.

Based on the insights acquired during the analysis, we presented probabilistic generative models of browsing sessions. The models capture and cluster behavior of the users online. Models can be used for gaining insights about frequent patterns as well as for predicting the actions of users, thus allowing to present the right content in the right place.

Finally, we tried to extend traditional structures of browsing sessions. We presented interfaces to browse images using completely new paradigms: parallel browsing in the case of *PRiSMA* and multidimensional browsing in the case of *Metro*.

10.1. Main Results

In this section we go through the goals of the thesis presented in Section 1.1 and discuss the results we obtained from the research.

The first goal is to access if, among all the data present in the server logs, some information is more useful than other for understanding the browsing behavior of users. The models presented in this thesis show that this is the case. Indeed, the contextual information, such as the referrer URL, appears to be a much better indicator of what the user is going to see than, for example, user demographics.

Secondly, the thesis aimed at understanding if there is a distinction between search and browsing in the structure of web sessions. This goal resulted in a much broader analysis of user browsing behavior in photo sharing platforms. However, comparing the results of Sections 5.1.2 and 5.2.1, we discover that search sessions are indeed shorter and that their structure can be represented as a tree. On the contrary, browsing sessions consists of long sequence of photos, the photostreams. We show that this structure is so frequent, that it is possible to represent a session as a sequence of photostreams, thus making the representation more compact.

For the purpose of exploring ways to combine search and browsing we presented *PRiSMA*, an application of image search that allows for parallel browsing. Through a user study we show that people are willing to use the application and parallel browsing is not perceived as an added complexity but rather as novel and useful approach. Moreover, participants perceived that the structured presentation allows for a more focused search.

The last goal is related to the content that is browsed. We were interested in the characteristics of the browsing graph, and in the existence of frequent information needs shared by many people. We discover that it is possible to build a photostream browsing graph and that this graph shows clear communities. Moreover, the exemplary summaries of sessions in Section 6.4.2 are indeed the frequent browsing paths that user take to satisfy their information needs.

10.2. Lessons Learned

In this section, we present some conclusions that we obtained from the work that has been done. We hope that this section can suggest ways to improve existing applications and system as well as encourage further research in the topic.

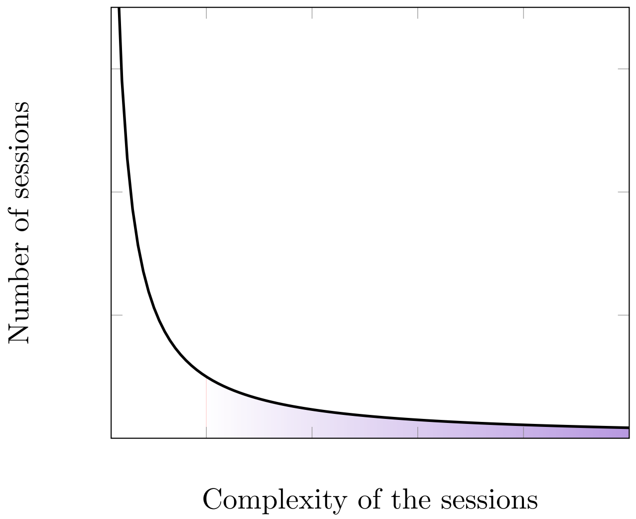


Figure 10.1: The theoretical distribution of “complexity” of sessions against the number of sessions.

10.2.1. The Long Tail of User Browsing Behavior

During the thesis, one of the problems we encountered was the size of the data. This made going manually through all sessions intractable and the hypothesis were stated observing a limited amount of examples.

In addition to this, most sessions do not contain much information that can be exploited. For example, in Section 3.3.4 we showed that many sessions of the users in news portals are very short and contain very simple activities (*e.g.*, click on the first article in the homepage).

Figure 10.1 shows the theoretical distribution of information that can be extracted from sessions. The figure is an exemplary distribution and is not based on any real data, although we may get a similar distribution by looking for example at the number of items visited in sessions (*e.g.*, Figure 5.4a). What we try to communicate with the plot is that the majority of user sessions are relatively simple, and that there is a fraction of sessions (the tail on the right in the figure), that are quite complex.

In order to leverage sessions to build intelligent applications, both “simple” and “complex” sessions are useful, depending on the case. For example, if the task is to optimize daily activities (*e.g.*, improving the layout of news portals to make news articles more accessible to the user) simple sessions can be used, since they contain a strong signal of the activity of the users. This is what we did in Chapter 8.

The tail of the distribution, instead, can be used to develop exploratory applications that could lead to *serendipitous* discoveries. “Serendipity” is defined in the Oxford English Dictionary as “the faculty of making happy and unexpected discoveries by accident”. The term was originally coined by Horace Walpole in a letter to Sir Horace Mann on January 28, 1754 [122]. Serendipity has been long studied in the context of information retrieval (empirical studies have been performed by Foster and Ford [56] and Roberts [124]) but many authors expressed concerns about the opportunity for serendipitous information encounters in information retrieval systems [65, 39], since artificial filters and document ranking can excessively limit searches.

We argue that complex sessions contain the most interesting data for serendipitous applications. In the case of search, long sessions are due to the user struggling to get a result, or consecutive searches towards the same goal (*cf.*, *research missions* [47]). In the case of browsing, long sessions do not necessarily mean that the user is looking for something, but rather that he or she is moving in the information space. However, this movement is not random but rather structured. For example, in Section 5.2 we showed that users browse photos using the notion of photostream. Following the steps of other users may therefore enhance content discovery, as shown in the visualization presented in Section 5.3.

10.2.2. The Importance of Context

Chapters 7 and 8 proved the importance of context in the browsing behavior of users. When dealing with user browsing, “tell me what you see” is not enough to be able to “tell you what you are”. One has to add the *where* and the *when* also.

With the advent of mobile devices, context is becoming more and more important. Think for example in Google Now.[1] It is an intelligent personal assistance that uses context to make recommendations or deliver information to the user. For example, Google Now may recommend nearby restaurants based on one’s location, or display traffic information to the workplace early in the morning.

[1] http://www.google.com/now/

Context is still underexploited in the web. There are indeed opportunities in customizing the content and portal to better accommodate contextual user needs.

10.2.3. Per-session or Per-user

There are generally two approaches when dealing with modeling of user browsing behavior, as seen when presenting related work (Section 2.3): the per-session and per-user basis. In this thesis, we adopted the per-session approach. There are multiple advantages to do so.

First of all, the user is sometimes too coarse-grained and sometimes does not have a direct relationship with the individual. For example, it may appear that two or more people, *e.g.*, in a family, use the same terminal to access the internet. In this case, modeling the behavior of the user may average the contributions of the family members, obtaining useless if not even wrong results. This does not happen with sessions. It is indeed reasonable to assume in the majority of cases that a session has been done by a single individual, especially when they are short enough.

Secondly, the per session basis solves the issue of the cold start problem, *i.e.*, the case in which a new user is accessing the website. When modeling sessions and not users, each sessions is treated equally, whether it belongs to a new user as well as to a known user.

Finally, per-session models can be applied virtually anywhere, since they do not require user login. They are insensitive to privacy concerns since the information they use is naturally aggregated.

10.2.4. There Is No *One Model to Rule Them All*

The final remark is about models of browsing sessions. In this thesis we have done a few different assumptions about the shape of browsing sessions, namely sequential (*e.g.*, Section 5.2), hierarchical (Section 5.1), or parallel (Section 9.1).

To the limit, complex applications as *PRiSMA* give rise to sessions that show a *graph* structure. Indeed, the possibility to organize the filmstrips by topic, color, or location (*cf.*, Section 9.1.1) splits a filmstrip into different branches, thus making a session a tree. If we allow merging two parallel strips, the session may become a graph.

The choice of the structure of the sessions has great influence not only on the complexity of models but also on the user interaction. Allowing parallel browsing made *PRiSMA* a brand new application that allows the user to better explore the information space.

10.3. Future Work

The directions in which the work presented in this thesis can be expanded are presented in the following paragraphs.

Characterization. Regarding the analysis of browsing data, future work may include deeper analysis of user actions within each page layout, as well as content analysis and meta-data analysis to gain insights into how the content itself affects the navigation patterns. Moreover, it would be interesting to conduct a detailed comparison of image search behavior on photo-sharing platforms with general web image search, also aimed at understanding the intent behind image search queries.

Summarization. Session summarization can be extended from various perspectives. First, taking inspiration from the wide literature on pattern mining, more efficient algorithms can be devised: for instance by a depth-first visit of the search space or by taking advantage of the fact that we only look for maximal patterns. Second, methods can be devised to adaptively decide which constraint to check first as the computation progresses [21]. Finally, future research can study how to avoid setting rigid thresholds and make the constraints soft [19].

Modeling. Regarding modeling of user browsing behavior, it would be interesting to study the impact of our model in terms of user understanding in greater detail. A straightforward way to test the acceptance rate of optimized web pages is to compare the number of clicks in a controlled environment with that of an alternative layout. Another interesting line of research deals with ways to combine session- and user-based approaches. Our contextual approach is *orthogonal* to personalized methods and a combination could possibly benefit from both worlds.

***PRiSMA* and *Metro*.** Finally, the new light shed on alternative session models by Chapter 9 could be augmented by improving the two applications. For *PRiSMA*, we also intend to explore the use of other image collections with different characteristics (*e.g.*, Flickr) and implement different search dimensions, for example time or the images' source. Both features would be particularly useful for journalists as they will allow to compare images across time and images from different sources. For *Metro*, we plan to include functionalities to reduce and aggregate the information displayed, by clustering similar entities. Finally, it would be interesting to compare *Metro* to other interfaces performing similar tasks and evaluate its performance by means of a user study .

Bibliography

Each reference indicates at the end the pages where it appears.

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Databases (VLDB'94)*, pages 487–499, 1994. 77

[2] Mohamed Ahmed, Stella Spagna, Felipe Huici, and Saverio Niccolini. A peek into the future: Predicting the evolution of popularity in user generated content. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 607–616. ACM, 2013. 14

[3] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. Influence and correlation in social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*, pages 7–15, 2008. 13

[4] Corin R. Anderson, Pedro Domingos, and Daniel S. Weld. Relational markov models and their application to adaptive web navigation. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, 2002. 14, 15

[5] Paul André, Edward Cutrell, Desney S. Tan, and Greg Smith. Designing novel image search interfaces by understanding unique characteristics and usage. In *Human-Computer Interaction INTERACT 2009*, volume 5727 of *Lecture Notes in Computer Science*, pages 340–353. Springer, 2009. 10, 61

[6] R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press, 1999. 2

[7] Eytan Bakshy, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the 2011 International Conference on Web Search and Data Mining*, pages 65–74, 2011. 13

[8] Roja Bandari, Sitaram Asur, and Bernardo A Huberman. The pulse of news in social media: Forecasting popularity. In *International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2012. 14

[9] Frank M. Bass. A new product growth model for consumer durables. *Management Science*, 15:215–227, 1969. 13

[10] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks. In *International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2009. 17

[11] David Bawden. Information systems and the stimulation of creativity. *Journal of Information Science*, 12(5):203, 1986. 2

[12] Grigory Begelman, Philipp Keller, Frank Smadja, et al. Automated tag clustering: Improving search and exploration in the tag space. In *Collaborative Web Tagging Workshop at WWW2006*, pages 22–26, 2006. 131

[13] Richard Bellman. On a routing problem. Technical report, DTIC Document, 1956. 75

[14] Fabrício Benevenuto, Tiago Rodrigues, Meeyoung Cha, and Virgílio Almeida. Characterizing user behavior in online social networks. In *ACM SIGCOMM*, pages 49–62. ACM, 2009. 9

[15] Anastasia Bezerianos, Pierre Dragicevic, J. Fekete, Juhee Bae, and Ben Watson. Geneaquilts: A system for exploring large genealogies. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6): 1073–1081, 2010. 17

[16] Daniel Billsus and Michael J. Pazzani. A hybrid user model for news story classification. *Courses and Lectures-International Centre for Mechanical Sciences*, 99:108, 1999. 14

[17] Daniel Billsus and Michael J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2):147–180, 2000. 14

[18] Christopher M. Bishop et al. *Pattern Recognition and Machine Learning*. Springer, 2006. 117

[19] Stefano Bistarelli and Francesco Bonchi. Interestingness is not a dichotomy: Introducing softness in constrained pattern mining. In *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2005)*, 2005. 91, 148

[20] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis, and Sebastiano Vigna. The query-flow graph: model and applications. In *Proceedings of the 2008 ACM Conference on Information and Knowledge Management (CIKM 2008)*, 2008. 10

[21] Francesco Bonchi, Fosca Giannotti, Alessio Mazzanti, and Dino Pedreschi. Adaptive constraint pushing in frequent pattern mining. In *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2005)*, 2003. 148

[22] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002. 10, 46

[23] Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. Visualization of navigation patterns on a web site using model-based clustering. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'00)*, 2000. 14, 118, 124

[24] Robert Capra and Jason Raitz. Diamond browser: Faceted search on mobile devices. In *Fifth Workshop on Human-Computer Interaction and Information Retrieval (HCIR 2011)*, 2011. 16

[25] Carlos Castillo, Mohammed El-Haddad, Jürgen Pfeffer, and Matt Stempeck. Characterizing the life cycle of online news stories using social media reactions. *arXiv preprint arXiv:1304.3010*, 2013. 14

[26] Lara D. Catledge and James E. Pitkow. Characterizing browsing strategies in the world-wide web. *Computer Networks and ISDN systems*, 27(6):1065–1073, 1995. 9, 11, 23

[27] Deepayan Chakrabarti, Ravi Kumar, and Kunal Punera. Quicklink selection for navigational query results. In *Proceedings of the 18th International Conference companion on World Wide Web*, pages 391–400, 2009. 13

[28] Gloria Chatzopoulou, Cheng Sheng, and Michalis Faloutsos. A first step towards understanding popularity in youtube. In *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, pages 1–6. IEEE, 2010. 14

[29] Karine Chevalier, Cécile Bothorel, and Vincent Corruble. Discovering rich navigation patterns on a web site. In *Proceedings of Discovery*

Science, 2003. 14, 15

[30] Luca Chiarandini and Alejandro Jaimes. Browsing-based content discovery. In *Proceedings of the Designing Interactive Systems Conference. ACM*, 2012. 41

[31] Luca Chiarandini, Michele Trevisiol, and Alejandro Jaimes. Discovering social photo navigation patterns. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages 31–36. IEEE, 2012. 5, 32

[32] Luca Chiarandini, Luca Maria Aiello, Neil OHare, and Alejandro Jaimes. Metro: Exploring participation in public events. In *Social Informatics*, pages 40–45. Springer, 2013. 129

[33] Luca Chiarandini, Przemyslaw A. Grabowicz, Michele Trevisiol, and Alejandro Jaimes. Leveraging browsing patterns for topic discovery and photostream recommendation. In *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media, Cambridge, MA*, pages 71–80, 2013. 41

[34] Flavio Chierichetti, Ravi Kumar, and Andrew Tomkins. Stochastic models for tabbed browsing. In *Proceedings of the 19th international conference on World wide web*, pages 241–250. ACM, 2010. 10, 14, 42

[35] Youngok Choi. Investigating variation in querying behavior for image searches on the web. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–10, 2010. 131

[36] Chun Wei Choo, Brian Detlor, and Dan Turnbull. Information seeking on the web: An integrated model of browsing and searching. *first monday*, 5(2), 2000. 14

[37] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009. 98, 102

[38] James Samuel Coleman, Elihu Katz, Herbert Menzel, et al. *Medical Innovations: A Diffusion Study*. Bobbs Merrill, 1966. 13

[39] James W. Cooper and John M. Prager. Anti-serendipity: finding useless documents and similar documents. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 8–pp. IEEE, 2000. 146

[40] David J. Crandall, Dan Cosley, Daniel P. Huttenlocher, Jon M. Kleinberg, and Siddharth Suri. Feedback effects between similarity and social influence in online communities. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data*

Mining (KDD'08), pages 160–168, 2008. 13

[41] Abhinandan S. Daş, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280. ACM, 2007. 119

[42] Resul Daş and İbrahim Türkoğlu. Creating meaningful data from web logs for improving the impressiveness of a website by using path analysis method. *Expert Systems with Applications*, 36(3):6635–6644, 2009. 14, 15

[43] Resul Daş and İbrahim Türkoğlu. Extraction of interesting patterns through association rule mining for improvement of website usability. *Istanbul University-Journal of Electrical & Electronics Engineering*, 9 (18), 2010. 14, 15

[44] Arthur P. Dempster, Nan M. Laird, Donald B. Rubin, et al. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977. 116

[45] Mukund Deshpande and George Karypis. Selective markov models for predicting web page accesses. *ACM Transactions on Internet Technology*, 4(2):163–184, 2004. 14, 15

[46] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 57–66, 2001. 13

[47] Debora Donato, Francesco Bonchi, Tom Chi, and Yoëlle S. Maarek. Do you want to take notes?: identifying research missions in yahoo! search pad. In *Proceedings of the 19th International Conference companion on World Wide Web*, pages 321–330, 2010. 146

[48] Marian Dörk, Nathalie Henry Riche, Gonzalo Ramos, and Susan T. Dumais. Pivotpaths: Strolling through faceted information spaces. *IEEE Trans. Vis. Comput. Graph*, 18(12):2709–2718, 2012. 142

[49] Peter G. B. Enser. Query analysis in a visual information retrieval context. *Journal of Document and Text Management*, 1(1):25–52, 1993. 11, 46

[50] Jianping Fan, Daniel A Keim, Yuli Gao, Hangzai Luo, and Zongmin Li. Justclick: Personalized image recommendation via exploratory search from large-scale flickr images. *Circuits and Systems for Video Technology*, 19(2):273–288, 2009. 16

[51] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, et al. Knowledge discovery and data mining: Towards a unifying framework. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, volume 96, pages 82–88, 1996. 20

[52] Flavio Figueiredo, Fabrício Benevenuto, and Jussara M. Almeida. The tube over time: characterizing popularity growth of youtube videos. In *Proceedings of the 2011 International Conference on Web Search and Data Mining*, pages 745–754. ACM, 2011. 12, 32

[53] Timothy La Fond and Jennifer Neville. Randomization tests for distinguishing social influence and homophily effects. In *Proceedings of the 19th International Conference companion on World Wide Web*, pages 601–610, 2010. 13

[54] LR Ford and Delbert Ray Fulkerson. *Flows in networks*, volume 3. Princeton Princeton University Press, 1962. 75

[55] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3–5):75 – 174, 2010. ISSN 0370-1573. 55

[56] Allen Foster and Nigel Ford. Serendipity and information seeking: an empirical study. *Journal of Documentation*, 59(3):321–340, 2003. 146

[57] Jill Freyne, Rosta Farzan, Peter Brusilovsky, Barry Smyth, and Maurice Coyle. Collecting community wisdom: integrating social search & social navigation. In *Proceedings of the 12th International Conference on Intelligent User Interfaces*, pages 52–61. ACM Press, 2007. 61

[58] Michael Gamon and Arnd Christian König. Navigation Patterns from and to Social Media. In *International AAAI Conference on Weblogs and Social Media (ICWSM)*, pages 203–206, 2009. 11

[59] Giorgos Giannopoulos, Ulf Brefeld, Theodore Dalamagas, and Timos Sellis. Learning to rank user intent. In *Proceedings of the 2011 ACM Conference on Information and Knowledge Management (CIKM 2011)*, pages 195–200. ACM, 2011. 14

[60] Jacob Goldenberg, Barak Libai, and Eitan Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12(3):211–223, 2001. 13

[61] Amit Goyal, Francesco Bonchi, and Laks V. S. Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the 2010 International Conference on Web Search and Data Mining*, pages 241–250, 2010. 13

[62] Amit Goyal, Francesco Bonchi, and Laks V. S. Lakshmanan. A data-

based approach to social influence maximization. *Proceedings of the 33st International Conference on Very Large Databases (VLDB'11)*, 5(1):73–84, 2011. 13

[63] Przemyslaw A. Grabowicz, José J. Ramasco, Esteban Moro, Josep M. Pujol, and Victor M. Eguiluz. Social features of online networks: The strength of intermediary ties in online social media. *PLoS ONE*, 7(1), 2012. 57

[64] Şule Gündüz and M. Tamer Özsu. A web page prediction model based on click-stream tree representation of user behavior. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, 2003. 15

[65] Ted Gup. Technology and the end of serendipity. *Education Digest*, 63:48–50, 1998. 146

[66] Mangesh Gupte, Pravin Shankar, Jing Li, Shanmugauelayut Muthukrishnan, and Liviu Iftode. Finding hierarchy in directed online social networks. In *Proceedings of the 20tt International Conference companion on World Wide Web*, pages 557–566, 2011. 74, 75, 76, 79

[67] Peter Haider, Luca Chiarandini, and Ulf Brefeld. Discriminative clustering for market segmentation. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*, pages 417–425. ACM, 2012. 15

[68] Peter Haider, Luca Chiarandini, Ulf Brefeld, and Alejandro Jaimes. Contextual models for user interaction on the web. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD 2012)*, 2012. 110

[69] Malik Tahir Hassan and Asim Karim. Impact of behavior clustering on web surfer behavior prediction. *Journal of Information Science and Engineering*, 27:1855–1870, 2011. 14, 15

[70] Susan Havre, Beth Hetzler, and Lucy Nowell. Themeriver: Visualizing theme changes over time. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, pages 115–123. IEEE, 2000. 17

[71] Birgit Hay, Geert Wets, and Koen Vanhoof. Mining navigation patterns using a sequence alignment method. *Knowledge and Information Systems*, 6:150–163, 2004. 15

[72] Shawndra Hill, Foster Provost, and Chris Volinsky. Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science*, 21(2):256–276, 2006. 13

[73] Natascha Hoebel and Roberto V. Zicari. On clustering visitors of a

web site by behavior and interests. In *Advances in Intelligent Web Mastering*, volume 43, pages 160–167. Springer Berlin / Heidelberg, 2007. 14, 15

[74] Christoph Hölscher and Gerhard Strube. Web search behavior of internet experts and newbies. *Computer networks*, 33(1):337–346, 2000. 9

[75] Alex Holub, Pierre Moreels, and Pietro Perona. Unsupervised clustering for google searches of celebrity images. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pages 1–8. IEEE, 2008. 137

[76] Jun Huan, Wei Wang, and Jan Prins. Efficient mining of frequent subgraphs in the presence of isomorphism. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)*, pages 549–552, 2003. 12

[77] Jun Huan, Wei Wang, Jan Prins, and Jiong Yang. Spin: mining maximal frequent subgraphs from graph databases. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, pages 581–586, 2004. 12

[78] Jeff Huang and Ryen W. White. Parallel browsing behavior on the web. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, pages 13–18. ACM, 2010. 11, 16

[79] Xiangji Huang, Fuchun Peng, Aijun An, and Dale Schuurmans. Dynamic web log session identification with statistical language models. *Journal of the American Society for Information Science and Technology*, 55(14):1290–1303, 2004. 19

[80] Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000)*, pages 13–23, 2000. 12

[81] Bernard J. Jansen. Searching for Digital Images on the Web. *Journal of Documentation*, 64(1):81–101, 2008. 48, 139

[82] Bernard J. Jansen, Abby Goodrum, and Amanda Spink. Searching for multimedia: analysis of audio, video and image web queries. *World Wide Web*, 3(4):249–254, 2000. 10

[83] Bernard J. Jansen, Danielle L. Booth, and Amanda Spink. Determining the informational, navigational, and transactional intent of Web queries. *Information Processing and Management*, 44:1251–1266,

2008. 10

[84] Jing Jiang, Christo Wilson, Xiao Wang, Wenpeng Sha, Peng Huang, Yafei Dai, and Ben Y. Zhao. Understanding latent interactions in online social networks. In *Proceedings of the 10th annual conference on Internet measurement*, pages 369–382. ACM, 2010. 9

[85] Long Jin, Yang Chen, Tianyi Wang, and Athanasios V. Vasilakos. Understanding user behavior in online social networks: A survey. *IEEE Communications Magazine*, page 145, 2013. 9

[86] Yushi Jing, Henry Rowley, Jingbin Wang, David Tsai, Chuck Rosenberg, and Michele Covell. Google image swirl: a large-scale content-based image visualization system. In *Proceedings of the 21st International Conference companion on World Wide Web*, pages 539–540. ACM, 2012. 16

[87] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th international ACM SIGIR conference on Research and development in information retrieval*, 2005. 14

[88] Michael Kaufmann and Dorothea Wagner. *Drawing graphs: methods and models*, volume 2025. Springer Verlag, 2001. 62

[89] Diane Kelly and Jaime Teevan. Implicit feedback for inferring user preference: A bibliography. *SIGIR Forum*, 37(2):18–28, 2003. 14

[90] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, 2003. 13

[91] Nam Wook Kim, Stuart K. Card, and Jeffrey Heer. Tracing genealogical data with timenets. In *Proceedings of the International Conference on Advanced Visual Interfaces*, pages 241–248. ACM, 2010. 17

[92] Lerman Kristina. Social browsing & information filtering in social media. *CoRR*, abs/0710.5697, 2007. 11

[93] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, pages 79–86, 1951. 96

[94] Ravi Kumar and Andrew Tomkins. A characterization of online browsing behavior. In *Proceedings of the 19th international conference on World wide web*, pages 561–570. ACM, 2010. 9, 11

[95] Michihiro Kuramochi and George Karypis. Frequent subgraph discovery. In *Proceedings of the First IEEE International Conference on*

Data Mining (ICDM'01), pages 313–320, 2001. 12

[96] Kristina Lerman and Laurie Jones. Social browsing on flickr. *Arxiv preprint cs/0612047*, pages 1–4, 2006. 11, 12, 32

[97] Jure Leskovec, Ajit Singh, and Jon M. Kleinberg. Patterns of influence in a recommendation network. In *Advances in Knowledge Discovery and Data Mining*, pages 380–389, 2006. 13

[98] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306. ACM, 2011. 14, 15

[99] Yu-Ru Lin, Hari Sundaram, Munmun De Choudhury, and Aisling Kelliher. Temporal patterns in social media streams: Theme discovery and evolution using joint analysis of content and context. In *Multimedia and Expo (ICME), 2009 IEEE International Conference on*, pages 1456–1459, 2009. 11

[100] Marek Lipczak, Michele Trevisiol, and Alejandro Jaimes. Analyzing Favorite Behavior in Flickr. In *Advances in Multimedia Modeling*, 2013. 11

[101] Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 17(4):491–502, 2005. 93

[102] Mathias Lux, Christoph Kofler, and Oge Marques. A classification scheme for user intentions in image search. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 3913–3918, New York, NY, USA, 2010. ACM. 10, 46

[103] Lucrezia Macchia, Francesco Bonchi, Francesco Gullo, and Luca Chiarandini. Mining summaries of propagations. *Proceedings of the 13th IEEE International Conference on Data Mining (ICDM'13)*, 0: 498–507, 2013. 71

[104] Eren Manavoglu, Dmitry Pavlov, and C. Lee Giles. Probabilistic user behavior models. In *Proceedings of the Third IEEE International Conference on Data Mining*, 2003. 14

[105] Silviu Maniu, Neil O'Hare, Luca Maria Aiello, Luca Chiarandini, and Alejandro Jaimes. Search behaviour on photo sharing platforms. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013. 41

[106] Andrew McCallum, Kamal Nigam, and Lyle H. Ungar. Efficient

clustering of high-dimensional data sets with application to reference matching. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'00)*, pages 169–178, 2000. doi: 10.1145/347090.347123. 35

[107] Mark Meiss, John Duncan, Bruno Gonçalves, José J Ramasco, and Filippo Menczer. What's in a session: tracking individual behavior on the web. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 173–182. ACM, 2009. 11

[108] Luis Carlos Molina, Lluís Belanche, and Àngela Nebot. Feature selection algorithms: A survey and experimental evaluation. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 306–313. IEEE, 2002. 93

[109] Alan L. Montgomery, Shibo Li, Kannan Srinivasan, and John C. Liechty. Modeling online browsing and path analysis using clickstream data. *Marketing Science*, 23(4):579–595, 2004. 14

[110] Philip M. Morse. On browsing: the use of search theory in the search for information. *Bulletin of the Operations Research Society of America, Vol. 19 supplement, p.1*, 1971. 2

[111] Saket Navlakha, Rajeev Rastogi, and Nisheeth Shrivastava. Graph summarization with bounded error. In *SIGMOD*, pages 419–432, 2008. 13

[112] Radford M. Neal and Geoffrey E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*, 89:355–368, 1998. 118

[113] Siegfried Nijssen and Joost N. Kok. A quickstart in frequent structure mining can make a difference. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, pages 647–652, 2004. 12

[114] Michael Ogawa and Kwan-Liu Ma. Software evolution storylines. In *Proceedings of the 5th international symposium on Software visualization*, pages 35–42. ACM, 2010. 17

[115] Neil O'Hare, Luca Maria Aiello, and Alejandro Jaimes. Predicting participants in public events using stock photos. In *Proceedings of the 13th ACM international conference on Multimedia*, 2012. 28

[116] George Pallis, Lefteris Angelis, and Athena Vakali. Validation and interpretation of web users' sessions clusters. *Information Processing & Management*, 43(5):1348–1367, 2007. 14, 15

[117] Young-Hoon Park and Peter S. Fader. Modeling browsing behavior

at multiple websites. *Marketing Science*, 23(3):280–303, 2004. 14

[118] Peter Pirolli and Wai-Tat Fu. Snif-act: A model of information foraging on the world wide web. In *User Modeling 2003*, pages 45–54. Springer, 2003. 14

[119] Catherine Plaisant, Brett Milash, Anne Rose, Seth Widoff, and Ben Shneiderman. Lifelines: visualizing personal histories. In *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground*, pages 221–227. ACM, 1996. 17

[120] Kristen Purcell, Lee Rainie, Amy Mitchell, Tom Rosenstiel, and Kenny Olmstead. Understanding the participatory news consumer. Website, June 2010. URL `http://www.journalism.org`. 109

[121] Nikhil Rasiwasia, Pedro J. Moreno, and Nuno Vasconcelos. Bridging the gap: Query by semantic example. *IEEE Transactions on Multimedia*, 9(5), 2007. 131

[122] Theodore G. Remer. Serendipity and the three princes, from the peregrinaggio of 557. *University of Okalmahoma Press, Norman (OK)*, 1965. 146

[123] Kan Ren. FreeEye - Interactive Intuitive Interface for Large-scale Image Browsing. *Interface*, pages 757–760, 2009. 16

[124] Royston M. Roberts. Serendipity: Accidental discoveries in science. *Serendipity: Accidental Discoveries in Science, by Royston M. Roberts, pp. 288. ISBN 0-471-60203-5. Wiley-VCH, June 1989.*, 1, 1989. 146

[125] Daniel M. Romero, Brendan Meeder, and Jon M. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on Twitter. In *Proceedings of the 20tt International Conference companion on World Wide Web*, pages 695–704, 2011. 13

[126] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *National Academy of Sciences*, 105(4):1118–1123, 2008. 55

[127] Martin Rosvall and Carl T. Bergstrom. Mapping Change in Large Networks. *PLoS ONE*, 5(1):e8694+, Jan 2010. ISSN 1932-6203. doi: 10.1371/journal.pone.0008694. 17

[128] Martin Rosvall and Carl T. Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PLoS ONE*, 6(4):e18209, 04 2011. 55

[129] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. Prediction of

information diffusion probabilities for independent cascade model. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 67–75, 2008. 13

[130] Aju Thalappillil Scaria, Rose Marie Philip, Robert West, and Jure Leskovec. The last click: Why users give up information network navigation. In *Proceedings of the 2014 International Conference on Web Search and Data Mining*, 2014. 14

[131] Gerald Schaefer. A next generation browsing environment for large image repositories. *Multimedia Tools and Applications*, 47(1):105–120, 2010. 61

[132] Klaus Schoeffmann and David Ahlström. Similarity-based visualization for image browsing revisited. In *Multimedia (ISM), 2011 IEEE International Symposium on*, pages 422–427. IEEE, 2011. 61

[133] Klaus Schoeffmann, David Ahlström, and Christian Beecks. 3d image browsing on mobile devices. In *Multimedia (ISM), 2011 IEEE International Symposium on*, pages 335–336. IEEE, 2011. 16

[134] Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. Metro maps of science. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*, pages 1122–1130. ACM, 2012. 17

[135] Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. Trains of thought: Generating information maps. In *Proceedings of the 21st international conference on World Wide Web*, pages 899–908. ACM, 2012. 17

[136] A. K. Sharma, Amit Goel, Payal Gulati, et al. A Novel Approach for clustering web user sessions using RST. In *Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on*, pages 657–659. IEEE, December 2009. 12

[137] Sara Shatford. Analyzing the subject of a picture: a theoretical approach. *Cataloging and Classification Quarterly*, 6(3):39–62, Spring 1986. 46

[138] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12): 1349–1380, 2000. 11

[139] Ramakrishnan Srikant and Yinghui Yang. Mining web logs to improve website organization. In *Proceedings of the 10th International Conference companion on World Wide Web*, pages 430–437, 2001. 12,

13, 69

[140] Grant Strong, Enamul Hoque, Minglun Gong, and Orland Hoeber. Organizing and browsing image search results based on conceptual and visual similarities. *Advances in Visual Computing*, pages 481–490, 2010. 16

[141] Gabor Szabo and Bernardo A Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88, 2010. 14

[142] Taiki Takashita, Tsuyoshi Itokawa, Teruaki Kitasuka, and Masayoshi Aritsugi. Tag recommendation for flickr using web browsing behavior. In *Computational Science and Its Applications–ICCSA 2010*, volume 6017 of *Lecture Notes in Computer Science*, pages 412–421. Springer, 2010. 11

[143] Andrew Thatcher. Web search strategies: The influence of web experience and task type. *Information Processing & Management*, 44(3): 1308–1329, 2008. 17

[144] Yuanyuan Tian, Richard A. Hankins, and Jignesh M. Patel. Efficient aggregation for graph summarization. In *SIGMOD*, pages 567–580, 2008. 13

[145] Dian Tjondronegoro, Amanda Spink, and Bernard J. Jansen. A study and comparison of multimedia web searching: 1997-2006. *J. Am. Soc. Inf. Sci. Technol.*, 60(9):1756–1768, 2009. 11, 48

[146] Pancho Tolchinsky, Luca Chiarandini, and Alejandro Jaimes. Prisma: searching images in parallel. In *Proceedings of the 13th ACM international conference on Multimedia*, pages 985–988. ACM, 2012. 129

[147] Liang-Chun Jack Tseng, Dian Wirawan Tjondronegoro, and Amanda H. Spink. Analyzing web multimedia query reformulation behavior. In *The 14th Australasian Document Computing Symposium*. CSIRO, 2009. The contents of that proceedings can be freely accessed online (see Official URL). 10

[148] Athena Vakali, Jaroslav Pokornỳ, and Theodore Dalamagas. An overview of web data clustering practices. In *Current Trends in Database Technology-EDBT 2004 Workshops*, pages 597–606. Springer, 2005. 12

[149] Masoud Valafar, Reza Rejaie, and Walter Willinger. Beyond friendship graphs: a study of user interactions in Flickr. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 25–30. ACM, 2009. 11

[150] Thomas Valente. *Network Models of the Diffusion of Innovations.* Hampton Press, 1955. 13

[151] Hongning Wang, ChengXiang Zhai, Feng Liang, Anlei Dong, and Yi Chang. User modeling in search logs via a nonparametric bayesian approach. In *Proceedings of the 2014 International Conference on Web Search and Data Mining*, 2014. 11

[152] Ingmar Weber and Carlos Castillo. The demographics of web search. In *Proceedings of the 33th international ACM SIGIR conference on Research and development in information retrieval*, pages 523–530. ACM, 2010. 97

[153] Harald Weinreich, Hartmut Obendorf, Eelco Herder, and Matthias Mayer. Off the beaten tracks: exploring three aspects of web navigation. In *Proceedings of the 15th International Conference companion on World Wide Web*, pages 133–142. ACM, 2006. 131

[154] Stina Westman and Pirkko Oittinen. Image retrieval by end-users and intermediaries in a journalistic work context. In *Proceedings of the 1st international conference on Information interaction in context*, pages 102–110. ACM, 2006. 10, 11, 46, 48

[155] Stina Westman, Antti Lustila, and Pirkko Oittinen. Search strategies in multimodal image retrieval. In *Proceedings of the 1st international conference on Information interaction in context*, pages 13–20. ACM, 2008. 10, 42

[156] Alan Wexelblat and Pattie Maes. Footprints: history-rich tools for information foraging. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, pages 270–277. ACM, 1999. 61

[157] Ryen W. White and Steven M. Drucker. Investigating behavioral variability in web search. In *Proceedings of the 16th International Conference companion on World Wide Web*, pages 21–30. ACM, 2007. 10, 43

[158] Max L. Wilson, Ryen W. White, et al. Evaluating advanced search interfaces using established information-seeking models. *Journal of the American Society for Information Science and Technology*, 60(7): 1407–1422, 2009. 14

[159] Biao Xiang, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, and Hang Li. Context-aware ranking in web search. In *Proceedings of the 33th international ACM SIGIR conference on Research and development in information retrieval*, 2010. 10

[160] Songhua Xu and Francis C M Lau. A New Visual Search Interface for Web Browsing. In *Proceedings of the 2009 International Conference on Web Search and Data Mining*, 2009. 16

[161] Xifeng Yan and Jiawei Han. gSpan: Graph-based substructure pattern mining. In *Proceedings of the Second IEEE International Conference on Data Mining (ICDM'02)*, pages 721–724, 2002. 12

[162] Xifeng Yan, Hong Cheng, Jiawei Han, and Philip S. Yu. Mining significant graph patterns by leap search. In *SIGMOD*, pages 433–444, 2008. 12

[163] Jaewon Yang and Jure Leskovec. Patterns of temporal variation in online media. In *Proceedings of the 2011 International Conference on Web Search and Data Mining*, pages 177–186. ACM, 2011. 12

[164] Ka-Ping Yee, Danyel Fisher, Rachna Dhamija, and Marti Hearst. Animated exploration of dynamic graphs with radial layout. In *Presented at IEEE Symposium on Information Visualization*, 2001. 17

[165] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408. ACM, 2003. 131

[166] Tom Yeh, Kristen Grauman, Konrad Tollmar, and Trevor Darrell. A picture is worth a thousand keywords: image-based object search on a mobile platform. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 2025–2028. ACM, 2005. 16

[167] Alexander Ypma and Tom Heskes. Automatic categorization of web pages and user clustering with mixtures of hidden markov models. In *WEBKDD 2002 - Mining Web Data for Discovering Usage Patterns and Profiles*, pages 35–49. Springer, 2003. 14

[168] Eric Zavesky, Shih-Fu Chang, and Cheng-Chih Yang. Visual islands: intuitive browsing of visual search results. *Electrical Engineering*, pages 617–626, 2008. 16

Appendix A

Categorizations

A.1. List of FlickrBrowsing Source Categories

Table A.1 shows the URL categories used to categorize URLs that do not belong to Flickr in the FlickrBrowsing dataset.

Category	Examples
search	search.yahoo.com, google.com
social	facebook.com, tumblr.com
mail	mail.yahoo.com, gmail.com
aggregator	reddit.com, stumbleupon.com
blog	blogspot.com, blogger.com
photo	flickrhivemind.net, compfight.com
microblog	twitter.com
forum	discussion forums
news	news.yahoo.com, cnn.com
shop	ebay.com, amazon.com
video	youtube.com, vimeo.com
geo	maps.google.com, maps.yahoo.com
wiki	wikipedia.org, wikimedia.org
sport	sports.yahoo.com
autos	autos.yahoo.com

Table A.1: 15 URL categories in the FlickrBrowsing dataset.

A.2. List of `FlickrBrowsing` Page Layouts

We list below and in the next pages (Table A.2) all page layouts of the `FlickrBrowsing` dataset. There is a total of 96 layouts. The italicized parts of the URLs stand for the identifiers: *group-id* for groups, *user-id* for users, *photo-id* for photos, and *set-id* for albums.

URL in Flickr	Name
`/`	Homepage
`/about`	About Flickr
`/abuse`	Report Abuse
`/account`	Your Account
`/activity`	Recent Activity: All activity
`/analog`	Explore Analog
`/apps`	Your Apps
`/bestpractices`	Best Practices for Organizations
`/cameras`	Camera Finder
`/commons`	The Commons
`/configurator`	Flickr Configurator
`/confirm`	Confirmation page
`/creativecommons`	Creative Commons
`/do`	Monkey see? Monkey do!
`/do/more`	Monkey see? Monkey do!
`/explore`	Explore
`/explore/interesting`	Explore interesting photos
`/galleries`	Explore Galleries
`/gettyimages`	Getty
`/gift`	Flickr gift
`/gp`	Flickr guestpass
`/groups`	Groups
/groups/*group-id*	Group page
/groups/*group-id*/admin	Group administration
/groups/*group-id*/discuss	Group discussion
/groups/*group-id*/members	Group members
/groups/*group-id*/pool	Group photos

URL in Flickr	Name
/groups/*group-id*/pool/map	Group photo map
/groups/*group-id*/pool/tags	Group tags
/groups/*group-id*/pool/with	People appearing in group's photos
/groups/*group-id*/rules	Group rules
/groups_create.gne	Create group
/groups_invite.gne	Invite to group
/groups_join.gne	Join group
/groups_leave.gne	Leave group
/guidelines	Flickr Community Guidelines
/help	Help
/iconbuilder	The Icon Builder
/import	Find your friends
/import/people	Find your friends
/invite	Invite your friends
/logout.gne	Log out
/logout_ok.gne	Log out
/mail	Flickr Mail: Your Inbox
/mail/contact_notifications	Flickr Mail: Contact Notifications
/mail/reply	Flickr Mail: Reply
/mail/sent	Flickr Mail: Your Sent Mail
/mail/write	Flickr Mail: Compose a Message
/map	Explore Anyones' photos on a Map
/nearby	Everyone's photos taken near you
/partners/getty	Getty
/photo_delete.gne	Delete photo
/photo_edit.gne	Edit photo
/photos	Explore
/photos/friends	From the people you know

URL in Flickr	Name
/photos/organize	Organize your photos
/photos/tags	Popular tags on Flickr
/photos/upload	Upload a photo
/photos/upload/basic	Upload a photo
/photos/*user-id*	Display all user photos
/photos/*user-id*/alltags	Display all user tags
/photos/*user-id*/archives	Display all user photos in cronological order
/photos/*user-id*/collections	View user albums
/photos/*user-id*/favorites	Display all user favorites
/photos/*user-id*/galleries	View user albums
/photos/*user-id*/map	Explore user photos on a map
/photos/*user-id*/page	Display all user photos
/photos/*user-id*/people	People featured in user photos
/photos/*user-id*/popular	Popular user photos
/photos/*user-id*/sets	View user albums
/photos/*user-id*/sets/*set-id*	Display all album photos
/photos/*user-id*/show	Display single photo
/photos/*user-id*/stats	User statistics
/photos/*user-id*/tags	User tags
/photos/*user-id*/upload	Upload a photo
/photos/*user-id*/with	People appearing in user's photos
/photos/*user-id*/*photo-id*	Display single photo
/photos/*user-id*/*photo-id*/favorites	People who favorited the photo
/photos/*user-id*/*photo-id*/in/contacts	Browse contacts photos
/photos/*user-id*/*photo-id*/in/faves-*user-id*	Browse user favorites
/photos/*user-id*/*photo-id*/in/photostream	Browse user photos
/photos/*user-id*/*photo-id*/in/pool-*group-id*	Browse group photos
/photos/*user-id*/*photo-id*/in/set-*set-id*	Browse user album
/photos/*user-id*/*photo-id*/meta	Photo metadata

URL in Flickr	Name
`/photos/`*`user-id`*`/`*`photo-id`*`/sizes`	Photo in different resolutions
`/photosets_deletecomment.gne`	Delete comment
`/photosets_editcomment.gne`	Edit comment
`/photosof`	People you follow
`/places`	Explore places
`/profile_delete.gne`	Delete profile
`/search`	Search photos
`/search/advanced`	Search photos
`/search/forum`	Search forum
`/search/groups`	Search groups
`/search/people`	Search people
`/search/show`	Search photos
`/services/api`	Flick API
`/services/apps`	Flick Apps
`/services/auth`	Authentication
`/services/developer`	The Flickr Developer Guide
`/services/feeds`	Flickr photo feed
`/services/oauth`	O-auth authentication
`/services/partners`	Flick parterns
`/signin`	Sign in
`/signup`	Sign up
`/tools`	Tools to upload and share photos
`/tour`	Flickr tour
`/upgrade`	Upgrade account
`/welcome`	Welcome to Flickr

Table A.2: List of page layouts in Flickr. The table shows the URL inside Flickr and the description of the layout.