About *SIGKDD Explorations*

Explorations is published twice yearly, in June/July and December/January each year. After the first two volumes, frequency may increase to quarterly. The newsletter is distributed in hardcopy form to all members of the ACM SIGKDD. It is also sent to ACM's network of libraries. Additionally, issues are published on the web and are free to the general public (http://www.acm.org/sigkdd/explorations/).

Our goal is to make *SIGKDD Explorations* an informative, rapid means of publication and a dynamic forum for communication with the Knowledge Discovery and Data Mining community. SIGKDD membership is growing at a very fast pace, and with KDD being a multi-disciplinary field, we hope that *Explorations* will facilitate its fusion and enhance the sense of community. Submissions will be reviewed by the editor and/or associate and guest editors as appropriate. We are particularly interested in short research and survey articles on various aspects of data mining and KDD. *Explorations* is also a forum for publishing position papers, controversial positions, challenges to the community, product reviews, book reviews, news items and other items of interest to the field. Please see:

http://www.acm.org/sigkdd/explorations/instructions.htm

Advertiser Information:

Explorations accepts advertisements related to data mining and KDD, including company, book, vendor, and service advertisements. For rates and instructions on submitting an ad, please see:

http://www.acm.org/sigkdd/explorations/instructions.htm#advertise

Notice to Contributing Authors of SIGKDD Explorations:

By submitting your article for distribution in this Special Interest Group publication, you hereby grant to ACM the following non-exclusive, perpetual, worldwide rights:

- to publish in print on condition of acceptance by the editor
- to digitize and post your article in the electronic version of this publication
- to include the article in the ACM Digital Library
- to allow users to copy and distribute the article for noncommercial, educational or research purposes.

However, as a contributing author, you retain the copyright to your article and ACM will make every effort to refer requests for commercial use directly to you.

Notice to Past Authors of ACM-Published Articles:

ACM intends to create a complete electronic archive of all articles and/or other material previously published by ACM. If you have written work that was previously published by ACM in any journal or conference proceedings prior to 1978, or any SIG newsletter at any time, and you do NOT want this work to appear in the ACM Digital Library, please inform you respective editors and permissions@acm.org, stating the title of the work, the author(s), and where and when published.

Patent Mining: A Survey

Longhui Zhang
lzhan015@cs.fiu.edu

Lei Li
lli003@cs.fiu.edu

Tao Li
taoli@cs.fiu.edu

School of Computing and Information Sciences
Florida International University
Miami, FL 33199

ABSTRACT

Patent documents are important intellectual resources of protecting interests of individuals, organizations and companies. Different from general web documents, patent documents have a well-defined format including frontpage, description, claims, and figures. However, they are lengthy and rich in technical terms, which requires enormous human efforts for analysis. Hence, a new research area, called patent mining, emerges in recent years, aiming to assist patent analysts in investigating, processing, and analyzing patent documents. Despite the recent advances in patent mining, it is still far from being well explored in research communities. To help patent analysts and interested readers obtain a big picture of patent mining, we thus provide a systematic summary of existing research efforts along this direction. In this survey, we first present an overview of the technical trend in patent mining. We then investigate multiple research questions related to patent documents, including patent retrieval, patent classification, and patent visualization, and provide summaries and highlights for each question by delving into the corresponding research efforts.

Keywords

Patent Mining; Patent Information Retrieval; Patent Classification; Patent Visualization; Patent Valuation; Cross-Language Patent Mining; Patent Application

1. INTRODUCTION

Patent application is one of the key aspects of protecting intellectual properties. In the past decades, with the advanced development of various techniques in different application domains, a myriad of patent documents are filed and be approved. They serve as one of the important intellectual property components for individuals, organizations and companies. These patent documents are open to public and made available by various authorities in a lot of countries or regions around the world. For example, World Intellectual Property Organization (WIPO)[1] reported 1.98 million total patent applications filed worldwide in 2010.

Patent documents have great research values, beneficial to the industry, business, law, and policy-making communities. If patent documents are carefully analyzed, important technical details and relations can be revealed, leading business trends can be illustrated, novel industrial solutions can be inspired, and consequently vital investment decisions can be made [15]. Thus, it is imperative to carefully analyze patent documents for evaluating and maintaining patent values. In recent years, patent analysis has been recognized as an important task at the government level. Public patent authorities[2] in United States, United Kingdom, China and Japan have invested various resources to improve the performances of creating valuable patent analysis results for various patent analysis tasks.

However, patent analysis is a non-trivial task, which often requires tremendous amount of human efforts. In general, it is necessary for patent analysts to have a certain degree of expertise in different research domains, including information retrieval, data mining, domain-specific technologies, and business intelligence. In reality, it is difficult to find and train such analysts to match those multi-disciplinary requirements within a relatively short period of time. Another challenge of patent analysis is that patent documents are often lengthy, and full of technical and legal terminologies. Even for domain experts, it may also require a lot of time to read and analyze a single patent document. Therefore, patent mining plays an important role in automatically processing and analyzing patent documents [106; 127].

A patent document often contains dozens of items that can be grouped into two categories: (1) structured items, which are uniform in semantics and format (such as patent number, inventor, filing date, issued date, and assignees); and (2) unstructured items, which consist of text content in different length (including claims, abstracts, and descriptions of the invention.). Given such a well-defined structure, patent documents are considerably different from general web documents (e.g., web pages), most of which contain unstructured data, involving free texts, links, tags, images, and videos. Hence, the analysis of patent documents might be different from the one for web documents in terms of the format and various application-wise purposes.

In this survey, we comprehensively investigate multiple critical research questions in the domain of patent mining, including (1) how to effectively retrieve patent documents based on user-defined queries (See Section 3)? (2) how to efficiently perform patent classification for high-quality maintenance (See Section 4)? (3) how to informatively represent patent documents to users (See Section 5)? (4) how to explore and evaluate the potential benefit of patent documents (See Section 6)? and (5) how to effectively deal with cross-language patent documents (See Section 7)? For

[1] http://www.wipo.int/ipstats/en/general_info.html.

[2] http://www.wipo.int/directory/en/urls.jsp.

Table 1: Representative patent mining tasks and approaches.

Tasks	Techniques	References
Patent Retrieval (See Section 3)	Query Generation	[6; 7; 10; 17; 50; 76; 78; 79; 104; 108; 114; 118; 119]
	Query Expansion	[2; 9; 25; 28; 29; 30; 31; 34; 40; 43; 52; 68] [69; 72; 73; 74; 78; 83; 96; 98; 99; 107; 114]
Patent Classification (See Section 4)	Using Different Resources	[4; 33; 49; 56; 58; 59; 66; 86; 101]
	Using Different Classifier	[13; 19; 23; 24; 33; 103; 116]
Patent Visualization (See Section 5)	Structured Data Visualization	[42; 93; 97; 120; 122; 123]
	Unstructured Text Visualization	[5; 39; 61; 105; 124]
	Hybrid Visualization	[16; 51; 63; 80; 94; 97; 121; 124]
Patent Valuation (See Section 6)	Unsupervised Exploration	[3; 21; 45; 46; 60; 67; 81; 84; 109; 111]
	Supervised Evaluation	[21; 41; 46; 67; 85; 109]
Cross-Language Mining (See Section 7)	Machine Translation	[18; 26; 27; 32; 48; 53; 70; 77]
	Semantic Correspondence	[54; 62; 64; 47; 102; 110]

each question, we first identify several critical research challenges, and then discuss different research efforts and various techniques used for addressing these challenges. Table 1 summarizes different patent mining tasks, including patent retrieval, patent classification, patent visualization, patent exploration, and cross-language patent mining. Up-to-date references/lists related to patent mining can be found at http://users.cis.fiu.edu/∼lzhan015/patmining.html. In the following sections, we will briefly introduce the existing solutions to each task based on the techniques being utilized. The rest of the paper is organized as follows. In§ 2, we provide an introduction to patent documents by describing patent document structures, patent classification systems, and various patent mining tasks. Section 3 presents a summary of research efforts for addressing patent retrieval, especially, patent search. In Section 4, we investigate how patent documents can be automatically classified into different predefined categories. In Section 5, we explore how patent documents can be represented to analysts in a way that the core ideas of patents can be clearly illustrated and the correlations of different documents can be easily identified. In Section 6, we show that the quality of a patent document can be automatically evaluated based on some predefined measurements that help companies decide which patent is more important and should be further maintained for effective property protection. In Section 7, we present different techniques for cross-language patent mining, including approaches to solving machine translation and semantic correspondence. Section 8 discusses existing free and commercial patent mining systems that provide various functionalities to allow patent analysts to perform different patent mining tasks. Finally, Section 9 concludes our survey and discusses emerging research- and application-wise challenges in the domain of patent mining.

2. BACKGROUND

In this section, we first provide a brief overview of patent documents and their structure, and then describe the current patent classification systems, followed by introducing the tasks in the entire process of patent application.

2.1 The Structure of Patent Documents

According to World Intellectual Property Organization[3], the definition of a patent is: "*patents are legal documents issued by a government that grants a set of rights of exclusivity and protection to the owner of an invention. The right of exclusivity allows the patent owner to exclude others from making, using, selling, offering for sale, or importing the patented invention during the patent term, typically period from the earliest filing date, and in the country or countries where patent protection exists.*" Based upon the understanding of the definition, patent documents are one of the key components that serve to protect the intellectual properties of patent owners. Note that patents and inventions are two different yet interleaved concepts: patents are legal documents, whereas inventions are the content of patents. Different countries or regions may have their own patent laws and regulations, but in general there are two common types of patent documents: utility patents and design patents. Utility patents describe technical solutions related to a product, a process, or a useful improvement, etc., whereas design patents often represent original designs related to the specifications of a product. In practice, due to the distinct properties of these two types of patents, the structure of patent document may vary slightly; however, a typical patent document often contains several requisite sections, including a front page, detailed specifications, claims, declaration, and/or a list of drawings to illustrate the idea of the solution.

Figure 1 shows an example of the front page of a patent document. In general, a `frontpage` contains four parts, described as follows:

1. `Announcement`, which includes Authority Name (e.g. United States Patent), Patent No., and Date of Patent (i.e., patent publication date).;

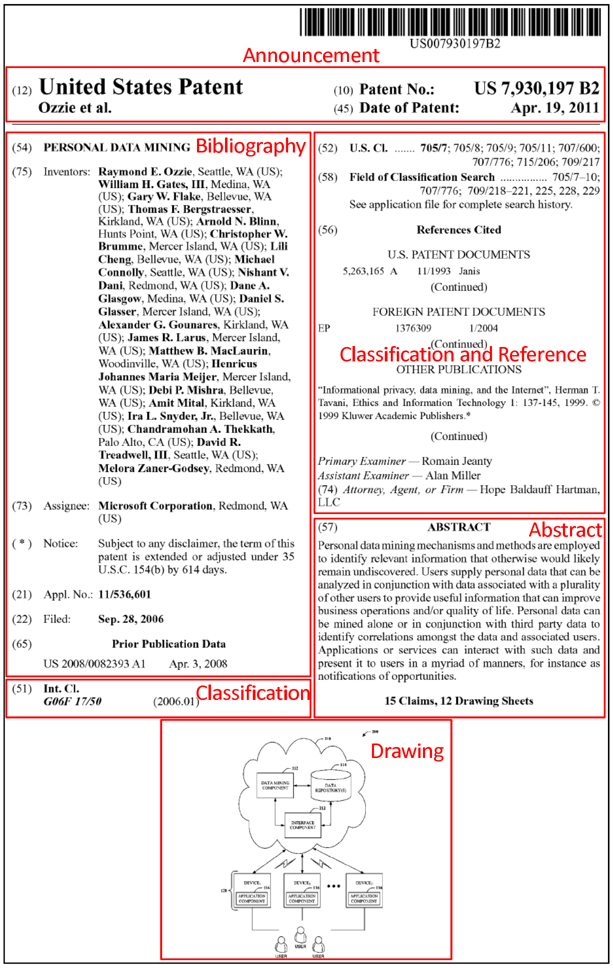
2. `Bibliography`, which often includes Title, Inventors, Assignee, Application No., and Date of filing.;

3. `Classification` and `Reference`, which include International Patent Classification Code, Region-based Classification Code (e.g., United State Classification Code), and/or other patent classification categories, along with references assigned by the examiner;

4. `Abstract`, which may contain a short description of the invention and sometimes a drawing that is the most representative one in terms of illustrating the general idea of the invention.

Beside the front page, a patent document contains detailed description of the solution, claims, and/or a list of draw-

[3] http://www.wipo.int.


Announcement
Bibliography
Classification and Reference
Abstract
Classification
Drawing
200
210
112
114
212
DATA MINING COMPONENT
DATA REPOSITORY(S)
INTERFACE COMPONENT
120
DEVICE₁
DEVICE₂
DEVICE_N
116
APPLICATION COMPONENT
USER

US007930197B2

(12) **United States Patent**
Ozzie et al.

(10) **Patent No.:** **US 7,930,197 B2**
(45) **Date of Patent:** **Apr. 19, 2011**

(54) **PERSONAL DATA MINING**

(75) Inventors: **Raymond E. Ozzie**, Seattle, WA (US); **William H. Gates, III**, Medina, WA (US); **Gary W. Flake**, Bellevue, WA (US); **Thomas F. Bergstraesser**, Kirkland, WA (US); **Arnold N. Blinn**, Hunts Point, WA (US); **Christopher W. Brumme**, Mercer Island, WA (US); **Lili Cheng**, Bellevue, WA (US); **Michael Connolly**, Seattle, WA (US); **Nishant V. Dani**, Redmond, WA (US); **Dane A. Glasgow**, Medina, WA (US); **Daniel S. Glasser**, Mercer Island, WA (US); **Alexander G. Gounares**, Kirkland, WA (US); **James R. Larus**, Mercer Island, WA (US); **Matthew B. MacLaurin**, Woodinville, WA (US); **Henricus Johannes Maria Meijer**, Mercer Island, WA (US); **Debi P. Mishra**, Bellevue, WA (US); **Amit Mital**, Kirkland, WA (US); **Ira L. Snyder, Jr.**, Bellevue, WA (US); **Chandramohan A. Thekkath**, Palo Alto, CA (US); **David R. Treadwell, III**, Seattle, WA (US); **Melora Zaner-Godsey**, Redmond, WA (US)

(73) Assignee: **Microsoft Corporation**, Redmond, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 614 days.

(21) Appl. No.: **11/536,601**

(22) Filed: **Sep. 28, 2006**

(65) **Prior Publication Data**

US 2008/0082393 A1 Apr. 3, 2008

(51) **Int. Cl.**
G06F 17/50 (2006.01)

(52) **U.S. Cl.** **705/7**; 705/8; 705/9; 705/11; 707/600; 707/776; 715/206; 709/217

(58) **Field of Classification Search** 705/7–10; 707/776; 709/218–221, 225, 228, 229
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,263,165 A	11/1993	Janis

(Continued)

FOREIGN PATENT DOCUMENTS

EP	1376309	1/2004

(Continued)

OTHER PUBLICATIONS

"Informational privacy, data mining, and the Internet", Herman T. Tavani, Ethics and Information Technology 1: 137-145, 1999. © 1999 Kluwer Academic Publishers.*

(Continued)

Primary Examiner — Romain Jeanty
Assistant Examiner — Alan Miller
(74) *Attorney, Agent, or Firm* — Hope Baldauff Hartman, LLC

(57) **ABSTRACT**

Personal data mining mechanisms and methods are employed to identify relevant information that otherwise would likely remain undiscovered. Users supply personal data that can be analyzed in conjunction with data associated with a plurality of other users to provide useful information that can improve business operations and/or quality of life. Personal data can be mined alone or in conjunction with third party data to identify correlations amongst the data and associated users. Applications or services can interact with such data and present it to users in a myriad of manners, for instance as notifications of opportunities.

15 Claims, 12 Drawing Sheets

Figure 1: Front page of a patent document.

ings. The `description` section, in general, depicts the background and summary of the invention, brief description of the drawings, and detailed description of preferred embodiments. The `claim` section is the primary component of a patent document, which defines the scope of protection conveyed by the invention. It often contains two types of claims: (1) the independent claim which stands on itself; and (2) the dependent claims which refer to its antecedent claim.

A patent document is often lengthy, compared with other types of documents, e.g., web pages. Although the structure of a patent document is well-defined, a myriad of obscure and ambiguous text snippets are often involved, and various technical terms are often used in the content, which render the analysis of patent document more difficult.

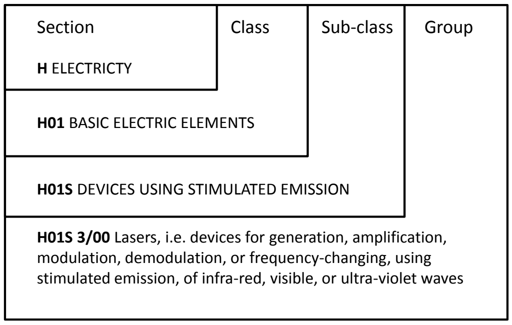
2.2 Patent Classification Criteria

Before the publication of patent applications, one or more classification codes are often assigned to patent documents based on their textual contents for the purpose of efficient management and retrieval. Different patent authorities may maintain their own classification hierarchies, such as the United States Patent Classification (USPC) in the United States, the International Patent Classification (IPC) for the World Intellectual Property Organization, and the Derwent classification system fixed by Thomson Reuters. In the following, we will introduce the classification taxonomies of IPC and USPC in more details.

2.2.1 *IPC Taxonomy*

IPC was established in 1971 based on Patent Cooperation Treaty [22]. This hierarchical patent classification system categorizes patents to different technological groups. There are over 100 countries using IPC system to classify their national patent applications. Specifically, the IPC category taxonomy contains 8 sections, 120 classes, 630 subclasses, 7,200 main groups and approximately 70,000 sub-groups. A typical IPC category contains a class label and a piece of text description to indicate the specific category content.

In IPC, all technological fields are first grouped into 8 sections represented by one of the capital letters from A to H[4], including (A) "Human necessities"; (B) "Performing operations, transporting"; (C) "Chemistry, metallurgy"; (D) "Textiles, paper"; (E) "Fixed constructions"; (F) "Mechanical engineering, lighting, heating, weapons, blasting"; (G) "Physics"; and (H) "Electricity". Then, within each section, the technological fields are regrouped into classes as the second level of the IPC taxonomy. Each class consists of one or more subclasses, which are treated as the third level of the taxonomy. Finally, each subclass is further divided into subdivisions referred to as "groups". As an illustrative example, Figure 2 describes the class label "H01S 3/00" and its ancestors.



Section

H ELECTRICTY

Class

H01 BASIC ELECTRIC ELEMENTS

Sub-class

H01S DEVICES USING STIMULATED EMISSION

Group

H01S 3/00 Lasers, i.e. devices for generation, amplification, modulation, demodulation, or frequency-changing, using stimulated emission, of infra-red, visible, or ultra-violet waves

Figure 2: An example of IPC.

2.2.2 *USPC Taxonomy*

The USPC system was developed in 1836, which is the first patent taxonomy established in the world [88]. In USPC, the patent categories are organized as a two-level taxonomy, i.e., class and subclass. Each class has a designated class number, and includes a descriptive title, class schedule, and definitions. Then each class is subdivided into a number of subclasses. A subclass has a number, a title, an indent level indicated by one or more dots, a definition, a hierarchical relationship to other subclasses in a class, and relationships to other subclasses in other classes. A subclass is the smallest searchable group of patents in USPC.

2.3 Tasks in Patent Analysis and Investigation

Based upon the filing status of a patent document, a patent mining system can be decomposed into two modules: (1) *Pre-filing* module, in which the patent documents are carefully examined to ensure the non-infringement; and (2) *Post-*

[4] http://www.wipo.int/classifications/ipc/en.

filing module, in which patent documents are maintained and analyzed. The general architecture of a patent mining system is depicted in Figure 3.

During the *pre-filing* process, or say, the application process, there are two major tasks:

1. Classifying the patent application into multiple predefined categories (e.g., IPC and USPC). This task aims to not only restrict the searching scope, but also ease the maintenance of patent applications/documents.

2. Searching all relevance patent documents from patent databases and non-patent documents from online resources. The primary goal of this task is to examine the infringement/patentability, and assigning a list of appropriate references for better understanding the idea of the patent application.

Currently in most intellectual property authorities and/or patent law firms, these two tasks are often being conducted manually. In practice, these two tasks, especially the latter one, may require specific domain expertise and a huge amount of time/human efforts.

The major focus of the *post-filing* process is to maintain and analyze patent documents in order to provide fully functional support to various types of enterprises. For example, a company plans to develop a new product. Prior to the design/implementation of this product, it is essential to determine what related products have already been produced and patented. Therefore, a typical task is to perform a comprehensive investigation towards the related domain/products by virtue of patent search. By doing this, the company is able to obtain an overview of the general technologies applied in the corresponding domain, as well as the technical details of relevant products. In general, in the process of *post-filing*, besides the task of patent search, three additional tasks are often involved:

1. Patent visualization, which aims to represent patent documents to help patent analysts easily understand the core idea of patents;

2. Patent valuation, which explores patent documents in different ways to evaluate their value, potential, impact, etc.;

3. Cross-language mining, which localizes patent information from patent documents that are described by multiple languages.

However, due to the large volume of patent files and diverse writing styles of patent applications, these processes are time-consuming, and often require a lot of human efforts for patent reading and analysis. The ultimate goal of these efforts is to provide automatic tools to ease the procedure of patent analysis. In the following sections, we will introduce the existing academic/industrial efforts in designing patent mining algorithms and building patent mining applications using the architecture shown in Figure 3.

3. PATENT RETRIEVAL

Patent retrieval is a subdomain of information retrieval, in which the basic elements to search are patent documents. Due to the characteristics of patent documents and special requirements of patent retrieval, patent search is quite different from searching general web documents. For example, queries in patent search are generally much longer and more complex than the ones in web search.

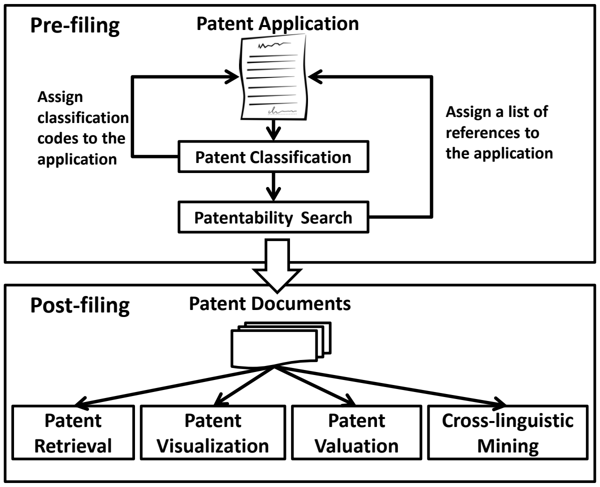


Figure 3: The architecture of a patent mining system.

With the domain-specific requirement of patent retrieval, patent search has gained great attention in the last decade in both academia and industry. Currently, there are numerous benchmark collections of patent documents available in information retrieval community, and several workshops and symposiums on patent retrieval have been organized, including NTCIR[5], CLEF[6] and TREC[7]. In 2003, the third NTCIR workshop [44] firstly provided benchmark collections of patent documents for enhancing research on patent information processing. They assigned the "Patent Retrieval Task" to explore the effect of retrieving patent documents in real-world applications. The recent advancement in patent search is driven by the "Intellectual Property" task initialized by CLEF [87]. Several teams participated in the prior-art search task of the CLEF-IP 2010 and proposed approaches to reduce the number of returned patent documents by extracting a set of key terms and expanding queries for broader coverage.

Table 2: Challenges in patent retrieval.

Challenges	Reasons
Low Readability	People may use rhetorical structures and ambiguous terms to defend their invention in order to obtain broader protection.
Lengthy Query	People often use the whole patent document as a query to perform searching.
High Recall	Missing one strongly relevant document in patent retrieval is unacceptable because of the tremendous cost of patent lawsuit.

Despite the recent advances, the task of patent retrieval remains challenging from multiple perspectives. We summa-

[5] http://research.nii.ac.jp/ntcir/index-en.html.
[6] http://ifs.tuwien.ac.at/∼clef-ip.
[7] http://trec.nist.gov.

rize several challenges related to patent retrieval as listed in Table 2. In the following, we first introduce various types of patent search tasks in Section 3.1, and then discuss existing solutions/approaches to the aforementioned challenges. A summary of patent retrieval techniques is depicted in Figure 4. Specifically, in Section 3.2 we discuss how to improve the readability of patent documents; in Section 3.3 we introduce existing methods that assist patent examiners in generating query keywords; and in Section 3.4 we describe the techniques to expand the query keyword set.

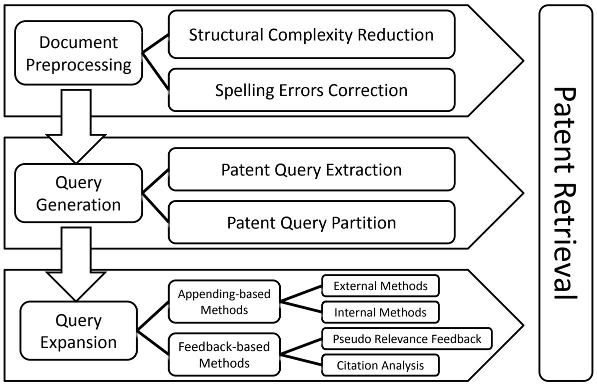


Figure 4: A summary of patent retrieval techniques.

3.1 Patent Search and a Typical Scenario

In practice, there are five representative patent search tasks listed as follows:

- *Prior-Art Search*, which aims at understanding the state-of-the-art of a general topic or a targeted technology. It is often referred to as patent landscaping or technology survey. The scope of this task mainly focuses on all the available publications[8] worldwide.

- *Patentability Search*, which tries to retrieve relevant documents worldwide that have been published prior to the application date, and may disclose the core concept in the invention. This task is often performed before/after patent application.

- *Invalidity Search*, which searches the available publications that invalidate a published patent document. This task is usually performed after a patent is granted.

- *Infringement Search*, which retrieves valid patent publications that are infringed by a given product or patent document. In general, the search operates on the claim section of the available patent documents.

- *Legal Status Search*, which determines whether an invention has freedom to make, use, and sell; that is, whether the granted patent has lapsed or not.

In Figure 5, we provide an overview of the procedure to perform patent search tasks. As depicted, it contains 4 major steps:

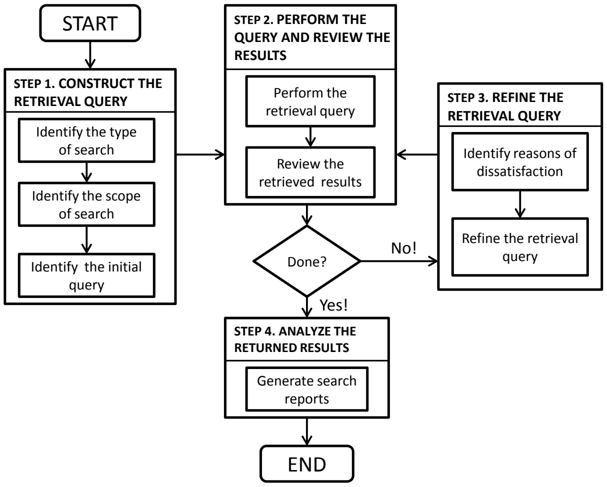

START
STEP 1. CONSTRUCT THE RETRIEVAL QUERY
Identify the type of search
Identify the scope of search
Identify the initial query
STEP 2. PERFORM THE QUERY AND REVIEW THE RESULTS
Perform the retrieval query
Review the retrieved results
Done?
No!
Yes!
STEP 3. REFINE THE RETRIEVAL QUERY
Identify reasons of dissatisfaction
Refine the retrieval query
STEP 4. ANALYZE THE RETURNED RESULTS
Generate search reports
END

Figure 5: A typical procedure of patent search.

Step 1 Construct the retrieval query:
An initial action is to determine the type of patent search task (as aforementioned) based on the purpose of patent retrieval. Then, the search scope can be identified accordingly. For example, patentability search is to retrieve relevant documents that are published prior to the filing/application date, and therefore the scope of patentability search contains all the available documents worldwide. Finally, we need to construct the initial retrieval query based on the user's information need, as well as the type of the task. For example, in the task of invalidity search, both the core invention and the classification code of the patent document need to be identified.

Step 2 Perform the query and review the results:
Queries are executed in the scope of the task identified in **Step 1**, and relevant documents are returned to the user. Then the user will review the returned results to determine whether the documents are desired. If so, go to **Step 4**; otherwise, go to **Step 3**.

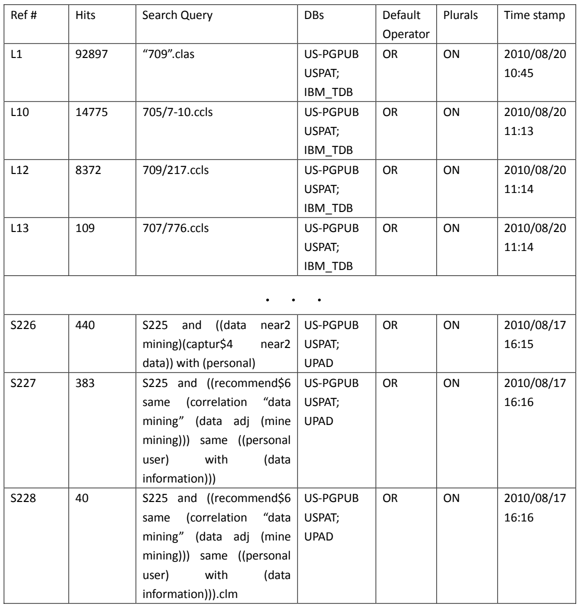
Step 3 Refine the retrieval query:
If the returned results in **Step 2** are not satisfactory (e.g., too many documents, too few results, or many irrelevant results), we need to refine search queries in order to improve the search results. For example, we can put more constrains (hyponyms) in the query if we want to reduce the number of returned documents, or remove several constrains (hyponyms) if we get too few results, or replace the query with new keywords if the results are irrelevant.

Step 4 Analyze the returned results:
After a user reviews each returned document, he/she will write a search report based on the search task in accordance with the patent law and regulation. The search report, in general, consists of: (1) a summary of the invention; (2) classification codes; (3) databases or retrieval tools used for search; (4) relevant documents; (5) query logs; and (6) retrieval conclusions.

We take patentability search as an illustrative example to further explain the search procedure. Suppose a patent ex-

[8] Here the publications are public literatures, including patent documents and scientific papers.

aminer tries to perform the patentability search for a patent application related to "Personal Data Mining". In Step 1, he/she will read the application file and extract keywords such as "data mining", "capture data", and "correlation connection link", and generate the search query based on these keywords. Then he/she will perform the search query within a series of patent databases, such as USPAT and IBM_TDB, and iteratively refine the query according to the search results in Step 2 and 3. Finally, he/she will read all 40 "hits" (the returned documents) to find a list of relevant documents and write a search report in Step 4. Figure 6 shows a query log of this example[9].



Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time stamp
L1	92897	"709".clas	US-PGPUB USPAT; IBM_TDB	OR	ON	2010/08/20 10:45
L10	14775	705/7-10.ccls	US-PGPUB USPAT; IBM_TDB	OR	ON	2010/08/20 11:13
L12	8372	709/217.ccls	US-PGPUB USPAT; IBM_TDB	OR	ON	2010/08/20 11:14
L13	109	707/776.ccls	US-PGPUB USPAT; IBM_TDB	OR	ON	2010/08/20 11:14
. . .						
S226	440	S225 and ((data near2 mining)(captur$4 near2 data)) with (personal)	US-PGPUB USPAT; UPAD	OR	ON	2010/08/17 16:15
S227	383	S225 and ((recommend$6 same (correlation "data mining" (data adj (mine mining))) same ((personal user) with (data information)))	US-PGPUB USPAT; UPAD	OR	ON	2010/08/17 16:16
S228	40	S225 and ((recommend$6 same (correlation "data mining" (data adj (mine mining))) same ((personal user) with (data information))).clm	US-PGPUB USPAT; UPAD	OR	ON	2010/08/17 16:16

Figure 6: A sample query log of patent search.

3.2 Patent Document Preprocessing

In Section 2.1, we have introduced the typical structure of patent documents. Besides the structured content in the front page, a patent document, in practice, often contains a large amount of unstructured textual information. In order to ensure the patentability of patent documents and maximize the scope of the protection, patent attorneys or inventors, in general, use complex sentences with domain-specific words to describe the invention, which renders patent documents difficult to understand or read, even for domain experts. This phenomenon is more common in the claims, which is the most important part of a patent document, as claims often define the implementation of essential components of the patent invention. In order to help users quickly grasp the core idea of a patent document, and consequently improve the efficiency of patent retrieval, it is imperative to refine the readability of patent documents.

A patent document often involves complex structure and/or lexicon. To ease the understanding of patent document, researchers usually try to reduce both *structural complexity* and *lexical complexity* using techniques of information retrieval, data mining, natural language processing, etc. For example, in [91], Shinmori et al. utilize nature language processing methods to reduce the structural complexity. They predefine six relationships (procedure, component, elaboration, feature, precondition, composition) to capture the structure information of Japanese patent claims. In addition, they use cue-phrase-based approaches to extract both cue phrase tokens and morpheme tokens, and then employ them to create a structure tree to represent the first independent claim. Their experimental results on NTCIR3 patent data collection indicate that the proposed tree-based approach can achieve better performance in terms of accuracy. In contrast, Sheremetyeva [90] proposes the similar approach to capture both the structure and lexical content of claims from US patent documents. The author decomposes the long claim sentences into short segments, and then analyzes the dependence relations among them. After that, a tree-basd representation is provided to capture both content and structure information of claims, and consequently the readability of the patent documents is improved.

Besides the complexity, patent documents often contain some spelling errors. Stein et al. [92] indicate that many patents from USPTO contain the spelling errors, e.g., "Samsung Inc" may be written as "Sumsung Inc". Such errors may increase the inconsistency of the patent corpus and hence may deteriorate the readability of patent documents. Thus, they provide an error detection approach to identify the spelling errors in the field of patent assignee (e.g., company name). The experiments have shown that both precision and recall can be improved after they correct the spell errors.

3.3 Patent Query Generation

In general, users may specify only several keywords in adhoc web search. Most web-based search systems have the restriction on the length of the input query, e.g., the maximum number of query keywords in Google search engine is 32. One possible reason is that the retrieval response time of search engines increases along with the length of the input. Comparatively in patent retrieval systems, a patent query often consists of tens or even hundreds of keywords on average. A common practice of generating such a query is to manually extract representative terms from original patent documents or add additional technological terms. This is often achieved by patent examiners, which requires a tremendous amount of time and human efforts. Also, patent examiners are expected to have strong technological background in order to provide a concise yet precise query. To assist patent examiners in generating patent queries, a lot of research work has been proposed in the last decade. In general, there are two automatic ways to produce a patent query, i.e., *query extraction* and *query partition*.

3.3.1 Query Extraction

Query extraction aims to extract representative information from an invention that describes the core idea of the invention. The simplest way of query extraction is to extract the abstract which is the summary of the invention given by the patent applicant, or the independent claims which define the scope of the protection. However, the extracted information based on abstracts or claims may not be suitable to form the patent query. The reason is straightforward: applicants often describe the abstract/claim without enough technical details in order to decrease the retrievability of their patent, and the terms in the abstract/claims often contain

[9] http://portal.uspto.gov/pair/PublicPair.

obscure meaning (e.g., "comprises" means "consists at least of") [106].

To alleviate this issue, Konishi [55] tries to expand the query by selecting terms from the explanative sentences in the description. As mentioned in Section 2, the description section of a patent document consists of the detailed information of the invention. Additional efforts along this direction involve [76; 119] that extract query terms from different sections of a patent document to automatically transform a patent file into a query. In [119], different weights are assigned to terms from different sections of patents. Their experiments on a USPTO patent collection indicate that using the terms from the description section can produce high-quality queries, and using the term frequency weighting scheme can achieve superior retrieval performance. In [76], a patent query is constructed by selecting the most representative terms from each section based on both log-likelihood weighting model and parsimonious language model [38]. While the authors only consider 4 sections, including title, abstract, description and claims, they draw the same conclusion that extracting terms from the description section of a patent document is the best way to generate queries. Mahdabi et al. [73] further propose to utilize the international patent code as an additional indicator to facilitate automatic query generation from the description section of patents.

In addition to extracting query terms from a single section [73; 76; 119], Konishi [55] exploits the combination of queries from multiple sections to build a query. The intuition is that the terms extracted from a single section is more cohesive from the ones from different sections, whereas the terms of multiple sections can help emphasize the differences between sections. Therefore, the generated queries from single sections can be treated as subqueries for searching patent documents. The experiments [55] demonstrate that the best retrieval performance could be achieved by combining the extracted terms from the abstract, claims, and description sections.

However, the aforementioned approaches require to assign weights to terms from different sections. In most cases, the weights of terms are difficult to obtain, and hence have to be heuristically assigned. To further improve the retrieval, Xue and Croft consider to employ additional features, including patent structural features, retrieval-score features, and the combinations of these features to construct a "learning-to-rank" model [118]. Their experiments on a USPTO patent collection demonstrate that the combination of terms and noun-phrases from the summary field can achieve the best retrieval performance.

3.3.2 Query Partition

An alternative way for query generation is to automatically partition the query document into multiple subtopics, and generate keywords based on each subtopic. Along this direction, several partition-based approaches have been proposed to improve the quality of patent queries. For example, Takaki et al. [95] partition the original query document into multiple subtopics, and then builds sub-queries to retrieval similar documents for each subtopic. A entropy-based "relevance score" of each subtopic is defined to determine relevance documents. However, this method involves extracting terms from the query document for each subtopic element, and hence the time complexity will increase along with the number of subtopics. Borgonovi et al. [11] present a similar approach to segment original query into subtopics. Instead of extracting terms form subtopics, they treat subtopics as sub-queries, and directly use them to execute the search and merge results obtained from each sub-query as the final result. Another approach [10] splits the original query document into multiple sentences, and then treats each sentence as an individual query to perform search. The top k relevant documents of each sub-query are merged as the final retrieval result. The empirical evaluation demonstrates that this approach is able to achieve reasonable retrieval performance, and also can significantly improve the running time compared with other baselines.

3.4 Patent Query Expansion

Patent search, as a recall-orientated search task, does not allow missing relevant patent documents due to the highly commercial value of patents and high costs of processing a patent application or patent infringement. Thus, it is important to retrieve all possible relevant documents rather than finding only a small subset of relevant patents from the top ranked results. To this end, a common practice is to enrich the query keywords in order to improve the keyword coverage, which is often referred to as *query expansion*. Recently, many query expansion techniques have been introduced in the field of patent search to improve the effectiveness of the retrieval. As discussed in [69; 78], the methods for tackling this problem can be categorized into two major groups: (1) *appending-based methods*, which either introduce similar terms or synonyms from patent document or external resources, or extract new terms from patent document to expand or reformulate a query; and (2) *feedback-based methods*, which modify the query based on the retrieved results, e.g. using pseudo relevance feedback or citation analysis.

3.4.1 Appending-Based Methods

Appending-based methods try to append additional terms to the original keyword set. In practice, the additional terms can be extracted from either the query document or the external resources, e.g., Wordnet and Wikipedia. Based on the information sources utilized by query expansion, this type of methods can be further decomposed into two groups: (1) methods that employ the query document as the expansion basis; and (2) methods that use external resources to expand the query.

Internal methods: This type of techniques exploits the query patent document itself as the resource to expand the original keyword set. The general process is to extract relevant or new terms that represent the major idea of the invention. A lot of query expansion approaches fall into this group. For example, Konishi [55] expands query terms by virtue of the "explanative sentences" extracted from the description section of the query patent, where the explanative sentences are obtained based on the longest common substring with respect to the original keyword set. In addition, several approaches [69; 99] use multi-language translation models to create a patent-related synonyms set (SynSet) from a CLEP-IP patent collection, and expand the original query based on SynSet. Parvaz et al. [73] introduce various features that can be used to estimate the importance of the noun-phrase queries. In their method, important noun-phrase queries are selected to reformulate original keyword set. These approaches are able to improve the retrieval performance; however, the improvement purely based on the extraction paradigm is quite marginal.

To further enhance the retrieval capability, semantic relations, e.g., the keyword dependencies, between query keywords are often explored. For example, Krishnan et al. [57] propose an approach to identifying the extracted treatment and causal relationships from medical patent documents. In [83], linguistic clues and word relations are exploited to identify important terms in patent documents. Based on the extracted relations between problems and solutions, the original query is reformulated. The evaluation shows that by considering the semantic relations of keywords, the retrieval performance can be improved to a great extent.

External methods: This type of techniques aims to utilize external resources, e.g., WordNet and Wikipedia, to expand original queries. WordNet is a large lexical database of English that groups different terms into sets of cognitive synonyms. It is often employed by researchers from the information retrieval community to enhance retrieval effectiveness. Recently, WordNet has been used to facilitate the process of query expansion in patent retrieval. For instance, Magdy and Jones [69] build a keyword-based synonym set with extracted synonyms and hyponyms from WordNet, and utilize this synonym set to improve the retrieval performance. However, in some cases it cannot obtain reasonable results due to the deficiency of contextual information. To solve this problem, Al-Shboul and Myaeng [2] introduce another external resource, i.e., Wikipedia, to capture the contextual information, i.e., the category dependencies. Based on the category information of Wikipedia, another query candidate set is generated. Finally, the WordNet-based synonym set and the Wikipedia-based candidate set are integrated to refine the original query.

Besides the public resources available online, the domain-specific ontology is another reliable resource that can be utilized to expand the keyword set. For example, Mukherjea et al. [82] apply Unified Medical Language System as an ontology to facilitate keyword-based patent query expansion in biomedical domain, and the result can be refined based on the semantic relations defined by the ontology. Another useful resource is the patent classification information that defines the general topic/scope of patent documents [1; 35]. Mahdabi et al. [75] treat patent classification information as domain knowledge to facilitate query expansion. Based on the international patent classification information, a conceptual lexicon is created and serves as a candidate pool to expand the keyword set. To further improve the effectiveness of patent retrieval, the proximity information of patent documents is exploited to restrict the boundary of query expansion. Recently, Tannebaum et al. [99; 100] introduce the query logs as expert knowledge to improve query expansion. Based on the analysis of query logs, they extract the frequent patterns of query terms and treat them as rules to expand the original keyword set.

3.4.2 Feedback-Based Methods

The idea of relevance feedback [89] is to employ user feedbacks to improve the search result in the process of information retrieval. However in practice, it is often difficult to obtain direct user feedbacks on the relevance of the retrieved documents, especially in patent retrieval. Hence, researchers usually exploit indirect evidence rather than explicit feedback of the search result. Generally, there are two types of approaches to acquire indirected relevant feedback: *pseudo relevance feedback* and *citation analysis*.

Pseudo relevance feedback: Pseudo relevance feedback (Pseudo-RF) [117], also known as blind relevance feedback, is a standard retrieval technique that regards the top k ranked documents from an initial retrieval as relevant documents. It automates the manual process of relevance feedback so that the user gets improved retrieval performance without an extended interaction [78]. Pseudo-RF has been extensively explored in the area of patent retrieval. Several related approaches have been proposed to employ Pseudo-RF to facilitate the retrieval performance of patent search. In NTCIR3, Kazuaki [52] exploits two relevance feedback models, including the Rocchio [89] model and Taylor expansion based model, and then extends relevance feedback methods to pseudo relevance feedback methods by assuming the top-ranked k documents as relevant documents. In NTCIR4 [43] and NTCIR5 [96], several participants attempt to utilize different Pseudo-RF approaches to improve the retrieval effectiveness. However, existing studies indicate that Pseudo-RF based approaches perform relatively poor on patent retrieval tasks, as it suffers from the problem of topic drift due to the ambiguity and synonymity of terms [71]. To alleviate the negative effect of topic drift, Bashir and Rauber [8] provide a clustering-based approach to determine whether a document is relevant or irrelevant. Based upon the intra-cluster similarity, they select top ranked documents as relevant feedback from top ranked clusters. Recently, Mahdabi et al. [74] utilize a regression model to predict the relevance of a returned document combined with a set of features (e.g. IPC clarity and query clarity). Their experiments demonstrate the superiority of the proposed method over the standard pseudo relevance feedback method. Based on this approach, in [73], they introduce an additional key-phrase extraction method by calculating phrase importance scores to further improve the performance.

Citation analysis: There are two types of citations assigned to patent documents: applicant-assigned citations and examiner-assigned citations. The first type of citations are produced by patent applicants, and often appear in the specification of patent applications in a way similar to the case that research papers are cited. Comparatively, citations assigned by patent examiners are often obtained based on the results from patentability search of the patent application, and hence might be more accurate because of the authority of the examiners.

Citations are good indicators of relevance among patent documents, and thus are often utilized to improve the search results. For example, Fuji [25] considers the cited documents as relevance feedback to expand the original query. Based on the empirical evaluation, the retrieval performance can be significantly improved by virtue of patents citation information. In CLEF 2009 IP track, Magdy et al. [68] propose to automatically extract the applicant-assigned citations from patent documents, and utilize these cited documents to facilitate patent retrieval. They further improve the citation feedback method by introducing additional terminological resources such as Wikipedia [72].

4. PATENT CLASSIFICATION

Patent classification is an important task in the process of patent application, as it provides functionalities to enable flexible management and maintenance of patent documents. However in recent years, the number of patent documents is rapidly increasing worldwide, which increases the demand for powerful patent mining systems to automatically categorize patents. The primary goal of such systems is to replace the time-consuming and labor-intensive manual categorization, and hence to offer patent analysts an efficient way to manage patent documents.

Since 1960, automatic classification has been identified as an interesting problem in text mining and natural language processing. Nowadays, in the field of text classification, researchers have devised many excellent algorithms to address this problem. However, as we previously described, it is still a non-trivial task in the domain of patent mining due to the complexity of patent documents and patent classification criteria. There are several challenges during the process of patent classification, including (1) patent documents often involve the sophisticated structures, verbose pages, and rhetorical descriptions, which renders automatic classification ineffective as it is difficult to extract useful features; (2) the hierarchical structure of the patent classification schema is quite complex, e.g. there are approximately 72,000 subgroups in the bottom level of IPC taxonomy; and (3) the huge volume of patent documents, as well as the increasing variety of patent topics, exacerbates the difficulty of automatic patent classification.

To overcome these challenges, researchers have put a lot of efforts in designing effective classification systems in the past decades. The major focus along this research direction includes (1) utilizing different types of information to perform classification; and (2) testing the performance of different classification algorithms on patent documents.

4.1 On Using Different Resources

The bag-of-words (BOW) model is often employed to represent unstructured text document. In the domain of patent document classification, the BOW representation has been widely explored. For example, Larkey [58] proposes a patent classification system in which terms and phrases are selected to represent patent documents, weighted by the frequency and structural information. Based on the vector space model, KNN (K-Nearest Neighbors) and Naïve Bayes classification models are employed to categorize US patent documents. The experiments indicate that the performance of KNN-based classifier is better than that of Naïve Bayes in the task of patent classification. After that, Koster et al. [56] propose a new approach which employs the Winnow algorithm [33] to classify patent applications. The BOW-based model is utilized to represent patent documents. Based on their experiment result, they state that the accuracy of using full-text documents is much better than that of abstracts.

The popularity of the BOW-based representation is originated from its simplicity. However, it is often difficult to convey the relationships among terms by using the BOW-based model. To address this issue, Kim et al. [49] propose a new approach to facilitate patent classification by introducing the semantic structural information. They predefine six semantic tags, including technological field, purpose, method, claim, explanation and example. Given a patent document, they convert it to the new representation based on these semantic tags. They then calculate the similarity based on both the term frequency and the semantic tag. Finally, KNN-based model is exploited to automatically classify the Japanese patent documents. The proposed approach achieves 74% improvement over the prior approaches in Japanese patent classification.

It has been widely recognized that patent classification is difficult due to the complexly structure and professional criteria of the current patent classification schema. Hence, beside exploiting the existing patent classification schema to categorize patent documents, some researchers explore the possibility of using other types of taxonomies to fulfill this task. For example, in [86], Pesenhofer et al. exploit a new taxonomy generated from Wikipedia to categorize patent documents. Cong et al. [66] design a TRIZ-based patent classification system in which TRIZ [4] is a widely used technical problem solving theory. These systems provide flexible functionalities to allow users to search relevant patent documents based on the applied taxonomy.

4.2 On Using Different Classifiers

Following the aforementioned efforts, researchers are also interested in exploring what types of classification algorithm can help improve the classification accuracy. For example, Fall et al [23; 24] compare the performance of different classification algorithms in categorizing patent documents, including Naïve Bayes, Support Vector Machine (SVM), KNN, and Winnow. Besides, they also compare the effect of utilizing different parts of patent documents, such as titles, claims, and the first 300 words of the description. Their experiments have shown that SVM achieves the best performance for class-level patent document categorization, and it is the best way to use the first 300 words of the description for representing patent documents.

As mentioned in Section 2, the IPC classification system is a five-level classification schema which contains more than 70,000 sub-groups in the bottom level. The fine-grained class label information renders patent classification more difficult. To alleviate this problem, Chen et al. [19] present a hybrid categorization system that contains three steps. Firstly, they train an SVM classifier to categorize patent documents to different sub-classes; they then train another SVM classifier to separate the documents to the bottom level of IPC; finally, they exploit KNN classification algorithms to assign the classification code to the given patent document based on the selected candidates. In their experiments, they compare various approaches employed in the sub-group level patent classification and show that their approach achieves the best performance.

Besides the traditional classification models, hierarchical approaches have also been explored, given the fact that the patent classification schema can naturally be represented as a taxonomy, as described in Section 2. For example, in [13], Cai and Hofmann present a novel hierarchical classification method that generalizes SVM. In their method, structured discriminant functions are used to mirror the class hierarchy. All the parameters are learned jointly by optimizing a common objective function with respect to a regularized upper bound on the empirical loss. The experiments on the WIPO-alpha patent collection demonstrate the effectiveness of their method. Another hierarchical model involves [103], in which the taxonomy information is integrated into an online classifier. The results on the WIPO-alpha and Espace A/B patent collections show that the method outperforms other state-of-the-art approaches significantly.

5. PATENT VISUALIZATION

The complex structure of patent documents often prevents the analysts from quickly understanding the core idea of patents. To resolve this issue, it would be helpful to visualize patent documents in a way that the gist of patents can be clearly shown to the analysts, and the correlations between different patents can be easily identified. This is often referred to as *patent visualization*, an application of information visualization.

As introduced in Section 1, a patent document contains dozens of items for analysis, which can be grouped into two categories:

- *structured data*, including patent number, filing date, issued date, and assignees, which can be utilized to generate a patent graph by employing data mining techniques;

- *unstructured text*, consisting of textual content of patent documents, such as abstract, descriptions of the invention, and major claims, which can be used to generate a patent map by employing text mining techniques.

In the following, we will discuss how patent documents can be visualized using these two types of data, as well as the integration of them.

5.1 Using Structured Data

For the purpose of analysis, structured data in patent documents are often represented as graphs. The primary resource used for constructing graphs is the citation information among different patents. By analyzing the citation graph, it is easy to discover interesting patterns with respect to particular patent documents. An example of patent citation graphs is illustrated in Figure 7a. Along this direction, several research work has been published, in which graphs are used to model patent citations. For example, in [42], Huang et al. create a patent citation graph of high-tech electronic companies in Taiwan between 1998 and 2000, where each point denotes an assignee, and the link between two points represents the relationship between them. They categorize the companies into 6 major groups, and apply graph analysis to show the similarity and distinction between different groups.

Citation analysis has been the most frequently adopted tool in visualizing the relationships of patent documents. However in some cases, it is difficult to capture the big picture of all the patent documents purely using a citation graph, as citations are insufficient to grasp the inner relations among patents. To alleviate this issue, Yoon and Park propose a network-based patent analysis method, in which the overall relationship among patents is represented as a visual network [123]. In addition, the proposed method takes more diverse keywords into account and produces more meaningful indices, which enable deeper analysis of patent documents. Tang et al. [97] further extend this idea by constructing a patent heterogeneous network, which involves a dynamic probabilistic model to characterize the topical evolution of patent documents within the network.

5.2 Using Unstructured Text

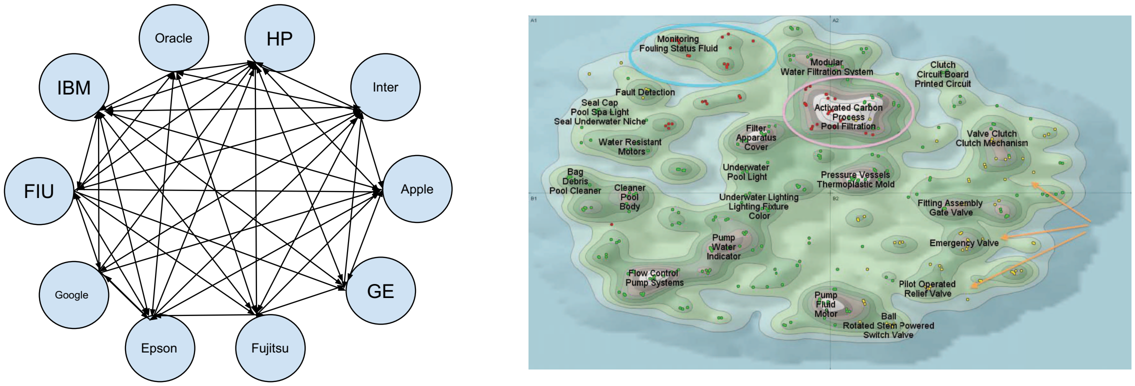
Unstructured text in patent documents provides rich information of the core ideas of patents, and therefore it becomes the primary resource for patent analysts to perform content analysis. Compared with the citation analysis, the content-based patent map has considerable advantages in latent information extraction and global technology visualization. It can also help reduce the burden of domain knowledge dependance. In the last decade, several visualization approaches have been proposed to explore the underlying patterns of patent documents and present them to users. For example, in [124], Yoon et al. present three types of patent maps, including technology vacuum map, claim point map, and technology portfolio map, all of which are generated from the unstructured text of patent documents. Figure 7b shows a patent landscape map. Similarly, Atsushi et al. [5] propose a technology portfolio map generated using the concept-based vector space model. In their model, they apply single value decomposition on the word co-occurrence matrix to obtain the word-concept matrix, and then exploit the concept-based vector to represent patent documents. To generate the patent landscape map, they employ the hierarchical clustering method based on the calculated document-concept matrix. More recently, Lee et al. [61] present an approach to generating the technology vacuum map based on patent keyword vectors. They employ principal component analysis to reduce the space of keyword features to make suitable for use on a two-dimensional map, and then identify the "technology vacuum areas" as the blank zones with sparse density and large size in the map.

5.3 Integrating Structured and Unstructured Data for Visualization

Unstructured text is useful for analyzing the core ideas of patents, and structure data provide evidences on the correlations of different patent documents. These two types of information are often integrated together for the purpose of visualization. As a representative work, Kim et al. [51] propose a novel visualization method based on both structured and unstructured data. Specifically, they first collect keywords from patent documents under a specific technology domain, and represent patent documents using keyword-based vectors. They then perform clustering on patent documents to generate k clusters. With the clustering result, they form a semantic network of keywords, and then build up a patent map by rearranging each keyword node according to its earliest filing date and frequency in patent documents. Their approach not only describes the general picture of the targeted technology domain, but also presents the evolutionary process of the corresponding techniques. In addition, natural language prossing is utilized to facilitate patent map generation [125]. Compared with the traditional technology vacuum map purely built on patent content, this approach integrates bibliographic information of patent documents, such as assignee and file date, to construct the patent maps. The generated patent map is able to assist experts in understanding technological competition trends in the process of formulating R&D strategies.

6. PATENT VALUATION

Patent documents are the core of many technology organizations and companies. To support decision making, it is imperative to assess the quality of patent documents for further actions. In practice, a common process of evaluating the importance/quality of patent documents is called *patent valuation*, which aims to assist internal decision making for patent protection strategies. For example, companies may



(a) Patent Assignee Citation Graph (Source:NodeXL)

(b) Water Patent Landscape Map (Source:CleanTech)

Figure 7: Representative examples of patent visualization.

create a collection of related patents, called *patent portfolio* [113], to form a "super-patent" in order to increase the coverage of protection. In this case, a critical question is how to explore and evaluate the potential benefit of patent documents so as to select the most important ones. To tackle this issue, researchers often resort to two types of approaches: *unsupervised exploration* and *supervised evaluation*. In the following, we discuss existing research publications related to patent valuation from these two perspectives.

6.1 Unsupervised Exploration

Unsupervised exploration on the importance of patent documents is often oriented towards two aspects: *influence power* and *technical strength*. The former relies on the linkage between patent documents, e.g., citations, whereas the latter mainly focus on the content analysis.

Influence power: The first work of using citations to evaluate the influence power of patent documents involves [20]. In this work, a citation graph is constructed, where each node indicates a patent document, and nodes link to others based on their citation relations. The case study of semi-synthetic penicillin demonstrates the effectiveness of using citation counts in assessing the influence power of patents. In [3], Albert et al. further extend the idea of using citation counts, and prove the correctness of citation analysis to evaluate patent documents. In addition, two related techniques are proposed, including the bibliographic coupling that indicates two patent documents share one or more citation, and co-citation analysis that indicates two patent documents have been cited by one or more patent documents. Based on these two techniques, Huang et al. [42] integrate the bibliographic coupling analysis and multidimensional scaling to assess the importance of patent documents. Further, ranking-based approaches can also be applied to the process of patent valuation. For example, Fujii [25] proposes the use of PageRank [12] to calculate citation-based score for patent documents.

Technical strength: Unlike approaches that rely on the analysis of the influence power of patent documents, some research publications focus on the analysis of the technical strength of inventions, which is relevant to the content of patents. For instance, Hasan et al. [36] define the technical strength as claim originality, and exploit text mining approaches to analysis the novelty of patent documents. They use NLP techniques to extract the key phrases from the claims section of patent documents, and then calculate the originality score based on the extracted key phrases. This valuation method has been adopted by IBM, and is applied to various patent valuation scenarios; however, the term-based approaches suffer the problem of term ambiguity, which may deteriorate the rationality of the scores in some cases. To alleviate this issue, Hu et al. [41] exploit the topic model to represent the concept of the patents instead of using words or phrases. In additional, they state that traditional patent valuation approaches cannot handle the case that the novelty of patents evolves over time, i.e., the novelty may decrease along time. Therefore, they exploit the time decay factor to capture the evolution of patent novelty. The experiment indicates that their proposed approach achieves the improvement compared with the baselines.

6.2 Supervised Evaluation

The aforementioned approaches define the importance of patent documents from either content or citation links. In essence, they are unsupervised methods as the goal is to extract meaningful patterns to assess the value of patents purely based on the patent itself. In practice, besides these two types of resources, some other information may also be available to exploit. Some researchers introduce other types of patent related records, such as patent examination results [37], patent maintenance decisions [46], and court judgments [65], to generate predicated models to evaluate patent documents. For example, Hido et al. [37] create a learning model to estimate the patentability of patent applications from the historical Japan patent examination data, and then use the model to predict the examination decision for new patent applications. They define the patentability prediction problem as a binary classification problem (reject or approval). In order to obtain an accuracy classifier, they exploit four types of features, including patent document structure, term frequency, syntactic complexity, and word age [36]. From their experiments, they demonstrate the superiority of the proposed method in estimating the examination decision. Jin et al. [46] construct a heterogeneous information network from patent documents corpus, in which nodes could be inventors, classification codes, or patent documents and edges could denote the classification similarity, the citation relation or inventor cooperation, etc. Based on this heterogeneous network, they define interesting features, such as meta features, novelty features, and writing quality features, to created a patent quality model that is able to predict the value of patents and give the maintenance decision suggestion. Liu et al. [65] propose a graphical model that discovers the valid patents which have highly probability to achieve the victory during the patent litigation process. Based on the patent citation count and court judgments, they define a latent variable to estimate the quality of patent documents. They further incorporate various quality-related features, e.g., citation quality, complexity, reported coverage, and claim originality, to improve the probabilistic model. The experiments indicate that their approach achieves promising performance for predicting court decisions.

7. CROSS-LANGUAGE PATENT MINING

Patent documents are quite sensitive to regions, i.e., patents from different regions might be described by different languages. However in reality, patent analysts prefer to receive localized patent information, even if they are described by multiple languages. For example, a patent document is written by English, but an analyst from Spain expects that this patent can be translated to Spanish for better understanding. In addition, international patent documents are required to be written by the language accepted worldwide, which is often referred to as patent globalization. In such cases, cross-language patent mining is needed to support patent localization/globalization.

In the current stage of cross-language patent mining, the primary task is cross-language information retrieval, which enables us to retrieve information from other languages using a query written in the language that we are familiar with. In general, a cross-language patent retrieval system can be constructed using two techniques: *machine translation* and *semantic correspondence*. In the following, we describe the details of these two techniques and discuss existing research efforts on this direction.

7.1 Using Machine Translation

A well-known technique to address cross-language retrieval is machine translation. By translating a query to the desired language, the problem can be reduced to a monolingual information retrieval task that various approaches can be employed. Popular machine translation systems, such as Google Translate[10], Bing Translator[11], and Cross Language[12], have been widely exploited in tackling the problem of cross-language patent retrieval [18; 48; 70; 77]. The NTCIR Workshop holds a machine translation track to encourage researchers to practice the cross-lingual patent retrieval task [27]. In [77], Makita et al. present a multilingual patent retrieval system based on the method proposed in [26], which employs a probabilistic model to reduce the ambiguity of query translation. As indicated in the report of NTCIR9 Patent Machine Translation task [32], several participants propose word-based and phrase-based translation approaches by exploiting Moses [53], an open source toolkit for statistical machine translation. Their experiments demonstrate that lexicon-based approaches are able to achieve acceptable performance; however, the domain-specific terms and structural sentences of patent documents are difficult to translate. Hence, it is imperative to explore the syntactic structure of patents when performing patent document translation.

7.2 Using Semantic Correspondence

An alternative way of building a cross-language patent search engine is to explore the semantic correspondence among languages. The basic idea is to first construct the semantic relations of a pair of languages, and then interpret the query to another language. In [64], Littman et al. present a novel approach which creates a cross-language space by exploiting latent semantic indexing(LSI) in cross-language information retrieval domain. Base on the research of [64], Li et al. [62] propose a new approach to retrieve patent documents in the Japanese-English collection. They introduce the method of kernel canonical correlation analysis [110] to build a cross-language sematic space from Japanese-English patent documents. The empirical evaluation shows that the proposed method achieves significant improvement over the state-of-the-art. However, it may require a lot of efforts to build a cross-language semantic space, and also the performance of this type of approaches is restricted by the quality of the semantic space.

8. APPLICATIONS

Patent mining aims to assist patent analysts in efficiently and effectively managing huge volume of patent documents. It is essentially an application-driven area that has been extensively explored in both academia and industry. There are a lot of online patent mining systems, either with free access or having commercial purposes. Table 3 lists several representative systems that provide flexible functionalities of patent retrieval and patent analysis (Part of the content is obtained from *Intellogist*[13]).

Patent mining systems, e.g., *Google Patent*[14], *Baidu Patent*[15] and *FreePatentOnine*[16], provide free access and basic retrieval functionalities and are very easy to use for the majority. In addition, a list of patent authorities, e.g., USPTO[17], EPO[18], WIPO[19], provide advanced search functions to allow professional users to input more complex patent queries for high-recall retrieval. These authority-based systems usually require more human efforts and domain expertise.

Some leading companies, e.g., Thomson Reuters, Questel, and Lexisnesxis, offer commercial patent mining systems. Compared with the systems with free access, commercial systems provide more advanced features to assist analysts in retrieval and processing patent documents. These commercial systems often have:

- Widespread scope. Most commercial systems not only cover patent data from multiple authorities, but also

[10] http://translate.google.com.

[11] http://www.bing.com/translator.

[12] http://www.crosslanguage.co.jp.

[13] http://www.intellogist.com.

[14] https://www.google.com/?tbm=pts.

[15] http://zhuanli.baidu.com.

[16] http://www.freepatentsonline.com.

[17] http://www.uspto.gov.

[18] http://www.epo.org.

[19] http://www.wipo.int.

Table 3: Comparison among different patent mining systems.

Systems	Thomson Innovation	Orbit	Total Patent	ProQuest	PatFT	Espacenet	Patent Scope	Google Patent	Free Patents Online
Owner	Thomson Reuters	Questel	LexisNexis	Quest	USPTO	EPO	WIPO	Google	Free Patents Online
Data Coverage(Number of authorities)	8	21	32	3	1	2	1	6	3
Legal Status Data	Yes	Yes	Yes	Yes	No	No	No	No	No
Non-Patent Sources	Yes	Yes	Yes	Yes	No	Yes	No	No	No
Legal Status Data	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Quick Search	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Advanced Search	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Keyword Term Highlighting	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Personalize Result	Yes	Yes	Yes	No	Yes	No	No	No	Yes
Keep Queries History	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes
Queries Combination	Yes	Yes	Yes	Yes	No	No	No	No	No
Bulk Documents Download	Yes	Yes	Yes	Yes	No	Yes	No	No	No
Warning Mechanism	Yes	Yes	Yes	No	No	No	No	No	No
Statistical Analysis	Yes	Yes	Yes	Yes	No	No	Yes	No	No
Patents Graphs	Yes	Yes	Yes	Yes	No	No	No	No	No
Keyword Analysis	Yes	Yes	Yes	Yes	No	No	No	No	No
Advanced Analysis	Yes	Yes	Yes	Yes	No	No	No	No	No

integrate other types of resources. For example, Thomson Reuters includes science and business articles, Questel combines news and blogs, and Lexisnesxis considers law cases. These resources are complementary to patent documents and are able to enhance the analysis power of the systems.

- Cutting-edge analysis. Commercial systems often provide patent analysis functionalities, by which more meaningful and understandable results can be obtained. For example, Thosmson Innovation provides a function called *Themescape* that identifies common themes within the search results by analyzing the concept clusters and then vividly presents them to users.

- Export functionality. Compared with free patent retrieval systems that do not allow people to export the search results, most commercial systems provide customized export functions that enable users to select and save different types of information.

Recently, several patent mining systems have been proposed in academia, most of which are constructed by utilizing the available online resources. For example, *PatentSearcher* [40] leverages the domain semantics to improve the quality of discovery and ranking. The system uses more patent fields, such as abstract, claims, descriptions and images, to retrieve and rank patents. *PatentLight* [14] is an extension of *PatentSearcher*, which categorizes the search results by virtue of the tags of the XML-structure, and ranks the results by considering flexible constraints on both structure and content. Another representative system is called *PatentMiner* [97], which studies the problem of dynamic topic modeling of patent documents and provides the topic-level competition analysis. Such analysis can help patent analysts identify the existing or potential competitors in the same topic. Further, there are some mining systems focusing on patent image search. For instance, *PATExpert* [115] presents a semantic multimedia content representation for patent documents based on semantic web technologies. *PatMedia* [112] provides patent image retrieval functionalities in content-based manner. The visual similarity is realized by comparing visual descriptors extracted from patent images.

9. CONCLUDING REMARKS

In this survey, we comprehensively investigated several technical issues in the field of patent mining, including patent search, patent categorization, patent visualization, and patent evaluation. For each issue, we summarize the corresponding technical challenges exposed in real-world applications, and explore different solutions to them from existing publications. We also introduce various patent mining systems, and discuss how the techniques are applied to these systems for efficient and effective patent mining. In summary, this survey provides an overview on existing patent mining techniques, and also sheds light on specific application tasks related to patent mining.

With the increasing volume of patent documents, a lot of application-oriented issues are emerging in the domain of patent mining. In the following, we identify a list of challenges in this domain with respect to several mining tasks.

- *Figure-Based Patent Search* introduces patent drawings as additional information to facilitate traditional patent search tasks, as technical figures are able to vividly demonstrate the core idea of invention in some domains, especially in electronics and mechanisms. The similarity between technical figures may help improve the accuracy of patent search.

- *Product-Based Patent Search*: In general, a product may be associated with multiple patents. For example, "iPhone" contains a list of key components, such as touchscreen, frame, adapter, and operating systems. What are the patents related to each component? We call this case as product-based patent search, which provides the component-level patent search results for a product.

- *Patent Infringement Analysis* aims to decide whether two patent documents are similar or one is covered by another. In general, the analysts have to manually read through lengthy patent documents to determine the equivalence/coverage. It is necessary to automate this process, or at least to provide concise summaries to ease the understanding.

- *Large-Scale Patent Retrieval* aims to alleviate the scalability issue of patent search engines. Due to the large volume of patent documents, the performance of traditional patent retrieval systems cannot meed the expectation of patent analysts. To resolve this problem, patent documents need to be carefully processed and indexed.

- *Multi-Label Hierarchical Patent Classification* denotes the process of automatically categorizing patent documents into the pre-defined classification taxonomies [13], e.g., IPC or USPC. This is a crucial step in patent document management and maintenance. However, existing approaches to solving this problem cannot efficiently handle large classification taxonomies.

- *Technique Evolution Analysis* involves generating a technology evolution tree for a given topic or a classification code related to granted patents [126]. It is a representative application of patent visualization, which enables us to effectively understand technological progress, comprehend the evolution of technologies and grab the emergence of new technologies.

- *Detecting Potential Collaborators/Competitors*: When a company would like to design a new product, a problem usually encountered by the company is who to collaborate with. Identifying potential collaborators is helpful to reduce the cost, as well as to accelerate the process of the product. In addition, the company needs to acquire features of similar products by the competitors.

- *Cross-Domain Patent Recommendation*: Online news services give people opportunities to quickly grasp the trending techniques in industry by reading technical news articles. However, tech news articles often contain a list of uncommon terms that cannot be easily understood by the audience, and consequently hinder news readers' reading experience. Therefore, it would be helpful to present patent summaries to news readers for better understanding of tech news.

Some challenges, such as the scalability and classification issues, are imperative to solve in order to assist patent analysts in efficiently and effectively performing patent analysis tasks. Other challenges can stimulate the emergence of new types of patent-oriented applications, such as evolutionary analysis and drawing-based retrieval. Even though it is impossible to describe all algorithms and applications in detail for patent mining, we believe that the ideas and challenges discussed in this survey should give readers a big picture of this field and several interesting directions for future studies.

10. ACKNOWLEDGMENTS

The work was supported in part by the National Science Foundation under grants DBI-0850203, HRD-0833093, CNS-1126619, and IIS-1213026, the U.S. Department of Homeland Security under grant Award Number 2010-ST-06200039, and Army Research Office under grant number W911NF-1010366 and W911NF-12-1-0431.

11. REFERENCES

[1] S. Adams. Comparing the ipc and the us classification systems for the patent searcher. *World Patent Information*, 23(1):15–23, 2001.

[2] B. Al-Shboul and S. Myaeng. Query phrase expansion using wikipedia in patent class search. *Information Retrieval Technology*, pages 115–126, 2011.

[3] M. Albert, D. Avery, F. Narin, and P. McAllister. Direct validation of citation counts as indicators of industrially important patents. *Research Policy*, 20(3):251–259, 1991.

[4] G. S. Altšuller. *The innovation algorithm: TRIZ, systematic innovation and technical creativity.* Technical Innovation Center, Inc., 1999.

[5] H. Atsushi and T. YUKAWA. Patent map generation using concept-based vector space model. *working notes of NTCIR-4, Tokyo*, pages 2–4, 2004.

[6] L. Azzopardi, W. Vanderbauwhede, and H. Joho. Search system requirements of patent analysts. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 775–776. ACM, 2010.

[7] R. Bache and L. Azzopardi. Improving access to large patent corpora. *Transactions on large-scale data-and knowledge-centered systems II*, pages 103–121, 2010.

[8] S. Bashir and A. Rauber. Improving retrievability of patents with cluster-based pseudo-relevance feedback documents selection. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1863–1866. ACM, 2009.

[9] S. Bashir and A. Rauber. Improving retrievability of patents in prior-art search. *Advances in Information Retrieval*, pages 457–470, 2010.

[10] S. Bhatia, B. He, Q. He, and S. Spangler. A scalable approach for performing proximal search for verbose patent search queries. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2603–2606. ACM, 2012.

[11] F. Borgonovi. Divided we stand, united we fall: Religious pluralism, giving, and volunteering. *American Sociological Review*, 73(1):105–128, 2008.

[12] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998.

[13] L. Cai and T. Hofmann. Hierarchical document categorization with support vector machines. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 78–87. ACM, 2004.

[14] S. Calegari, E. Panzeri, and G. Pasi. Patentlight: a patent search application. In *Proceedings of the 4th Information Interaction in Context Symposium*, pages 242–245. ACM, 2012.

[15] R. S. Campbell. Patent trends as a technological forecasting tool. *World Patent Information*, 5(3):137–143, 1983.

[16] M. Carrier. A roadmap to the smartphone patent wars and frand licensing. *Antitrust Chronicle*, 4, 2012.

[17] S. Cetintas and L. Si. Effective query generation and postprocessing strategies for prior art patent search. *Journal of the American Society for Information Science and Technology*, 63(3):512–527, 2012.

[18] M. Chechev, M. Gonzàlez, L. Màrquez, and C. España-Bonet. The patents retrieval prototype in the molto project. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 231–234. ACM, 2012.

[19] Y. Chen and Y. Chang. A three-phase method for patent classification. *Information Processing & Management*, 2012.

[20] P. Ellis, G. Hepburn, and C. Oppenhein. Studies on patent citation networks. *Journal of Documentation*, 34(1):12–20, 1978.

[21] P. Érdi, K. Makovi, Z. Somogyvári, K. Strandburg, J. Tobochnik, P. Volf, and L. Zalányi. Prediction of emerging technologies based on analysis of the us patent citation network. *Scientometrics*, pages 1–18, 2012.

[22] J. Erstling and I. Boutillon. Patent cooperation treaty: At the center of the international patent system. *Wm. Mitchell L. Rev.*, 32:1583–1600, 2005.

[23] C. Fall, A. Torcsvari, K. Benzineb, and G. Karetka. Automated categorization in the international patent classification. In *ACM SIGIR Forum*, volume 37, pages 10–25, 2003.

[24] C. Fall, A. Törcsvári, P. Fievet, and G. Karetka. Automated categorization of german-language patent documents. *Expert Systems with Applications*, 26(2):269–277, 2004.

[25] A. Fujii. Enhancing patent retrieval by citation analysis. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 793–794. ACM, 2007.

[26] A. Fujii and T. Ishikawa. Japanese/english cross-language information retrieval: Exploration of query translation and transliteration. *Computers and the Humanities*, 35(4):389–420, 2001.

[27] A. Fujii, M. Utiyama, M. Yamamoto, and T. Utsuro. Evaluating effects of machine translation accuracy on cross-lingual patent retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 674–675. ACM, 2009.

[28] S. Fujita. Technology survey and invalidity search: A comparative study of different tasks for japanese patent document retrieval. *Information processing & management*, 43(5):1154–1172, 2007.

[29] D. Ganguly, J. Leveling, and G. Jones. United we fall, divided we stand: A study of query segmentation and prf for patent prior art search. In *Proceedings of the 4th workshop on Patent information retrieval*, pages 13–18. ACM, 2011.

[30] D. Ganguly, J. Leveling, W. Magdy, and G. Jones. Patent query reduction using pseudo relevance feedback. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1953–1956. ACM, 2011.

[31] J. Gobeill, A. Gaudinat, P. Ruch, E. Pasche, D. Teodoro, and D. Vishnyakova. Bitem site report for trec chemistry 2010: Impact of citations feeback for patent prior art search and chemical compounds expansion for ad hoc retrieval. In *TREC*, 2010.

[32] I. Goto, B. Lu, K. P. Chow, E. Sumita, and B. K. Tsou. Overview of the patent machine translation task at the ntcir-9 workshop. In *Proceedings of NTCIR*, volume 9, pages 559–578, 2011.

[33] A. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. *Machine Learning*, 43:173–210, 2001.

[34] H. Gurulingappa, B. Müller, R. Klinger, H.-T. Mevissen, M. Hofmann-Apitius, C. M. Friedrich, and J. Fluck. Prior art search in chemistry patents based on semantic concepts and co-citation analysis. In *TREC*, 2010.

[35] C. G. Harris, R. Arens, and P. Srinivasan. Comparison of ipc and uspc classification systems in patent prior art searches. In *Proceedings of the 3rd international workshop on Patent Information Retrieval*, pages 27–32. ACM, 2010.

[36] M. A. Hasan, W. S. Spangler, T. Griffin, and A. Alba. Coa: Finding novel patents through text analysis. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1175–1184. ACM, 2009.

[37] S. Hido, S. Suzuki, R. Nishiyama, T. Imamichi, R. Takahashi, T. Nasukawa, T. Idé, Y. Kanehira, R. Yohda, T. Ueno, et al. Modeling patent quality: A system for large-scale patentability analysis using text mining. *JIP*, 20(3):655–666, 2012.

[38] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2004.

[39] Q. Honghua and Y. Xiang. Research on a method for building up a patent map based on k-means clustering algorithm. *Science Research Management*, 2:1–9, 2009.

[40] V. Hristidis, E. Ruiz, A. Hernández, F. Farfán, and R. Varadarajan. Patentssearcher: a novel portal to search and explore patents. In *Proceedings of the 3rd international workshop on Patent information retrieval*, pages 33–38. ACM, 2010.

[41] P. Hu, M. Huang, P. Xu, W. Li, A. K. Usadi, and X. Zhu. Finding nuggets in ip portfolios: core patent mining through textual temporal analysis. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1819–1823. ACM, 2012.

[42] M. Huang, L. Chiang, and D. Chen. Constructing a patent citation map using bibliographic coupling: A study of taiwan's high-tech companies. *Scientometrics*, 58(3):489–506, 2003.

[43] H. Itoh. Ntcir-4 patent retrieval experiments at ricoh. In *NTCIR-4*, 2004.

[44] M. Iwayama, A. Fujii, N. Kando, and A. Takano. Overview of patent retrieval task at ntcir-3. In *Proceedings of the ACL-2003 workshop on Patent corpus processing-Volume 20*, pages 24–32. Association for Computational Linguistics, 2003.

[45] A. Jaffe and M. Trajtenberg. *Patents, citations, and innovations: A window on the knowledge economy.* MIT press, 2005.

[46] X. Jin, S. Spangler, Y. Chen, K. Cai, R. Ma, L. Zhang, X. Wu, and J. Han. Patent maintenance recommendation with patent information network model. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 280–289. IEEE, 2011.

[47] Y. Jin. A hybrid-strategy method combining semantic analysis with rule-based mt for patent machine translation. In *Natural Language Processing and Knowledge Engineering (NLP-KE), 2010 International Conference on*, pages 1–4. IEEE, 2010.

[48] C. Jochim, C. Lioma, H. Schütze, S. Koch, and T. Ertl. Preliminary study into query translation for patent retrieval. In *Proceedings of the 3rd international workshop on Patent information retrieval*, pages 57–66. ACM, 2010.

[49] J. Kim and K. Choi. Patent document categorization based on semantic structural information. *Information processing & management*, 43(5):1200–1215, 2007.

[50] Y. Kim, J. Seo, and W. Croft. Automatic boolean query suggestion for professional search. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 825–834. ACM, 2011.

[51] Y. Kim, J. Suh, and S. Park. Visualization of patent analysis for emerging technology. *Expert Systems with Applications*, 34(3):1804–1812, 2008.

[52] K. Kishida. Experiment on pseudo relevance feedback method using taylor formula at ntcir-3 patent retrieval task. In *Proceedings of the Third NTCIR Workshop on Research in Information Retrieval, Automatic Text Summarization and Question Answering, NII, Tokyo. http://research. nii. ac. jp/ntcir*, 2003.

[53] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics, 2007.

[54] S. Kondo, M. Komachi, Y. Matsumoto, K. Sudoh, K. Duh, and H. Tsukada. Learning of linear ordering problems and its application to je patent translation in ntcir-9 patentmt. In *Proceedings of NTCIR*, volume 9, pages 641–645, 2011.

[55] K. Konishi. Query terms extraction from patent document for invalidity search. In *Proceedings of NTCIR*, volume 5, 2005.

[56] C. Koster, M. Seutter, and J. Beney. Multi-classification of patent applications with winnow. In *Perspectives of System Informatics*, pages 111–125, 2003.

[57] A. Krishnan, A. F. Cardenas, and D. Springer. Search for patents using treatment and causal relationships. In *Proceedings of the 3rd international workshop on Patent information retrieval*, pages 1–10. ACM, 2010.

[58] L. Larkey. *Some issues in the automatic classification of US patents*. Massachusetts univ amherst Department of computer Science, 1997.

[59] L. Larkey. A patent search and classification system. In *International Conference on Digital Libraries: Proceedings of the fourth ACM conference on Digital libraries*, volume 11, pages 179–187, 1999.

[60] C. Lee, Y. Cho, H. Seol, and Y. Park. A stochastic patent citation analysis approach to assessing future technological impacts. *Technological Forecasting and Social Change*, 79(1):16–29, 2012.

[61] S. Lee, B. Yoon, and Y. Park. An approach to discovering new technology opportunities: Keyword-based patent map approach. *Technovation*, 29(6):481–497, 2009.

[62] Y. Li and J. Shawe-Taylor. Advanced learning algorithms for cross-language patent retrieval and classification. *Information processing & management*, 43(5):1183–1199, 2007.

[63] Y.-R. Li. The technological roadmap of cisco's business ecosystem. *Technovation*, 29(5):379–386, 2009.

[64] M. Littman, S. Dumais, T. Landauer, et al. Automatic cross-language information retrieval using latent semantic indexing. *Cross-language information retrieval*, pages 51–62, 1998.

[65] Y. Liu, P. Hseuh, R. Lawrence, S. Meliksetian, C. Perlich, and A. Veen. Latent graphical models for quantifying and predicting patent quality. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1145–1153. ACM, 2011.

[66] H. T. Loh, C. He, and L. Shen. Automatic classification of patent documents for triz users. *World Patent Information*, 28(1):6–13, 2006.

[67] M. Lupu, F. Piroi, and A. Hanbury. Aspects and analysis of patent test collections. In *Proceedings of the 3rd international workshop on Patent information retrieval*, pages 17–22. ACM, 2010.

[68] W. Magdy and G. Jones. Applying the kiss principle for the clef-ip 2010 prior art candidate patent search task. In *Proceedings of the CLEF-2010 Conferences and Labs of the Evaluation Forum*, 2010.

[69] W. Magdy and G. Jones. A study on query expansion methods for patent retrieval. In *Proceedings of the 4th workshop on Patent information retrieval*, pages 19–24. ACM, 2011.

[70] W. Magdy and G. J. Jones. An efficient method for using machine translation technologies in cross-language patent search. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1925–1928. ACM, 2011.

[71] W. Magdy, J. Leveling, and G. J. Jones. Exploring structured documents and query formulation techniques for patent retrieval. *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, pages 410–417, 2010.

[72] W. Magdy, P. Lopez, and G. Jones. Simple vs. sophisticated approaches for patent prior-art search. *Advances in Information Retrieval*, pages 725–728, 2011.

[73] P. Mahdabi, L. Andersson, M. Keikha, and F. Crestani. Automatic refinement of patent queries using concept importance predictors. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 505–514. ACM, 2012.

[74] P. Mahdabi and F. Crestani. Learning-based pseudo-relevance feedback for patent retrieval. *Multidisciplinary Information Retrieval*, pages 1–11, 2012.

[75] P. Mahdabi, S. Gerani, J. X. Huang, and F. Crestani. Leveraging conceptual lexicon: query disambiguation using proximity information for patent retrieval. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 113–122. ACM, 2013.

[76] P. Mahdabi, M. Keikha, S. Gerani, M. Landoni, and F. Crestani. Building queries for prior-art search. *Multidisciplinary Information Retrieval*, pages 3–15, 2011.

[77] M. Makita, S. Higuchi, A. Fujii, and T. Ishikawa. A system for japanese/english/korean multilingual patent retrieval. *Proceedings of Machine Translation Summit IX(online at http://www. amtaweb. org/summit/MTSummit/papers. html)*, 2003.

[78] C. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

[79] E. Meij, W. Weerkamp, and M. de Rijke. A query model based on normalized log-likelihood. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1903–1906. ACM, 2009.

[80] H.-C. Meng. Innovation cluster as the national competitiveness tool in the innovation driven economy. *International Journal of Foresight and Innovation Policy*, 2(1):104–116, 2005.

[81] A. Messeni Petruzzelli, D. Rotolo, and V. Albino. Determinants of patent citations in biotechnology: An analysis of patent influence across the industrial and organizational boundaries. *Technological Forecasting and Social Change*, 2014.

[82] S. Mukherjea and B. Bamba. Biopatentminer: an information retrieval system for biomedical patents. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 1066–1077. VLDB Endowment, 2004.

[83] K.-L. Nguyen and S.-H. Myaeng. Query enhancement for patent prior-art-search based on keyterm dependency relations and semantic tags. *Multidisciplinary Information Retrieval*, pages 28–42, 2012.

[84] S. Oh, Z. Lei, P. Mitra, and J. Yen. Evaluating and ranking patents using weighted citations. In *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*, pages 281–284. ACM, 2012.

[85] K. OuYang and C. Weng. A new comprehensive patent analysis approach for new product design in mechanical engineering. *Technological Forecasting and Social Change*, 78(7):1183–1199, 2011.

[86] A. Pesenhofer, S. Edler, H. Berger, and M. Dittenbach. Towards a patent taxonomy integration and interaction framework. In *Proceedings of the 1st ACM workshop on Patent information retrieval*, pages 19–24. ACM, 2008.

[87] F. Piroi and J. Tait. Clef-ip 2010: Retrieval experiments in the intellectual property domain. In *Proceedings of the CLEF-2010 Conferences and Labs of the Evaluation Forum*, 2010.

[88] I. J. Rotkin, K. J. Dood, and M. A. Thexton. *A history of patent classification in the United States Patent and Trademark Office*. Patent Documentation Society, 1999.

[89] G. Salton. *The SMART retrieval system – experiments in automatic document processing*. Prentice-Hall, Inc., 1971.

[90] S. Sheremetyeva. Natural language analysis of patent claims. In *Proceedings of the ACL-2003 workshop on Patent corpus processing-Volume 20*, pages 66–73. Association for Computational Linguistics, 2003.

[91] A. Shinmori, M. Okumura, Y. Marukawa, and M. Iwayama. Patent claim processing for readability: structure analysis and term explanation. In *Proceedings of the ACL-2003 workshop on Patent corpus processing-Volume 20*, pages 56–65. Association for Computational Linguistics, 2003.

[92] B. Stein, D. Hoppe, and T. Gollub. The impact of spelling errors on patent search. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 570–579. Association for Computational Linguistics, 2012.

[93] C. Sternitzke, A. Bartkowski, and R. Schramm. Visualizing patent statistics by means of social network analysis tools. *World Patent Information*, 30(2):115–131, 2008.

[94] J. H. Suh and S. C. Park. Service-oriented technology roadmap (sotrm) using patent map for r&d strategy of service industry. *Expert Systems with Applications*, 36(3):6754–6772, 2009.

[95] T. Takaki, A. Fujii, and T. Ishikawa. Associative document retrieval by query subtopic analysis and its application to invalidity patent search. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 399–405. ACM, 2004.

[96] H. Takeuchi, N. Uramoto, and K. Takeda. Experiments on patent retrieval at ntcir-5 workshop. In *NTCIR-5*, 2005.

[97] J. Tang, B. Wang, Y. Yang, P. Hu, Y. Zhao, X. Yan, B. Gao, M. Huang, P. Xu, W. Li, et al. Patentminer: topic-driven patent analysis and mining. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1366–1374. ACM, 2012.

[98] W. Tannebaum and A. Rauber. Acquiring lexical knowledge from query logs for query expansion in patent searching. In *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*, pages 336–338. IEEE, 2012.

[99] W. Tannebaum and A. Rauber. Analyzing query logs of uspto examiners to identify useful query terms in patent documents for query expansion in patent searching: a preliminary study. *Multidisciplinary Information Retrieval*, pages 127–136, 2012.

[100] W. Tannebaum and A. Rauber. Mining query logs of uspto patent examiners. *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, pages 136–142, 2013.

[101] D. Teodoro, J. Gobeill, E. Pasche, D. Vishnyakova, P. Ruch, and C. Lovis. Automatic prior art searching and patent encoding at clef-ip'10. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[102] E. Terumasa. Rule based machine translation combined with statistical post editor for japanese to english patent translation. In *Proceedings of the MT Summit XI Workshop on Patent Translation*, volume 11, pages 13–18, 2007.

[103] D. Tikk, G. Biró, and A. Törcsvári. A hierarchical online classifier for patent categorization. *Emerging Technologies of Text Mining: Techniques and Applications. Idea Group Inc*, 2007.

[104] A. J. Trappey, C. Y. Fan, C. Trappey, Y.-L. Lin, and C.-Y. Wu. Intelligent recommendation methodology and system for patent search. In *Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on*, pages 172–178. IEEE, 2012.

[105] Y. Tseng et al. Text mining for patent map analysis. In *Proceedings of IACIS Pacific 2005 Conference*, pages 1109–1116, 2005.

[106] Y. Tseng, C. Lin, and Y. Lin. Text mining techniques for patent analysis. *Information Processing & Management*, 43(5):1216–1247, 2007.

[107] Y. Tseng, C. Tsai, and D. Juang. Invalidity search for uspto patent documents using different patent surrogates. In *Proceedings of NTCIR-6 Workshop*, 2007.

[108] Y. Tseng and Y. Wu. A study of search tactics for patentability search: a case study on patent engineers. In *Proceedings of the 1st ACM workshop on Patent information retrieval*, pages 33–36. ACM, 2008.

[109] N. Van Zeebroeck. The puzzle of patent value indicators. *Economics of Innovation and New Technology*, 20(1):33–62, 2011.

[110] A. Vinokourov, J. Shawe-Taylor, and N. Cristianini. Inferring a semantic representation of text via cross-language correlation analysis. *Advances in neural information processing systems*, 15:1473–1480, 2003.

[111] I. Von Wartburg, T. Teichert, and K. Rost. Inventive progress measured by multi-stage patent citation analysis. *Research Policy*, 34(10):1591–1607, 2005.

[112] S. Vrochidis, A. Moumtzidou, G. Ypma, and I. Kompatsiaris. Patmedia: augmenting patent search with content-based image retrieval. *Multidisciplinary Information Retrieval*, pages 109–112, 2012.

[113] R. P. Wagner and G. Parchomovsky. Patent portfolios. *U of Penn. Law School, Public Law Working Paper*, 56:04–16, 2005.

[114] J. Wang and D. W. Oard. Combining bidirectional translation and synonymy for cross-language information retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 202–209. ACM, 2006.

[115] L. Wanner, S. Brügmann, B. Diallo, M. Giereth, Y. Kompatsiaris, E. Pianta, G. Rao, P. Schoester, and V. Zervaki. Patexpert: Semantic processing of patent documentation. In *SAMT (Posters and Demos)*, 2006.

[116] T. Xiao, F. Cao, T. Li, G. Song, K. Zhou, J. Zhu, and H. Wang. Knn and re-ranking models for english patent mining at ntcir-7. In *Proceedings of NTCIR-7 Workshop Meeting*, 2008.

[117] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 4–11. ACM, 1996.

[118] X. Xue and W. Croft. Automatic query generation for patent search. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 2037–2040. ACM, 2009.

[119] X. Xue and W. Croft. Transforming patents into prior-art queries. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 808–809. ACM, 2009.

[120] Y. Yang, L. Akers, T. Klose, and C. Barcelon Yang. Text mining and visualization tools–impressions of emerging capabilities. *World Patent Information*, 30(4):280–293, 2008.

[121] Y. Y. Yang, L. Akers, C. B. Yang, T. Klose, and S. Pavlek. Enhancing patent landscape analysis with visualization output. *World Patent Information*, 32(3):203–220, 2010.

[122] T. Yeap, G. Loo, and S. Pang. Computational patent mapping: intelligent agents for nanotechnology. In *MEMS, NANO and Smart Systems, 2003. Proceedings. International Conference on*, pages 274–278. IEEE, 2003.

[123] B. Yoon and Y. Park. A text-mining-based patent network: Analytical tool for high-technology trend. *The Journal of High Technology Management Research*, 15(1):37–50, 2004.

[124] B.-U. Yoon, C.-B. Yoon, and Y.-T. Park. On the development and application of a self–organizing feature map–based patent map. *R&D Management*, 32(4):291–300, 2002.

[125] J. Yoon, H. Park, and K. Kim. Identifying technological competition trends for r&d planning using dynamic patent maps: Sao-based content analysis. *Scientometrics*, 94(1):313–331, 2013.

[126] L. Zhang, L. Li, T. Li, and Q. Zhang. Patentline: Analyzing technology evolution on multi-view patent graphs. In *Proceedings of the 37th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2014.

[127] L. Zhang and T. Li. Data mining applications in patent analysis. In *Data mining where theory meets practice*, pages 392–416. Xiamen University Press, 2013.

APPENDIX

A. BENCHMARK DATA

- *NTCIR Patent Data*[20]: This data set is provided by NII Testbeds and Community for Information access Research. The data set contains 3,496,252 unexamined Japanese patent applications and 1,315,470 grant patents of United States from 1993 to 2002. It is used to evaluate techniques related to patent mining such as patent retrieval, patent classification, and cross-language mining.
- *WIPO Patent Data*[21]: This patent collection is created by Fall, et al [23; 24], which aims to provide benchmark data for automatic patent classification. The data set contains about 75,000 patent applications in English (called WIPO-alpha) and 110,000 patent applications in German (called WIPO-de) from 1998 to 2002. Each patent application file consists of bibliographic data, abstract, claims, and description.
- *MAREC Patent Data*[22]: MAtrixware REsearch Collection (MAREC) is a standard patent data collection provided by Information Retrieval Facility for research purpose. It consists of over 19 million patent application and grated patents (1976-2008) from multiple authorities in 19 languages, the majority being English, German and French. MAREC has a wide usage in different areas such as patent information processing, patent retrieval, and patent translation.
- *ESPACE EP Patent Data*[23]: ESPACE EP is created by EPO, and consists of two sets of patent documents (EP-A and EP-B). Both patent collections contain bibliographic data, full text and embedded facsimile images of European patent documents from 1978 to 2006. The difference is that EP-A are patent applications, whereas EP-B are granted patents. These two patent collections are often used to carry out state-of-the-art searches on EP documents.

B. GLOSSARY

WIPO	World Intellectual Property Organization
USPTO	United States Patent and Trademark Office
EPO	European Patent Office
PCT	Patent Cooperation Treaty
IPC	International Patent Classification
USPC	United States Patent Classification
NTCIR	NII Testbeds and Community for Information access Research
CLEF	Conference and Labs of the Evaluation Forum
TREC	Text REtrieval Conference
USPAT	U.S. Patent Document Copies
IBM_TDB	IBM Technical Disclosure Bulletin
TRIZ	Theory of Inventive Problem Solving
IRF	Information Retrieval Facility

[20] http://research.nii.ac.jp/ntcir/permission/ntcir-8/perm-en-PATMN.html.

[21] http://www.wipo.int/classifications/ipc/en/ITsupport/Categorization/dataset/index.html.

[22] http://www.ir-facility.org/prototypes/marec.

[23] http://www.epo.org/searching/subscription/ep.html.

A Social Formalism and Survey for Recommender Systems

Daniel Bernardes, Mamadou Diaby*, Raphaël Fournier,
Françoise Fogelman-Soulié, Emmanuel Viennet

Université Paris 13, Sorbonne Paris Cité, L2TI, 93430, Villetaneuse, France.
* Work4 Labs, 3 Rue Moncey, 75009, Paris, France

{daniel.bernardes, mamadou.diaby, raphael.fournier, soulie,
emmanuel.viennet}@univ-paris13.fr

ABSTRACT

This paper presents a general formalism for Recommender Systems based on Social Network Analysis. After introducing the classical categories of recommender systems, we present our Social Filtering formalism and show that it extends association rules, classical Collaborative Filtering and Social Recommendation, while providing additional possibilities. This allows us to survey the literature and illustrate the versatility of our approach on various publicly available datasets, comparing our results with the literature.

Keywords

Recommender systems; Social Network Analysis; Collaborative Filtering; Social Recommenders.

1. INTRODUCTION

Recommender Systems (RSs) help users or groups of users deal with information overload by proposing to them items suited to their interests. The history of RSs started in the late 1990s with work by the GroupLens team at University of Minnesota [18] to recommend news, and by MovieLens in 1996 to recommend movies, which demonstrated that automated recommendations were very well received by users. Then Amazon, which had been incorporated in 1994, published its patent in 2001 and has been serving recommendations ever since, acting as a *de facto* reference showcase for the efficiency of RSs [33]. The Netflix competition (2006-2009) attracted over 41,000 participating teams [6] and turned RS into a hot topic among researchers.

The first papers on collaborative filtering showed how to use the opinions of similar users to recommend items to the active user [1]. Since then, research in RSs has become very active (see for example a recent RS survey including more than 250 references [8]) and RSs have been successfully used in many industry sectors to recommend items: movies (Netflix [6], MovieLens [41]), products (Amazon.com [33], La Boîte à Outils [47]), songs [3], jobs to Facebook users (Work4Labs.com [16]), books, friends, banners or content on a social site (Skyrock.com [44]) etc.

RSs exploit various sources of information: about users (their demographics), about products (their features) and about user interactions with the products [8; 25], either *explicit* (rating, satisfaction) or *implicit* (product purchased, book read, song heard, content clicked etc.) More recently, Social networks and social media (blogs, social tagging sites, etc) have emerged and become very active. A social site will allow users to construct profiles (public or semi-public), to share connections with other users and to view and traverse lists of connections made by others in the system [10]. Authors [8; 51; 56; 57] have thus proposed to also include information from social media (Facebook, Twitter, ...) because social relations obviously influence users' behaviors. There are two visions for social recommendation [56]:

- The narrow definition only considers RSs which combine users' data and data from their social relationships. It is the one most used in the literature. In this case, we need an *explicit social network* such as, for example, Facebook, to provide the social relationships;
- In a broader definition, a social RS is any recommendation system targeting social media (blogs, social tagging, video sharing, etc.) or even ecommerce. In this case, we might not have an explicit social network, but could still derive an *implicit social network*.

RSs implementations are based on various techniques [1]: content-based, collaborative filtering (both memory- and model-based), hybrid or social [56]. Performances are evaluated through various criteria [52].

In this paper, we propose a Social Filtering formalism (SF), based on Social Network Analysis (SNA), which allows us to describe, within the same formalism, both association rules, traditional Collaborative Filtering (CF) and Social Recommendation (SR), while providing additional ways to implement a RS, thus producing novel RSs.

The paper can thus be read as a survey on RSs, with experiments illustrating the various RSs presented. We have not tried, in this paper, to optimize hyper-parameters, but rather intended to present a wide repertoire of RSs tested in a uniform setting to allow for comparisons which are usually hard to make since, in the literature, each paper has its own settings and hyper-parameters choices.

The paper is organized as follows: in section 2, we introduce general concepts and notations, and we review traditional techniques in section 3. In section 4, we introduce our Social Filtering formalism and show in section 5 how it relates to conventional approaches. In section 6, we introduce evaluation metrics and various representative datasets. In section 7, we present extensive experimental results to illustrate the various RSs presented in the paper: we reproduce known results from the literature, and add new results, showing the benefits brought by our unifying formalism. Our conclusion identifies remaining issues and perspectives in section 8.

2. NOTATIONS

RSs use available data to generate lists of recommendations. Depending on the application, data can involve:

- **Usage**: the user visits a site and performs various actions on items shown (clicks, insertion-into-cart, purchases) which are collected in the site log files. Such actions provide feedback on the user's interest in a given item, which are essentially positive (the fact that the user did not act on the item at that time does not necessarily mean he did not like it). This data is often called *implicit* because the user did not generate it intentionally, but it comes as a side effect of the user's actions.
- **Ratings**: the user posts his evaluations or "likes" of items displayed on the site. These ratings can appear as stars, numbers or even comments. They can be positive or negative. Most users will not leave any evaluation and those who do might widely differ in their rating ways. Ratings are said to be *explicit* data, since the user intentionally acted to provide them.
- **Additional data**: in most cases, many more data sources exist on items and users.
  - Items: items such as products, contents or banners usually have associated attributes, such as for example the title and author of a book. Obviously these attributes are important to understand whether a user could be interested in the item.
  - Users: various types of information might be available, from a mere IP address to cookies to detailed attributes of registered customers (name, address, history of purchases, etc). Knowing the user in details should help building personalized recommendations for him/her. Additionally, information from social media (friends, followers, etc) might also be available.
  - Data from the RS: the RS itself produces ordered lists of items recommended to the user who might like them later (clicks, purchases, etc. will indicate that).

Usage and rating data can be represented by an *interaction* (or *preferences*) matrix R which encodes the actions of users on items. Table 1 below, for example, shows 4 users who rated 5 movies (ratings are shown by numbers). In the case where the user's interaction is a purchase or a click, matrix R is binary, with 1 indicating the item was purchased and 0 it was not. In cases where repeated consumption is possible, values in matrix R indicate the number of times the item was consumed, thus leading to R having the same structure as for ratings. In the following, we will use the word *consume* to indifferently mean rate, purchase or click.

	Monuments Men	**Django Unchained**	**Forrest Gump**	**Gran Torino**	**Pulp Fiction**
Amy	3			2	5
Paul		4			2
Rob	5	4	1		
Liz	2		3		

Table 1: Interaction matrix R in the case of ratings.

It should be noted that collecting implicit user's behavior is usually easier than requiring the user to provide explicit feedback or provide access to his / her social network. Most users purchase but very few items and rate even less: as few as 1% of users who consume an item might also rate it. As a result, matrix R is often very sparse, even more so in the case of ratings.

We will denote by L (resp. C) the total number of users or lines in R (resp. total number of items or columns). Matrix R is thus of dimensions L x C. Usually, matrix R is very large: a few millions × a few tens-hundreds of thousands.

3. STATE-OF-THE-ART

RSs have been studied for more than 15 years now [1; 8]. Traditional techniques are grouped into content-based, Collaborative Filtering, hybrid, and, more recently, social [8], which we describe in this section.

3.1 Content-based

Content-based RS [1; 35; 46] use items (or users) descriptions to define items' (or users) profiles. A user is then recommended items which profiles best match the items he best rated in the past, or items which users with most similar profiles best rated in the past. Sometimes, users only provide descriptions (instead of ratings), in this case items' profiles are constructed using these descriptions [16]. To implement Content-based RSs, we need a similarity measure (among items or users) and profiles comparison (for example k-nearest neighbors).

3.2 Collaborative Filtering

Collaborative Filtering is certainly the most widely used technique for implementing RSs. There exist two main groups of CF techniques: memory-based (or neighborhood methods [1]), and model-based (or latent factor models [42]).

As stated earlier in the introduction, CF methods use the opinion of a group of similar users to recommend items to the active user [1; 27; 32; 42; 54]. According to [42], the two key assumptions behind these systems are:

- Users who had similar tastes in the past will have similar tastes in the future;
- Users' preferences remain stable and consistent over time.

In the literature, there exist two main groups of CF techniques [59]: model-based or latent factor models [42; 34; 2] and memory-based or neighborhood methods [1; 32].

3.2.1 *Model-based methods*

Model-based RSs [11; 14; 29; 54; 60; 61] estimate a global model, through machine learning techniques, to produce unknown ratings. This leads to models that neatly fit data and therefore to RSs with good quality. However, learning a model may require lots of training data which could be an issue in some applications. In the literature many model-based CF systems have been proposed:

- Reference [11] proposes a probabilistic model in which ratings are integer valued; the model is then learnt using Bayesian networks;
- Reference [61] designs a CF system based on support vector machine (SVM) by iteratively estimating all missing ratings using an heuristic;
- Reference [29] develops a neural network-based collaborative method: a user-based and an item-based methods;
- Reference [28] describes various methods used for the Netflix prize[1] and in particular the matrix factorization methods which provided the best results.

One of the most efficient and best used model-based methods is matrix factorization [42; 59] in which users and items are represented in a low-dimensional latent factors space. The new representations of users ($\hat{U}$) and items ($\hat{I}$) are commonly computed by minimizing the regularized squared error [59]:

$$\min_{\hat{U},\hat{I}} \sum_{u,i} \left[\left(r_{u,i} - \hat{v}_u^T \hat{v}_i \right)^2 + \lambda_1 \parallel \hat{v}_u \parallel^2 + \lambda_2 \parallel \hat{v}_i \parallel^2 \right] \qquad (1)$$

where λ_1 and λ_2 are regularization parameters, $r_{u,i}$ is the rating that user u gave to item i, $\hat{v}_u$ and $\hat{v}_i$ are the new representations of user u and item i respectively, $\hat{U}$ and $\hat{I}$ are the new representation sets of the sets of users and items, respectively. Once the new representations of users and items $\hat{v}_u$ and $\hat{v}_i$ have been computed, we can obtain the predicted rating $\hat{r}_{u,i}$ as follows:

$$\hat{r}_{u,i} = \hat{v}_u^T \hat{v}_i \qquad (2)$$

Matrix factorization methods can be generalized to probabilistic models called Probabilistic Matrix Factorization [42; 50; 59]. These techniques are more suited to explicit feedback cases. They usually produce very good results but suffer from extremely large sizes of matrix R [3].

3.2.2 *Memory-based methods*

Memory-based CF techniques rely on the notion of similarity between users or items to build neighborhoods methods.

3.2.2.1 *Similarity.*

If a is the active user for whom we seek recommendations, u another user and i and j two items, we will denote:

- $I(a)$, $I(u)$ and $I(a\&u) = I(a) \cap I(u)$ the sets of items consumed by a, u, both a and u respectively.
- $U(i)$, $U(j)$ and $U(i\&j) = U(i) \cap U(j)$ the set of users who consumed i, j and both i and j, respectively.
- $\overrightarrow{l}(u)$ the line of matrix R for user u and $\overrightarrow{c}(i)$ its column for item i,
- $\bar{l}(u)$ the average of $\overrightarrow{l}(u)$ (average rating given by u or average number of items consumed by u) and $\bar{c}(i)$ the average of $\overrightarrow{c}(i)$ (i's average rating or average number of users who consumed i).

$$\bar{l}(u) = \frac{1}{C} \sum_{i=1}^{C} r_{ui} \qquad \bar{c}(i) = \frac{1}{L} \sum_{u=1}^{L} r_{ui}$$

The similarity between users a and u can be defined through many similarity measures, for example cosine, Pearson correlation coefficient (PCC) [1] or asymmetric cosine [3] similarities (equations (3), (4) & (5) below respectively):

$$\begin{aligned} \mathrm{Sim}(a,u) &= \cos\left[\overrightarrow{l}(a), \overrightarrow{l}(u)\right] \\ &= \frac{\sum_{i=1}^{C} r_{ai} \times r_{ui}}{\sqrt{\sum_{i=1}^{C} (r_{ai})^2} \sqrt{\sum_{i=1}^{C} (r_{ui})^2}} \end{aligned} \qquad (3)$$

$$\begin{aligned} \mathrm{Sim}(a,u) &= \mathrm{PCC}\left[\overrightarrow{l}(a), \overrightarrow{l}(u)\right] \\ &= \frac{\sum_{i \in I(a) \cap I(u)} \left[r_{ai} - \bar{l}(a)\right]\left[r_{ui} - \bar{l}(u)\right]}{\sqrt{\sum_{i \in I(a) \cap I(u)} \left[r_{ai} - \bar{l}(a)\right]^2} \sqrt{\sum_{i \in I(a) \cap I(u)} \left[r_{ui} - \bar{l}(u)\right]^2}} \end{aligned} \qquad (4)$$

$$\begin{aligned} \mathrm{Sim}(a,u) &= \text{asym-cos}_\alpha\left[\overrightarrow{l}(a), \overrightarrow{l}(u)\right] \\ &= \frac{\sum_{i=1}^{C} r_{ai} \times r_{ui}}{\left[\sum_{i=1}^{C} r_{ai}^2\right]^{\alpha} \times \left[\sum_{i=1}^{C} r_{ui}^2\right]^{1-\alpha}} \end{aligned} \qquad (5)$$

Note that, in the binary case, asymmetric cosine for $\alpha = \frac{1}{2}$ is equivalent to cosine similarity. The similarity between items i and j can be defined in the same fashion: cosine, Pearson correlation coefficient or asymmetric cosine (equations (6), (7) & (8) below respectively):

$$\begin{aligned} \mathrm{Sim}(i,j) &= \cos\left[\overrightarrow{c}(i), \overrightarrow{c}(j)\right] \\ &= \frac{\sum_{u=1}^{L} r_{ui} \times r_{uj}}{\sqrt{\sum_{u=1}^{L} (r_{ui})^2} \sqrt{\sum_{u=1}^{L} (r_{uj})^2}} \end{aligned} \qquad (6)$$

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

$$\begin{aligned}\mathrm{Sim}(i,j) &= \mathrm{PCC}\Big[\overrightarrow{c}(i), \overrightarrow{c}(j)\Big]\\ &= \frac{\displaystyle\sum_{u\in U(i)\cap U(j)} \big[r_{ui}-\bar{c}(i)\big]\big[r_{uj}-\bar{c}(j)\big]}{\sqrt{\displaystyle\sum_{u\in U(i)\cap U(j)}\big[r_{ui}-\bar{c}(i)\big]^2}\sqrt{\displaystyle\sum_{u\in U(i)\cap U(j)}\big[r_{uj}-\bar{c}(j)\big]^2}}\end{aligned} \tag{7}$$

$$\begin{aligned}\mathrm{Sim}(i,j) &= \text{asym-cos}_\alpha\Big[\overrightarrow{c}(i), \overrightarrow{c}(j)\Big]\\ &= \frac{\displaystyle\sum_{u=1}^{L} r_{ui}\times r_{uj}}{\Big[\displaystyle\sum_{u=1}^{L} r_{ui}^2\Big]^\alpha \times \Big[\displaystyle\sum_{u=1}^{L} r_{uj}^2\Big]^{1-\alpha}}\end{aligned} \tag{8}$$

It should be noted that both users and items similarity measures above take into account the actions on all items (resp. of all users): this is why these measures are called *collaborative.*

3.2.2.2 Collaborative Filtering scores.

CF techniques produce, for an active user a, a list of recommended items ranked through a *scoring function* (or aggregation function), which takes into account either users most similar to a (user-based CF) or items most similar to those consumed by a (item-based CF).

Let us thus denote $K(a)$ the *neighborhood* of a and $V(i)$ the neighborhood of item i. These neighborhoods can be defined in many ways (for example, N-nearest neighbors, for some given N, or neighbors with similarity larger than a given threshold, using user/item similarity).

The score functions are then defined for users and items as:

$$\begin{aligned}\mathrm{Score}(a,i) &= \sum_{u\in K(a)} r_{ui}\times f\big[\mathrm{Sim}(a,u)\big]\\ \mathrm{Score}(a,i) &= \sum_{j\in V(i)} r_{aj}\times g\big[\mathrm{Sim}(i,j)\big]\end{aligned} \tag{9}$$

where various functions f and g can be used [1]:

- For user-based CF: average rating (or popularity) of item i by neighbors of a in $K(a)$, weighted average rating and normalized average rating of nearest users weighted by similarity to a (from top to bottom in equation (10) below):

$$\begin{aligned}\mathrm{Score}(a,i) &= \frac{1}{\mathrm{card}\big[K(a)\big]}\sum_{u\in K(a)} r_{ui}\\ \mathrm{Score}(a,i) &= \frac{\displaystyle\sum_{u\in K(a)} r_{ui}\times\mathrm{Sim}(a,u)}{\displaystyle\sum_{u\in K(a)\cap U(i)}|\mathrm{Sim}(a,u)|}\\ \mathrm{Score}(a,i) &= \bar{l}(a) + \frac{\displaystyle\sum_{u\in K(a)\cap U(i)}\big(r_{ui}-\bar{l}(u)\big)\times\mathrm{Sim}(a,u)}{\displaystyle\sum_{u\in K(a)\cap U(i)}|\mathrm{Sim}(a,u)|}\end{aligned} \tag{10}$$

- For item-based CF: average rating by a of items neighbors of i in $V(i)$, weighted average rating, normalized average rating weighted by their similarity with i:

$$\begin{aligned}\mathrm{Score}(a,i) &= \frac{1}{\mathrm{card}\big[V(i)\big]}\sum_{j\in V(i)} r_{aj}\\ \mathrm{Score}(a,i) &= \frac{\displaystyle\sum_{j\in V(i)} r_{aj}\times\mathrm{Sim}(i,j)}{\displaystyle\sum_{j\in V(i)\cap I(a)}|\mathrm{Sim}(i,j)|}\\ \mathrm{Score}(a,i) &= \bar{c}(i) + \frac{\displaystyle\sum_{j\in V(i)\cap I(a)}\big(r_{aj}-\bar{c}(j)\big)\times\mathrm{Sim}(i,j)}{\displaystyle\sum_{j\in V(i)\cap I(a)}|\mathrm{Sim}(i,j)|}\end{aligned} \tag{11}$$

Another mechanism has been developed [3] to produce locality instead of explicitly defining neighborhoods. Functions f and g are defined so as to put more emphasis on high similarities (with high q, q'):

$$\begin{aligned}\mathrm{Score}(a,i) &= \sum_{u\in K(a)} r_{ui}\times\big[\mathrm{Sim}(a,u)\big]^q\\ \mathrm{Score}(a,i) &= \sum_{j\in V(i)} r_{aj}\times\big[\mathrm{Sim}(i,j)\big]^{q'}\end{aligned} \tag{12}$$

For $q=0$, this is equivalent to average rating, and for $q=1$, this is similar to weighted average rating.

We then rank items i by decreasing scores and retain the top k items $(i_1^a, i_2^a, ..., i_k^a)$ which are recommended to a, such that:

$$\mathrm{Score}(a, i_1^a) \geq \mathrm{Score}(a, i_2^a) \geq ... \geq \mathrm{Score}(a, i_k^a) \tag{13}$$

3.2.2.3 Conclusion on CF techniques.

Notice that while memory-based techniques produce ranked lists of items, model-based techniques predict ratings, through a score which can be used also to rank recommendations. In practice, all CF systems suffer from several drawbacks:

- New user/item: collaborative systems cannot make accurate recommendation to new users since they have not rated a sufficient number of items to determine their preferences. The same problem arises for new items, which have not obtained enough ratings from users. This problem is known as the *cold start* recommendation problem;

- Scalability: memory-based systems generally have a scalability issue, because they need to calculate the similarity between all pairs of users (resp. items) to make recommendations;

- Sparsity: the number of available ratings is usually extremely small compared to the total number of pairs user - item; as a result the computed similarities between users and items are not stable (adding a few new ratings can dramatically change similarities) and so predicted ratings are not stable either;

- Information: of course memory-based and model-based techniques use very limited information, namely ratings/purchases only. They could not use content on users or items, nor social relationships if these were available.

The representation of users and items in a low dimensional space in latent factor models mitigates the cold start recommendation problem but raises a scalability issue. In general, latent factor methods are known to generally yield better results than neighborhood methods [28; 42; 59].

3.3 Social Recommender Systems

Traditional RSs, and in particular model-based systems, rely on the (often implicit) assumption that users are independent, identically distributed (i.i.d). The same holds for items. However, this is not the case on social networks where users enjoy rich relationships with other members on the network. It has long been observed in sociology [40] that users' "friends" on such networks have similar taste (homophily). It is thus natural that new techniques [65] extended previous RSs by making use of social network structures. However, it was realized that the type of interaction taken into account could have a dramatic impact on the quality of the obtained social recommender [65]. In this section, we review three families of social recommender: one based on explicit social links, one based on trust and an emerging family based on implicit links.

3.3.1 Social Recommender Systems based on explicit social links

In this section, we assume that users are connected through explicit relationships such as friend, follower etc. Unsurprisingly, with the recent thrive of online social networks, it has been found that users prefer recommendations made by their friends than those provided by online RSs, which use anonymous people similar to them [53]. Most Social RSs are based on CF methods: social collaborative recommenders, like traditional CF systems, can be divided into two families: memory-based and model-based systems.

3.3.1.1 Memory-based Social Recommender.

Memory-based methods in social recommendation are similar to those in CF (presented in section *3.2.2*), the only difference being the use of explicit social relationship for computing similarities.

- In [64; 65] the authors present their social-network-based CF system (SNCF), a modified version of the traditional user-based CF and test it on Essembly.com[2] which provides two sorts of links: friends and allies.
  - In [64], they use a graph theoretic approach to compute users' similarity as the minimal distance between two nodes (using Dijkstra's algorithm for instance), instead of using the ratings' patterns as in traditional CF; it is assumed that the influence will exponentially decay as distance increases. They show that this method produces results worse than traditional CF;
  - In [65], the user's neighborhood is just simply its set of friends in the network (first circle). This approach provides results slightly worse than the best CF. But the computation load is much reduced: from computing the similarity of all pairs of users to just looking for the user's friends. This is a dramatic improvement to the scalability issues of CF. They also show that if the allies are used instead of friends, then the results are as good as CF, but at a much reduced computation cost.
- In [24], authors observe on a dataset from Yelp[3] that friends tend to give restaurant ratings (slightly) more similar than non-friends. However, immediate friends tend to differ in ratings by 0.88 (out of 5), which is rather similar to results in [65]. Their experimental setup compares their model-based algorithm (probabilistic), a Friends Average approach (which only averages the ratings of the immediate friend), a Weighted Friends (more weight is given to friends which are more similar according to cosine-similarity), a Naive Bayes approach and a traditional CF method. All methods which use the influences from friends achieve better results than CF in terms of prediction accuracy.
- Reference [12] presents SaND (Social Network and Discovery), a social recommendation system; SaND is an aggregation tool for information discovery and analysis over the social data gathered from IBM Lotus Connections' applications. For a given query, the proposed system combines the active user's score, scores from his connections and scores between terms and the query;
- Reference [51] proposes two social recommendation models: the first one is based on social contagion while the second is based on social influence. The authors define the social contagion model as a model to simulate how an opinion on certain items spreads through the social network;
- Reference [19] proposes a group recommendation system in which recommendations are made based on the strength of the social relationships in the active group. This strength is computed using the strengths of the social relationship between pairwise social links (scaled from 1 to 5 and based on daily contact frequency).

3.3.1.2 Model-based Social Recommenders.

Model-based methods in social recommenders represent users and items into a latent space vector (as described in section *3.2.1*) making sure that users' latent vectors are close to those of their friends.

- Reference [4] combined matrix factorization and friendship links to make recommendations: the recommendation score for the active user is the sum of the scores of his friends.
- Reference [38] proposes algorithms which yield better results than non-negative matrix factorization [31], probabilistic matrix factorization [42] and a trust-aware recommendation method [37]. It presents two social RSs:
  - A matrix factorization making sure that the latent vector of a given user is close to the weighted average latent vectors of his friends;

[2] `http://www.essembly.com`

[3] `http://www.yelp.com`

- A matrix factorization minimizing the difference between a user's and his friends' latent vectors individually.

Finally, a few social RSs combine social and content-based techniques. For example, [17] proposes two ways to aggregate users' preferences with those of their friends: enrich users' profiles with those of their friends or aggregate users' recommendation scores with those of their social relationships.

3.3.2 Trust and influence-based social Recommender Systems

As explained in [38] "trust relationships" are different from "social relationships" in many respects. Trust-aware RSs are based on the assumption that users have taste similar to other users they trust, while in social RSs, some of the active user's friends may have totally different tastes from him [38]. This was also observed in [65], with the differences between friends and allies, which represents a case where trust is explicitly provided by users.

In everyday life, people may ask other people (friends, relatives, someone they trust) for a recommendation. If the person cannot provide sufficient information, she may indicate another person whom she knows which could, and so on. The notion of trust network arises naturally: one tend to have faith in the opinion of people trusted by the people he trusts himself, transitively. Conversely, the notion of social influence has long been used in marketing, relying on the assumption that users are likely to make choices similar to their role-models [49]. The notion of influence can be seen as close to that of trust: when providing a friend with a referral, a trusted user influences her friend. It has long been known that this "word-of-mouth effect" can be used commercially, such as for example in viral marketing.

Recently, it was attempted to incorporate trust or influence knowledge into RSs. Beyond the mere expected increase in efficiency, computing trust may also alleviate recurrent problems of traditional RSs, such as data sparsity, cold start or shilling attacks (fake profile injections) to bias recommendations.

3.3.2.1 Trust computation.

The trust relationship is directional, i.e. the fact that user u_1 trusts user u_2 at some level t does not necessarily mean that u_2 trusts u_1 at the same or another level. Trust can be represented by a binary value, 0 for "not trusted user" and 1 for "trusted user", or through more gradual scales [20; 21; 39] or even with a probabilistic approach [15; 48]. Some models include an explicit notion of distrust [21; 67], but most of them ignore it.

For RSs, trust is computed over an explicit social network to increase the information available to generate recommendations. There exist two cases in the literature: either trust is provided explicitly in a trust network, or it has to be inferred.

In an explicit trust network, we propagate and aggregate trust to infer long chains of trust [67; 20]. Trust computation also requires an aggregation strategy, to combine estimates obtained from different paths from one user to another. Several operators may be used like minimum, maximum, (un)weighted sum and average. Different strategies may also be applied: propagate trust first, then aggregate; or aggregate first, then propagate (the latter allowing easier distributed computation).

In non-explicit trust networks, trust has to be inferred. For example, in [45], the author defines a profile and item-level trust, based on correct previous recommendations.

3.3.2.2 Trust-enhanced Recommender Systems.

In explicit trust networks, users provide the system with trust statements for their peers, be it on a gradual scale (Moleskiing [5]), allies (Essembly [65]) or lists of trusted and non-trusted people (Epinions [39]). Then, for the recommendation to a specific user u, trust is estimated between u and the relevant users, in order to weigh the recommendation computation, either through trust-based weighted mean (rating from user u for item i is an ordinary weighted mean of the ratings of users which have evaluated i, where the weights are the trust estimates for these users) or Trust-based CF (as in classical CF methods, replacing similarity-based weights by trust-based weights obtained via propagation and aggregation strategies as described above).

3.3.3 Social Recommender Systems based on implicit social links

Recently, a new type of social RSs has been introduced which rely not upon an explicit social network (as in section *3.3.1*) but upon networks which can be derived from users' behaviors and have thus been named *implicit networks*. Users will be – implicitly – connected if, for example, they take pictures in the same locations [23], they attend the same events or click on the same ads [44]. The implicit users' social network can then be used, as in section *3.3.1*) to build recommendations.

For example, [57] extracts from cooking recipes bipartite graph (*recipes*, *ingredients*) and sets the weight of the link in the ingredients network as the point-wise mutual information of the ingredients, extremities of the link. Authors then apply a discriminative machine learning method (stochastic gradient boosting trees), using features extracted from the ingredients network, to predict recipe ratings and recommend recipes. Results show that the structural features extracted from the ingredient networks are most critical for performances.

Similar approaches have been developed for RSs where no ratings are provided, but only information on whether objects were collected (product purchased, banner clicked, movie watched, song listened to).

- Reference [66] uses a resource-allocation process in the object network to define the weight on the links and an aggregate score : they show that their technique is more efficient on the MovieLens dataset than traditional CF;

- Reference [44] shows an example to recommend ads banners: for an active user, the similarity among banners is measured by the number of users who clicked on both. Then traditional item-based CF is applied. Authors show a 20-fold increase in number of cumulated clicks with the social RS compared to the random selection of 5 recommendations.

3.3.4 Conclusion

Social RSs are still relatively new. There is a lot of active research in this area and it should be expected that new results will extend the field of traditional systems to incorporate social information of all sorts. In particular, the field of social recommenders built on implicit social networks seems particularly promising and we will now dig deeper in this direction to produce our Social Filtering formalism.

4. SOCIAL FILTERING

Our **Social Filtering formalism** (SF) is based upon a bipartite graph and its projections (see [22; 66] for a discussion of bipartite graphs). A *bipartite graph* is defined over a set of nodes separated into two non-overlapping subsets: for example, users and items, items and their features, etc. A link can only be established between nodes in different sets: a link connects a user to the items she has consumed. The bipartite network is then projected into two (unipartite) networks, one for each set of nodes: a Users' and an Items' networks. In the projection (see Figure 1), two nodes are connected if they had common neighbors in the bipartite graph. The link weight can be used to indicate the number of shared neighbors. For example, two users are linked if they have consumed at least one item in common (we usually impose a more stringent condition: at least K items). The projected networks can thus be viewed as the network of users consuming at least K same items (users having the same preferences) and the network of items consumed by at least K' same users (items liked by the same people).

Projected networks can then be used to define neighborhoods [55] or recommendation algorithms which perform better than conventional CF on the MovieLens dataset [66]. This generic formalism extends these early contributions: we are able to reproduce results from various classical approaches, and we also provide new approaches, allowing more flexibility and potential for improved performances, depending on the dataset.

In the SF formalism, as in traditional CF, we build recommendations by defining neighborhoods and scoring functions.

4.1 Similarity

4.1.1 Support-based similarity

In the case of implicit feedback (binary interaction matrix R), in essence, the link between two users a and u (resp. i and j) represents an association rule $a \to u$ (resp. $i \to j$) with the link weight proportional to the rule support, where support of rule $a \to u$ (resp. $i \to j$) is defined as:

$$\text{Supp}(a \to u) = \frac{\#\text{ Items cons. by } a \text{ and } u}{\#\text{ Items}} = \frac{1}{C}\sum_{i=1}^{C} r_{ai}r_{ui}$$
$$\text{Supp}(i \to j) = \frac{\#\text{ Users who cons. } i \text{ and } j}{\#\text{ Users}} = \frac{1}{L}\sum_{u=1}^{L} r_{ui}r_{uj}$$

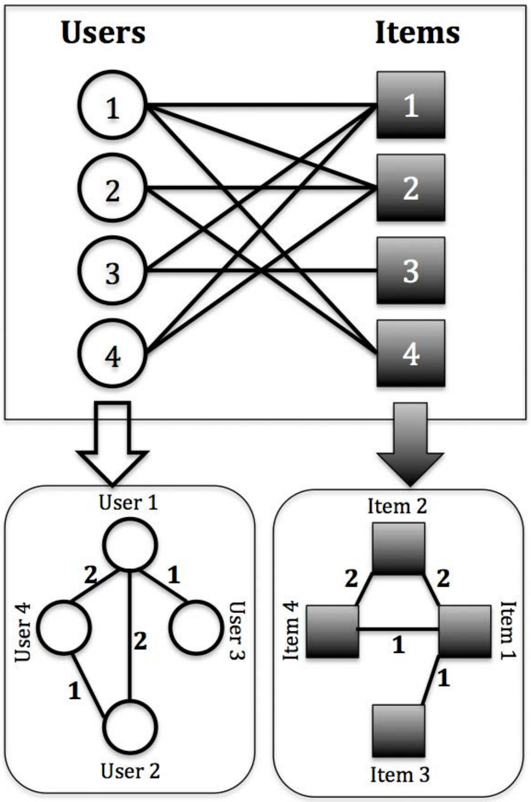


Figure 1: Bipartite graph and projections

In the case of a non-binary matrix R (ratings), support is similarly defined. Hence, *Support* is defined in general as:

$$\begin{aligned}\text{Supp}(a \to u) &= \frac{1}{C}\sum_{i=1}^{C} r_{ai}r_{ui} \\ \text{Supp}(i \to j) &= \frac{1}{L}\sum_{u=1}^{L} r_{ui}r_{uj}\end{aligned} \qquad (14)$$

Support is similar to cosine similarity (equations (3) and (6)), so that we can use support as a similarity measure. We will define support-based similarity of users (resp. items) as:

$$\begin{aligned}\text{Sim}(a, u) &= \text{Supp}(a \to u) \\ \text{Sim}(i, j) &= \text{Supp}(i \to j)\end{aligned} \qquad (15)$$

4.1.2 Confidence-based similarity

In the case of implicit feedback, the confidence of link $a \to u$

(resp. $i \to j$) is defined, as for association rules, by:

$$\mathrm{Conf}(a \to u) = \frac{\#\text{ Items cons. by } a \text{ and } u}{\#\text{ Items cons. by } a} = \frac{\sum_{i=1}^{C} r_{ai} r_{ui}}{\sum_{i=1}^{C} r_{ai}}$$

$$\mathrm{Conf}(i \to j) = \frac{\#\text{ Users who cons. } i \text{ and } j}{\#\text{ Users who cons. } i} = \frac{\sum_{u=1}^{L} r_{ui} r_{uj}}{\sum_{u=1}^{L} r_{ui}}$$

For ratings (non-binary matrix R), confidence will be similarly defined. Hence, *Confidence* is defined in general as:

$$\begin{aligned} \mathrm{Conf}(a \to u) &= \frac{\sum_{i=1}^{C} r_{ai} r_{ui}}{\sum_{i=1}^{C} r_{ai}} \\ \mathrm{Conf}(i \to j) &= \frac{\sum_{u=1}^{L} r_{ui} r_{uj}}{\sum_{u=1}^{L} r_{ui}} \end{aligned} \tag{16}$$

This is again similar to cosine similarity. We can thus also use confidence as a similarity measure. Confidence-based similarity of users (resp. items) is defined as:

$$\begin{aligned} \mathrm{Sim}(a, u) &= \mathrm{Conf}(a \to u) \\ \mathrm{Sim}(i, j) &= \mathrm{Conf}(i \to j) \end{aligned} \tag{17}$$

4.1.3 *Asymmetric confidence-based similarity*

We might want to define similarity between a and u (resp. i and j) from both $a \to u$ and $u \to a$ links, which is not the case for confidence. Following [3], we thus define the following Asymmetric Confidence-based Similarity of users/items, where α is a parameter to be tuned by cross-validation:

$$\begin{aligned} \mathrm{Sim}(a, u) &= \left[\mathrm{Conf}(a \to u)\right]^{\alpha} \left[\mathrm{Conf}(u \to a)\right]^{(1-\alpha)} \\ \mathrm{Sim}(i, j) &= \left[\mathrm{Conf}(i \to j)\right]^{\alpha} \left[\mathrm{Conf}(j \to i)\right]^{(1-\alpha)} \end{aligned} \tag{18}$$

This measure is identical to asymmetric cosine, generalizes confidence similarity of link $a \to u$ (for $\alpha = 0$) and of link $u \to a$ (for $\alpha = 1$) and, in the case where matrix R is binary, cosine similarity as well (for $\alpha = 0.5$).

4.1.4 *Jaccard Index-based similarity*

Jaccard Index [36] measures the similarity of lists by counting how many elements they have in common. The Jaccard Index of users a and u (resp. items i and j) is defined (in the binary case) as:

$$\mathrm{Jaccard}(a, u) = \frac{Card\left[\overrightarrow{l}(a) \cap \overrightarrow{l}(u)\right]}{Card\left[\overrightarrow{l}(a) \cup \overrightarrow{l}(u)\right]}$$

$$\mathrm{Jaccard}(i, j) = \frac{Card\left[\overrightarrow{c}(i) \cap \overrightarrow{c}(j)\right]}{Card\left[\overrightarrow{c}(i) \cup \overrightarrow{c}(j)\right]}$$

According to the definition of the Jaccard index, we thus have for users (and similarly for items):

$$\mathrm{Jaccard}(a, u) = \frac{\sum_{i=1}^{C} r_{ai} \times r_{ui}}{\sum_{i=1}^{C} r_{ai} + \sum_{i=1}^{C} r_{ui} - \sum_{i=1}^{C} r_{ai} \times r_{ui}}$$

$$= \frac{\#\text{Items Cons. By a and u}}{(\#\text{Items Cons. By a}) + (\#\text{Items Cons. By u}) - (\#\text{Items Cons. By a and u})}$$

As above, Jaccard index will be similarly defined if matrix R is not binary. Hence, Jaccard index for users / items is defined as:

$$\begin{aligned} \mathrm{Jaccard}(a, u) &= \frac{\sum_{i=1}^{C} r_{ai} \times r_{ui}}{\sum_{i=1}^{C} r_{ai} + \sum_{i=1}^{C} r_{ui} - \sum_{i=1}^{C} r_{ai} \times r_{ui}} \\ \mathrm{Jaccard}(i, j) &= \frac{\sum_{u=1}^{L} r_{ui} \times r_{uj}}{\sum_{u=1}^{C} r_{ui} + \sum_{u=1}^{C} r_{uj} - \sum_{u=1}^{C} r_{ui} \times r_{uj}} \end{aligned} \tag{19}$$

We then define the Jaccard-based Similarity of users / items as:

$$\begin{aligned} \mathrm{Sim}(a, u) &= \mathrm{Jaccard}(a, u) \\ \mathrm{Sim}(i, j) &= \mathrm{Jaccard}(i, j) \end{aligned} \tag{20}$$

4.2 Neighborhoods

Now that we have defined various similarity measures, we can define the neighborhood $K(a)$ of user a (resp. $V(i)$ of item i): we only give the definition for users, items are similar) in any of the following ways (all users or only the Top N or only those with similarity measure larger than a threshold can be chosen):

- $K(a)$ is the first circle of user a in the Users graph, where neighbors of a in the circle can be rank-ordered by any of the previously defined similarity measures: support-based; confidence-based; asymmetric confidence-based; Jaccard Index-based.
- $K(a)$ is the local community of user a in the Users graph, where local communities are defined as in [43].
- $K(a)$ is the community of user a in the Users graph (where communities are defined, for example, by maximizing modularity [22]).

These last two cases are completely novel ways to define neighborhoods: they exploit the homophily expected in social networks (even implicit as here), where users with the same behavior tend to connect and be part of the same community. In CF, users in $K(a)$ (with cosine similarity) are such that they have consumed at least one item in common (otherwise their cosine would be 0); in the SF setting such users would be linked in the Users graph ($K \geq 1$). In the above community-based definitions of $K(a)$, users in $K(a)$ might not be directly connected to active user a. These definitions thus embody some notion of paths linking users, through common usage patterns.

4.3 Social Collaborative Filtering scores

We now define scoring functions as for Collaborative filtering (equations (10), (11) and (12) above) with the various social similarity measures (equations (15), (17), (18) and (20)) and the various neighborhoods we have defined.

- For user-based SF: average rating (or popularity) of item i by neighbors of a in $K(a)$, weighted average rating, normalized average rating of nearest users weighted by similarity to a, as in equation (10) above.

- <u>For item-based SF</u>: average rating by a of items neighbors of i in $V(i)$, weighted average rating, normalized average rating of nearest items weighted by their similarity with i, as in equation (11) above.

As in equation (12), scores with locality parameters q and q' can be used with any of the similarity measures defined above.
We then rank-order items i by decreasing scores as in equation (13) and retain the top k $(i_1^a, i_2^a, \ldots, i_k^a)$ recommended to a.

5. SOCIAL FILTERING AND OTHER RSs

To implement the SF framework, we need to build the bipartite network and project it to unipartite networks, after choosing adequate parameters K and K' (Figure 1) and eliminating mega-hubs if necessary [44]. Then, depending upon the choice of similarity measure (equations (15), (17), (18) and (19)), neighborhoods and score function (equations (10), (11) or (12)) we obtain a RS which is equivalent to one of the following classical recommender systems:

- **<u>Association rules</u>** [9] of length 2 can be used for recommendation [47]. They are obtained from the item-based SF formalism with $K = K' = 1$, asymmetric confidence-based similarity with $\alpha = 0, N = 1$ ($V(i)$ is reduced to the first nearest neighbor) and local scoring function (equation (12)) with $q' = 1$.

- **<u>CF</u>** is obtained with $K = K' = 1$, asymmetric confidence with $\alpha = 0.5$ (cosine similarity) and the usual CF score functions (identical to those used by SF).

- **<u>CF with locality</u>** [3] is obtained with $K = K' = 1$, asymmetric confidence and score function as in equation (12).

- **<u>Social RS</u>**: if an explicit social network is available, such as a friendship network for example, then one can use that network as a Users' graph and proceed as in the SF framework. In [65], the authors use the first circle as neighborhood and show that their results are slightly worse than conventional CF, but at a much reduced computational cost.

- **<u>Content-based RS</u>**: instead of building a bipartite Users x Items graph, one could use the exact same methodology and build a bipartite Users x Users' attributes or Items x Items' attributes graph and recommend items liked by similar users or items similar to those consumed by the user, with a similarity measure based on the projected graphs.

As can be seen above, the SF formalism generalizes various well established recommendation techniques. However, it offers new possibilities as well: the Social Filtering formalism thus extends content-based, association rules and CF, with new similarity measures and new ways to define neighborhoods.

By only building once one bipartite graph and the projected unipartite graphs, we have at our disposal, in a unique framework, a full set of similarity measures, neighborhood definitions and scoring functions; thus allowing us to produce many different RSs, evaluate their performances and select the best.

We will illustrate in section 7 performances on various standard datasets, comparing our SF formalism to conventional techniques and showing original results produced within our SF framework.

6. EVALUATION

6.1 Performance metrics

A RS can be asked either to predict a rating (in the case of explicit feedback) or to rank items (in the case of implicit feedback). The two visions are quite different and globally correspond to model-based and memory-based approaches in CF respectively [1]. *Performance metrics* differ in these two cases [52].
<u>Predictive case</u>: we want to evaluate how close predicted rating $\hat{r}_{ij}$ is to actual rating r_{ij}. We thus use classical performance metrics from machine learning:

- AUC: Area Under the Curve [52].

- RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) defined as:

$$\begin{aligned} MAE &= \frac{1}{I} \sum_{i,j} |r_{ij} - \hat{r}_{ij}| \\ RMSE &= \sqrt{\frac{1}{I} \sum_{i,j} (r_{ij} - \hat{r}_{ij})^2} \end{aligned} \tag{21}$$

where I is the total number of tested ratings.
<u>Ranking case</u>: in that case, we want to evaluate whether recommended items were adequate for the user; for example, recommended items were later consumed. We thus have a Target set for each user which represents the set of items he consumed after being recommended. This can be implemented by splitting the available dataset into Training / Testing subsets (taking into account time stamps if available). In this case, metrics are those classically used in information retrieval:

- *Recall@k* and *Precision@k* are defined as:

$$\begin{aligned} \text{Recall@}k &= \frac{1}{L} \sum_{a} \frac{Card(R_a \cap T_a)}{Card(T_a)} \\ \text{Precision@}k &= \frac{1}{L} \sum_{a} \frac{Card(R_a \cap T_a)}{k} \end{aligned} \tag{22}$$

  where $R_a = (i_1^a, i_2^a, ..., i_k^a)$ is the set of k items recommended to a, T_a is the target set for a. We can also plot *Recall@k* as a function of the number of recommended items and compute the AUC for that curve.

- F_β-mesure: F_β is designed to take into account both recall and precision; F_1 is the most commonly used. F_β is defined as :

$$F_\beta@k = \frac{(1 + \beta^2) \times prec@k \times recall@k}{(\beta^2 \times prec@k) + recall@k} \tag{23}$$

- MAP@k (Mean Average Precision) was used, for example, in the Million Song Dataset challenge[4]; it is defined as:

$$\text{MAP@}k = \frac{1}{L}\sum_{a=1}^{L}\frac{1}{k}\sum_{i=1}^{k}\frac{C_{ai}}{i}1_{ai} \tag{24}$$

  where C_{ai} is the number of correct recommendations to user a in the first i recommendations ($Precision@i$ for user a) and $1_{ai} = 1$ if item at rank i is correct (for user a), 0 otherwise.

In practical situations, we are also interested in more *qualitative indicators* showing whether all users (resp. items) get recommendations (resp. are recommended) or which items are recommended: as we will see, some RSs might be better on performance indicators and poorer on these qualitative indicators and it will be the user's choice to trade performance decrease for qualitative indicators increase.

- **Users' coverage**: this is the proportion of users who get recommendations:

$$\text{UsersCoverage@}k = \frac{\#\text{ Users in Test with k Reco}}{L_{\text{test}}} \tag{25}$$

  where L_{test} is the number of users in the test set.

- When Users' coverage is partial, we want to know what the **average number of recommendations** was for the users with partial lists:

$$\text{AvNbRec@}k = \sum_{K=0}^{k-1} K \frac{\#\text{ Users in Test with k Reco}}{L_{\text{test}} - \#\text{ Users with k Reco}} \tag{26}$$

- **Items' coverage**: higher diversity (recommended items are varied) should result in more attractive recommendations [52]. We thus want to have a high coverage, i.e. a high proportion of items which get recommended. When this is evaluated on a test set (e.g. 10% of users), the Items' coverage will appear lower than if we were evaluating it on the full population of users. Yet, since all other indicators are evaluated on the test set, for homogeneity reasons, we will also report the evaluation of Items' coverage of the lists of recommended items for users in the test set.

$$\text{ItemsCoverage@}k = \frac{\#\text{ Items in Reco Lists}}{C} \tag{27}$$

- **Head/Tail coverage**: if we rank items by decreasing popularity (number of users who purchased the item), we call *Head* the 20% of items with highest popularity and *Tail* the remaining 80% [47]. Recommending only most popular items will result in relatively poor performances and low diversity. We thus define the rate of recommended items in the *Head* and in the *Tail*:

$$\text{RateHead@}k = \frac{1}{L_{\text{test}}}\sum_{u\in\text{Test}}\frac{\#\text{ Reco for u in Head}}{\#\text{ Reco for u}}$$
$$\text{RateTail@}k = \frac{1}{L_{\text{test}}}\sum_{u\in\text{Test}}\frac{\#\text{ Reco for u in Tail}}{\#\text{ Reco for u}} \tag{28}$$

  where the sum runs on the L_{test} users in the Test set.

Many more indicators are described in [52], but we will only use these for the experiments described in section 7.

6.2 Data sets

To demonstrate the generality of our framework, we present results on various datasets traditionally used in the literature. These datasets, shown in Table 2 are characterized by the number of users, items, preferences (*implicit* shown by a count of usage or *explicit* shown by ratings) and, in some cases, existing *explicit* social relationships.

Dataset	Preferences	Preferences Type	Users	Items	Explicit Social
LastFM	92,834	Count	1,892	17,632	25,434
MovieLens1M	1 M	Ratings	6,040	3,883	–
Flixster	8.2 M	Ratings	1 M	49,000	26.7 M
MSD	48 M	Count	1.2 M	380,000	–

Table 2: Datasets.

- **Lastfm**: this dataset[5] contains 92, 834 listening information counts of 17, 632 artists by 1, 892 users of the website Lastfm.com. There is an explicit friends' network with 25, 434 links.
- **Flixster**: this dataset[6] contains 8.2 M ratings of about 49, 000 movies by about 1 M users. There is an explicit friends' network with 26.7 M links.
- **MovieLens1M**: this dataset[7] contains 1, 000, 209 ratings of approximately 3, 900 movies made by 6, 040 users who joined MovieLens in 2000.
- **Million Song Dataset** (MSD): this dataset was used for a Kaggle challenge[4]. It contains 48 M listening counts of 380, 000 songs by 1.2 M users.

Some performance results of RSs on these datasets are already available in the literature. See for example:

- Lastfm: user-based CF and inferred explicit trust network [63];
- Flixster: user-based CF and matrix factorization [26];
- MovieLens1M: user-based CF and local scoring function with asymmetric confidence in [3]; CF and matrix-factorization [62];
- MSD: CF and local scoring function with asymmetric confidence [3].

7. EXPERIMENTS

We have implemented our SF formalism on the various datasets presented above. The goal of these experiments is **not** to demonstrate that our formalism produces better results than other RSs: this would have required systematic runs of multiple splits and optimization of parameter choices;

[4] http://kaggle.com/c/msdchallenge

[5] http://files.grouplens.org/datasets/hetrec2011/hetrec2011-lastfm-2k.zip

[6] http://www.cs.ubc.ca/~jamalim/datasets/

[7] http://files.grouplens.org/datasets/movielens/ml-1m.zip

whereas we only performed one run without optimizing parameters.

We rather intend to show the *versatility* of our formalism which provides many different ways to assemble the various ingredients and produce old and new RSs. We also give details, which are not always present in the literature, about the settings we used for the experiments. To allow for comparison, our code is available in open source[8].

Our experiments will be presented in two sets:

- First, we show that our SF formalism produces results identical to those found by conventional techniques, demonstrating that this formalism can be particularized into some of the existing techniques. Since results reported in the literature use various evaluation metrics and settings (training/test splits for example), we have defined homogeneous settings and reproduced classical methods to compare them to our formalism;
- Then, we show cases where our SF formalism generates new RSs.

In the experiments reported below, we have used various settings which are not always explicit in the literature:

- **Binarization of inputs**
  - Counts: we split the interval of counts into 10 bins with the same number of elements. We then replace each count in matrix R by the bin index;
  - Ratings: we transform ratings into value from 1 to 10 (for example rating 1 to 5 stars is multiplied by 2);
  - Then we transform values $r \in [0, 10]$ into a binary value. We have used the same setting as [3]: 10 is coded as 1, all others as 0. Then all users, resp. items, with their entire line, resp. column, at 0 are eliminated, which is a very drastic reduction of the number of users and items.
- **Data split**: to evaluate performances, we split data into two data sets, one used for training, the other for testing. We implemented the same technique as in [3] for the MSD challenge: take all users, randomize and split: 90% users for training and 10% kept for testing.
  - For training, we use all transactions of the 90% users.
  - We test on the remaining transactions of the test users: we input 50% of the transactions of each test user, and compare the obtained recommendations list to the remaining 50%.
- **Choices of similarity measure, neighborhood and scoring function**. These will be varied, producing the various RSs we want to implement.

For reference, to compare performances obtained in our experiments, we have implemented three classical techniques:

- **Popularity**: we rank items by decreasing popularity (the sum of 1s in the matrix R item column); popularity served as baseline in the MSD challenge[4];
- **Bigrams**: we used the apriori algorithm for association rules [9] with length 2 and thresholds on support and confidence at 1% and did not try to further optimize the settings; items are ranked in decreasing confidence of the rule generating them;
- **NMF (non-negative matrix factorization)**: we used the code[9] associated to [28], with maximal rank 100 and maximum number of iterations 10,000 and did not try to further optimize the settings.

Performances shown in Table 3 provide a baseline: figures in bold indicate the best performance of the corresponding category. We run our simulations on an Intel Xeon E7-4850 2,00 GHz (10 cores, 512 GB RAM), shared with members of the team (so concurrent usage might have happened in some of the experiments, with impact on reported time). Computing time in hours is thus indicative only (0:01:00 is 1 min, 4:20:00 is 4 hours 20 min). Our formalism was implemented using state-of- the-art libraries, such as Eigen[10] for computing similarities.

As can be seen, bigrams are very efficient in terms of performances on all four datasets (see also [47]) and they require no parameters tuning (except the support and confidence threshold). But bigrams have low Users' and Items' coverage, with all recommended items in the Head.
In contrast, NMF are very sensitive to parameters (maximal rank and maximum number of iterations) and since we did no try to optimize these parameters, we obtain low performances here. In addition, NMF do not scale well with increasing size of datasets. On the other hand, NMF have best Users' and Items' coverage (note that after 4 days of computing time, we stopped NMF on MSD). These two techniques illustrate the trade-off one has to make in practice: fine tune parameters vs. default parameters to obtain optimal performances, and performances vs. coverage. Finally scalability is indeed a critical feature.

7.1 SFs reproduce classical RSs

We have implemented CF with the code provided by the author[11] in [3], in 2 versions:

- **Classical CF** [1] with cosine similarity and average score. Results are shown in Table 4, lines CF IB (item-based) and CF UB (user-based);
- **CF with locality** [3]: we exactly reproduced the results shown in [3] for $k = 500$ recommendations as in the MSD challenge (we do not show these results here). We show results in Table 4 for $k = 10$ recommendations (columns CF IB_Aiolli) for values $q = 5$ and $\alpha = 0$. These results are coherent with those presented in [3].

[8] https://bitbucket.org/danielbernardes/socialfiltering

[9] https://github.com/kimjingu/nonnegfac-python

[10] http://Eigen.tuxfamily.org

[11] Code on http://www.math.uni.pd.it/~aiolli/CODE/MSD

Data sets	**LastFM**			**Flixster**			**MovieLens1M**			**MSD**		
Methods	**Popularity**	**Bigrams**	**NMF**	**Popularity**	**Bigrams**	**NMF**	**Popularity**	**Bigrams**	**NMF**	**Popularity**	**Bigrams**	**NMF**
MAP @ 10	0.058	**0.144**	0.005	0.038	**0.098**	0.005	0.097	**0.149**	0.008	0.022	**0.135**	*
Prec @ 10	0.053	**0.082**	0.048	0.061	**0.120**	0.075	0.150	**0.206**	0.141	0.017	**0.042**	*
Recall @ 10	0.165	**0.260**	0.161	0.092	**0.126**	0.116	0.120	**0.155**	0.113	0.055	**0.175**	*
Users Full Coverage	**100%**	52.86%	**100%**	**100%**	74.00%	79.65%	**100%**	99.50%	**100%**	**100%**	0%	*
Users Partial Coverage	0%	47%	0%	0%	26.00%	20.35%	0%	0.50%	0%	0%	100%	*
– Avg. num. of recs.	-	3.4	-	-	2.3	2.8	-	5.0	-	-	1.9	*
Items coverage	0.46%	1.13%	**2.87%**	0.06%	0.33%	**0.60%**	0.62%	**3.47%**	**3.47%**	**0.008%**	0.001%	*
– Proportion in Tail	0%	0%	**1.95%**	**0%**	**0%**	**0%**	0%	**0.30%**	0.01%	**0.00%**	**0.00%**	*
– Proportion in Head	100%	100%	**98.05%**	**100%**	**100%**	**100%**	**100%**	**99.70%**	99.99%	**100.00%**	**100.00%**	*
Computation time	**0:01:00**	0:05:00	1:00:00	**0:01:00**	0:05:00	4:00:00	**0:01:00**	0:05:00	2:00:00	**0:14:52**	0:30:00	> 4 days

Table 3: Performances of reference techniques.

Note that in Table 4, the neighborhood chosen simply consists of all users / items with non null similarity (the case with $q = 5$ restores some locality). We found that limiting to a Top N neighbors (we just tested $N = 100$) usually resulted in decreased performances and coverage. Note that, with the datasets sizes we use, a relative difference of 1% is significant.

We have then implemented our SF formalism to reproduce classical RSs:

- **Association rules**: we obtained, as expected, the exact same performances as those shown in Table 3;
- **Item-based CF**: we again obtained, as expected, from the SF formalism, the exact same results as those in Table 4 (lines CF IB and CF UB with cosine similarity, all users/items for neighborhoods $K(a)$ and $V(i)$ and weighted average score).
- **CF with locality**: we implemented our SF framework, with settings described in section 5 ($K = K' = 1$). The projected networks (and not the explicit network in the case of Lastfm) can be used with different similarity measures and scoring functions, producing various RSs. We again obtained, as expected, from the SF formalism the exact same results as those in Table 4 (lines CF IB Aiolli and CF UB Aiolli).

When comparing CF IB, CF UB, with cosine or Aiolli asymmetric confidence, we do not find one technique systematically best:

- **For CF IB**, asymmetric confidence has better performances and poorer Items' coverage than cosine on datasets Lastfm and Flixster; but on MovieLens and MSD, cosine is better for performances.
- **For CF UB**, asymmetric confidence is better than cosine for all datasets except MovieLens.
- **CF IB outperforms CF UB** for all datasets except Flixster (both in terms of performance and Users' and Items' coverage).

Note that parameters ($\alpha = 0, q = 5$) have not been optimized the way they were in [3]: it would certainly be possible to choose, by cross-validation, better parameters. But, as we said, the purpose of these experiments is not to fully optimize settings.

Comparing Table 3 and Table 4 shows that bigrams seem to perform best (except for MovieLens), but their Users' coverage is poor (because the filter on support and confidence limits the number of rules: the threshold was not optimized). These results show that our formalism covers these various classical techniques: association rules, CF (item or user-based) and CF with locality as in [3].

7.2 SF produces new RSs

We now show in Tables 5 and 6 implementations of the SF formalism with various choices of similarity measures (equations, (15), (17), (18) and (20)) and neighborhood $K(a)/V(i)$. We use as score function the weighted average (equations (10) and (11) middle) or, for the asymmetric confidence similarity, the *local score* of equation (12). In our implementation, we filter the links in the Users' and Items' networks with support / confidence less than 1%.

We thus have three differences with the classical implementations of CF above: one is the choice of neighborhood (first circle or local community in the implicit Users or Items network, instead of most similar users/items for CF); next one is the filtering of links in SF which results in a reduction in the numbers of neighbors; and last one is the choice of various similarity measures such as support, confidence, or Jaccard (equations (15), (17) and (20)). Here are our main findings:

- **Item-based Social Filtering**:
  - Results for SF IB with 1^{st} circle as neighborhood (in Table 5) show that the various similarity measures produce rather different performances. For all datasets, the best technique outperforms bigrams, while improving Users' coverage and Items' coverage. In addition, our implementation of SF IB, compared to CF IB, for the parameters of Table 4 ($\alpha = 0, q = 5$) shows improved performance for all datasets, except MovieLens. Note that Jaccard similarity usually produces poor performances but good Users' and Items' coverage and also succeeds in recommending items in the Tail. Except for MovieLens, where Jaccard similarity provides the best performances on all indicators.
  - Results for SF IB with local community as neighborhood are shown in Table 6. Because local communities are not really relevant in implicit networks, we just run experiments on Lastfm and Flixster. The use of local communities in these

Datasets	LastFM		Flixster		MovieLens1M		MSD	
Methods	**CF IB** (cosine, average)	**ACF IB** (α=0, q=5)	**CF IB** (cosine, average)	**ACF IB** (α=0, q=5)	**CF IB** (cosine, average)	**ACF IB** (α=0, q=5)	**CF IB** (cosine, average)	**ACF IB** (α=0, q=5)
MAP @ 10	0.066	**0.107**	0.082	0.082	**0.170**	0.147	0.075	**0.078**
Prec @ 10	0.054	**0.072**	0.090	**0.091**	**0.223**	0.199	**0.057**	0.053
Recall @ 10	0.154	**0.240**	0.164	0.164	**0.169**	0.150	0.158	**0.160**
Users Full Coverage	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.79%	**100.00%**
Users Partial Coverage	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.21%	**0.00%**
– Avg. num. of recs.	-	-	-	-	-	-	4.83	-
Items coverage	**22.16**%	12.44%	**15.24**%	5.40%	**10.61%**	6.80%	**42.02%**	22.88%
– Proportion in Head	**66.97%**	87.75%	**92.00%**	99.00%	**98.50%**	99.86%	**66.20%**	95.45%
– Proportion in Tail	**33.03%**	12.25%	**8.00%**	1.00%	**1.50%**	0.14%	**33.80%**	4.55%
Computation time	0:05:00	0:05:00	2:30:00	2:30:00	0:05:00	0:05:00	72:00:00	72:00:00
Methods	**CF UB** (cosine, average)	**ACF UB** (α=0, q=5)	**CF UB** (cosine, average)	**ACF UB** (α=0, q=5)	**CF UB** (cosine, average)	**ACF UB** (α=0, q=5)	**CF UB** (cosine, average)	**ACF UB** (α=0, q=5)
MAP @ 10	0.050	**0.071**	0.084	**0.089**	0.062	**0.105**	**0.069**	0.058
Prec @ 10	0.059	**0.062**	0.087	**0.093**	0.131	**0.161**	**0.044**	0.037
Recall @ 10	**0.185**	0.176	0.176	**0.188**	0.119	0.119	**0.144**	0.119
Users Full Coverage	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Users Partial Coverage	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
– Avg. num. of recs.	-	-	-	-	-	-	-	-
Items coverage	10.14%	**14.01%**	4.13%	**8.22%**	4.76%	**15.13%**	20.08%	**27.59%**
– Proportion in Head	89.76%	**80.62%**	99.70%	**99.43%**	99.76%	**95.34%**	96.97%	**82.22%**
– Proportion in Tail	10.24%	**19.38%**	0.30%	**0.57%**	0.24%	**4.66%**	4.03%	**17.77%**
Computation time	0:05:00	0:05:00	1:20:00	1:20:00	0:05:00	0:05:00	24:00:00	24:00:00

Table 4: Performances of Collaborative Filtering. CF and ACF are respectively the abbreviations of Collaborative Filtering and Asymmetric Cosine Collaborative Filtering.

cases results in a lower MAP, and generally degrades all performance metrics.

- **User-based Social Filtering**:
  - Results for SF UB in Table 5 vary depending on the dataset: sometimes the best SF UB (usually with asymmetric confidence) is better than CF UB or bigrams; sometimes it is worse. For MSD, SF UB has poorer MAP but better precision and recall than best CF IB or bigram.
- **Social RS**: we implemented our SF formalism using the explicit networks (Flixster and Lastfm) for Users' graph; the neighborhood is the 1st Circle of friends, Communities (computed with Louvain's algorithm [7]) or Local Communities (computed with our algorithm in [43]) and the score function is the average. Results in Table 7 show, as usual, various situations: on Lastfm 1st Circle has best performances, but Community better Users' coverage and Local Community better Items' coverage. For Flixster, Community is best, but 1st Circle has better Items' coverage and Local Community better Tail coverage.

7.3 Ensemble methods

As was demonstrated in many cases, and very notably in the Netflix challenge [6; 30], ensembles of RSs often get improved performances. We have thus implemented a few ensembles using a basic combination method, originated from [13], and used in [3]. To combine N recommenders, we assemble, for each user a, the N lists of k recommended items produced by the N RSs.

Let us denote $(i_1^n, \ldots, i_k^n)$ the list of items recommended to a (omitted in the notation) by RS n (with $n = 1, ..., N$). Some of the lists might have less than k elements. For each list, we assign points to the items in each list as follows: 1st item gets k points, 2nd item $k - 1$, etc. The lists are then fused and each item gets the sum of points in the various lists, with the ties resolved through a priority on RSs (the winner comes from the highest priority list).

In order to explore the potential of ensemble techniques, we tried combinations of pairs of RSs: we tested combinations of some of the best SF-based recommenders mentioned above (Tables 4, 5 and 6). Results shown in Table 8 are encouraging:

- For Lastfm, the best two-systems ensemble is CF-IB with Cosine similarity and weighted average score, combined with SF-IB with asymmetric confidence similarity and local score ($q = 1, \alpha = 0$), which gets a $MAP@10 = 0.16$. This performance is not better than the one obtained with a unique recommender.
- For Flixster, we obtain an improvement: combining SF-UB with asymmetric confidence similarity and local score ($q = 1, \alpha = 0$) and SF-UB with asymmetric confidence similarity and local score ($q = 5, \alpha = 0$) leads to a $MAP@10$ of 0.175, while the best performance was SF-UB with asymmetric confidence similarity and local score ($q = 5, \alpha = 0$) at $MAP@10 = 0.157$.

These preliminary results indeed confirm the potential of ensemble methods. They could certainly be enhanced by testing combinations of more systems, optimizing the parameters and implementing more adequate aggregation methods (Borda's aggregation method [13] seems rather popular in the literature, but can certainly be improved upon to merge recommendation results).

8. CONCLUSION

We have presented a simple and generic formalism based upon social network analysis techniques: by building once the projected Users' and Items' networks, the formalism allows reproducing a wide range of RSs found in the literature while also producing new RSs. This unique formalism thus provides very efficient ways to test the performances of many different RSs on the dataset at hand to select the most adequate in that case.

LastFM	**Item-based**							**User-based**						
Similarity Neighborhood = 1st circle	**Support**	**Confidence**	**Asym. Cos (q=1)**		**Asym. Cos (q=5)**		**Jaccard**	**Support**	**Confidence**	**Asym. Cos (q=1)**		**Asym. Cos (q=5)**		**Jaccard**
			α=0	**α=0.5**	**α=0**	**α=0.5**				**α=0**	**α=0.5**	**α=0**	**α=0.5**	
MAP @ 10	0.145	0.082	**0.161**	0.157	0.151	0.148	0.082	**0.062**	**0.062**	0.050	0.057	0.040	0.042	0.051
Prec @ 10	0.087	0.067	**0.087**	0.084	0.089	0.076	0.053	**0.130**	**0.130**	0.107	0.127	0.083	0.093	0.063
Recall @ 10	0.280	0.188	**0.280**	0.269	**0.283**	0.250	0.164	**0.208**	**0.208**	0.193	0.203	0.144	0.158	0.200
Users Full Coverage	56.45%	56.45%	56.45%	56.45%	56.45%	56.45%	**96.72%**	96.66%	96.66%	96.66%	96.66%	87.50%	96.66%	**96.72%**
Users Partial Coverage	43.00%	43.00%	43.00%	43.00%	43.00%	43.00%	3.28%	3.33%	3.33%	3.33%	3.33%	12.50%	3.33%	3.28%
– Avg. num. of recs.	3.9	3.9	3.9	3.9	3.9	3.9	3.8	9.0	9.0	9.0	9.0	7.3	9.0	4.8
Items coverage	0.92%	1.13%	0.96%	1.06%	0.96%	1.03%	**22.97%**	4.75%	4.75%	5.42%	5.25%	4.71%	5.82%	**10.32%**
– Proportion in Head	100%	100%	100%	100.00%	100.00%	100.00%	**60%**	99.33%	99.33%	94.66%	96.00%	**93.27%**	94.00%	95.20%
– Proportion in Tail	0%	0%	0%	0%	0%	0%	**40%**	0.66%	0.66%	5.33%	4.00%	**6.72%**	6.00%	4.79%
Computation time	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00

MovieLens1M	**Item-based**							**User-based**						
Similarity Neighborhood = 1st circle	**Support**	**Confidence**	**Asym. Cos (q=1)**		**Asym. Cos (q=5)**		**Jaccard**	**Support**	**Confidence**	**Asym. Cos (q=1)**		**Asym. Cos (q=5)**		**Jaccard**
			α=0	**α=0.5**	**α=0**	**α=0.5**				**α=0**	**α=0.5**	**α=0**	**α=0.5**	
MAP @ 10	0.127	0.091	0.133	0.154	0.144	0.144	**0.168**	0.097	0.084	0.093	0.075	**0.129**	0.088	0.062
Prec @ 10	0.172	0.151	0.182	0.208	0.196	0.203	**0.224**	0.225	0.209	0.203	0.175	**0.239**	0.191	0.130
Recall @ 10	0.133	0.104	0.141	0.159	0.148	0.150	**0.166**	**0.033**	0.031	0.030	0.025	0.032	0.027	0.119
Users Full Coverage	99.16%	99.16%	99.16%	99.16%	99.16%	99.16%	**100.00%**	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Users Partial Coverage	0.84%	0.84%	0.84%	0.84%	0.84%	0.84%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
– Avg. num. of recs.	5.0	5.0	5.0	5.0	5.0	5.0	-	-	-	-	-	-	-	-
Items coverage	1.70%	13.43%	1.92%	4.42%	2.48%	5.85%	**13.55%**	4.36%	4.39%	3.99%	4.08%	2.75%	4.08%	**5.45%**
– Proportion in Head	100%	100%	100%	100%	100%	100%	**98%**	96%	96%	95.63%	**95.00%**	100.00%	95.00%	99.73%
– Proportion in Tail	0%	0%	0%	0%	0%	0%	**2%**	4%	4%	4.38%	**5.00%**	0.00%	5.00%	0.27%
Computation time	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	0:15:00	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	0:30:00

Flixster	**Item-based**							**User-based**						
Similarity Neighborhood = 1st circle	**Support**	**Confidence**	**Asym. Cos (q=1)**		**Asym. Cos (q=5)**		**Jaccard**	**Support**	**Confidence**	**Asym. Cos (q=1)**		**Asym. Cos (q=5)**		**Jaccard**
			α=0	**α=0.5**	**α=0**	**α=0.5**				**α=0**	**α=0.5**	**α=0**	**α=0.5**	
MAP @ 10	0.079	0.074	0.080	0.084	**0.105**	0.096	0.078	0.124	0.107	**0.157**	0.141	**0.157**	0.134	0.041
Prec @ 10	0.100	0.096	0.102	0.104	**0.119**	0.117	0.086	0.279	0.250	0.279	0.250	**0.318**	0.250	0.069
Recall @ 10	0.113	0.097	0.115	0.112	0.126	0.117	**0.156**	0.005	0.005	0.005	0.005	0.006	0.005	**0.168**
Users Full Coverage	71.10%	71.10%	71.10%	71.10%	82.31%	71.10%	**100.00%**	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Users Partial Coverage	28.90%	28.90%	28.90%	28.90%	17.69%	28.90%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
– Avg. num. of recs.	2.4	2.4	2.4	2.4	3.6	2.4	-	-	-	-	-	-	-	-
Items coverage	0.29%	0.43%	0.29%	0.39%	0.29%	0.39%	**18.01%**	0.26%	0.27%	0.25%	0.25%	0.24%	0.25%	**6.60%**
– Proportion in Head	100%	100%	100%	100%	100%	100%	**95.46%**	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	**99.69%**
– Proportion in Tail	0%	0%	0%	0%	0%	0%	**4.54%**	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	**0.31%**
Computation time	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	0:05:00	43:00:00	0:10:00	0:10:00	0:10:00	0:10:00	0:10:00	0:10:00	112:00:00

MSD	**Item-based**							**User-based**						
Similarity Neighborhood = 1st circle	**Support**	**Confidence**	**Asym. Cos (q=1)**		**Asym. Cos (q=5)**		**Jaccard**	**Support**	**Confidence**	**Asym. Cos (q=1)**		**Asym. Cos (q=5)**		**Jaccard**
			α=0	**α=0.5**	**α=0**	**α=0.5**				**α=0**	**α=0.5**	**α=0**	**α=0.5**	
MAP @ 10	**0.135**	0.134	0.135	0.134	0.135	0.134	0.078	**0.054**	0.054	0.052	0.053	0.049	0.049	0.028
Prec @ 10	**0.042**	0.042	0.042	0.042	0.042	0.042	0.059	**0.090**	0.090	0.089	0.090	0.083	0.084	0.038
Recall @ 10	**0.175**	0.175	0.175	0.175	0.175	0.175	0.166	**0.178**	0.178	0.175	0.178	0.168	0.166	0.116
Users Full Coverage	**0.00%**	0.00%	0.00%	0.00%	0.00%	0.00%	**99.58%**	**99.01**%	99.01%	99.01%	0.990	97.37%	0.990	86.70%
Users Partial Coverage	100.00%	100%	100%	100%	100%	100%	**0.42%**	0.99%	0.99%	0.99%	0.99%	2.63%	1.01%	13.30%
– Avg. num. of recs.	1.9	1.9	1.9	1.9	1.9	1.9	**5.1**	7.0	7.0	7.0	7.0	6.4	7.0	2.9
Items coverage	0.0012%	0.0012%	0.0012%	0.0013%	0.0013%	0.0012%	**46.30%**	7.81%	7.81%	8.74%	8.50%	8.94%	9.33%	**23.37%**
– Proportion in Head	100%	100%	100%	100%	100%	100%	**78%**	97.83%	97.84%	97.56%	97.62%	97.24%	97.39%	**95.80%**
– Proportion in Tail	0%	0%	0%	0%	0%	0%	**22%**	2.17%	2.16%	2.44%	2.38%	2.76%	2.61%	**4.20%**
Computation time	0:10:00	0:10:00	0:10:00	0:10:00	0:10:00	0:10:00	3:00:00	13:50:00	14:10:00	12:52:00	12:51:00	13:37:00	13:00:00	7:00:00

Table 5: Performances of Social Filtering (implicit) with 1st circle; Asym. Cos is the abbreviation for Asymmetric Cosine.

Datasets	**Lastfm**						**Flixster**					
Similarity Neighborhood = Local community	**Support**	**Confidence**	**Asym. Conf. q=1**		**Asym. Conf. q=5**		**Support**	**Confidence**	**Asym. Conf. q=1**		**Asym. Conf. q=5**	
			α=0	**α=0.5**	α=0	**α=0.5**			α=0	**α=0.5**	**α=0**	**α=0.5**
SF-IB												
MAP @ 10	0.151	0.089	0.151	**0.152**	0.149	**0.152**	0.064	0.058	0.066	0.066	**0.081**	**0.081**
Prec @ 10	**0.080**	0.064	**0.080**	0.076	**0.080**	0.073	0.079	0.076	0.082	0.080	0.096	**0.097**
Recall @ 10	**0.256**	0.198	**0.256**	0.243	**0.256**	0.240	0.073	0.065	0.076	0.072	**0.083**	0.079
Users Full Coverage	51.67%	51.67%	51.67%	51.67%	51.67%	51.67%	80.22%	80.22%	80.22%	80.22%	80.22%	80.22%
Users Partial Coverage	48.33%	48.33%	48.33%	48.33%	48.33%	48.33%	19.78%	19.78%	19.78%	19.78%	19.78%	19.78%
Av. nb of recommendations	4.00	4.00	4.00	4.00	4.00	4.00	3.13	3.13	3.13	3.13	3.13	3.13
Items coverage	0.74%	**0.92%**	0.74%	0.89%	0.82%	0.85%	0.30%	**0.35%**	0.30%	0.34%	0.29%	0.34%
Rate-Head	100.00%	100%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Rate-Tail	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Computation time	0:05:00	0:06:00	0:08:00	0:10:00	0:09:00	1:10:00	0:18:00	0:18:00	0:23:00	0:22:00	0:21:00	0:23:00

Table 6: Performances of Social Filtering (implicit) with Local Community.

Dataset	Lastfm			Flixster		
Perf. Indicator @10 SF-UB	**1st circle**	**Community**	**Local Comm.**	**1st circle**	**Community**	**Local Comm.**
MAP @ 10	**0.101**	0.083	0.078	0.019	**0.038**	0.016
Prec @ 10	**0.074**	0.064	0.053	0.037	**0.057**	0.027
Recall @ 10	**0.239**	0.182	0.203	0.046	**0.093**	0.036
Users Full Coverage	80.16%	**88.09%**	56.52%	84.16%	**99.87%**	61.45%
Users Partial Coverage	19.84%	11.91%	43.48%	15.85%	0.13%	38.55%
Av. nb of recommendations	4.76	4.66	4.375	3.85	3.2	3.71
Items coverage	13.54%	8.22%	**14.25%**	**8.01%**	0.37%	7.96%
Rate-Head	89.93%	93.77%	**83.96%**	99.53%	100.00%	**99.18%**
Rate-Tail	10.07%	6.22%	**16.04%**	0.47%	0.00%	**0.82%**
Computation time	0:05:00	0:05:00	0:05:00	0:05:00	58:00:00	0:05:00

Table 7: Performances of Social Filtering user-based on explicit networks.

Datasets	**Lastfm**	**Flixster**
Perf. Indicator @10	**(CF-IB Cosine, w. average) and (SF-IB Asym. Conf. q=1, α=0)**	**(SF-UB Asym. Conf. q=5, α=0) and (SF-UB Asym. Conf. q=1, α=0)**
MAP @ 10	0.160	0.175
Prec @ 10	0.089	0.307
Recall @ 10	0.298	0.006
Users Full Coverage	56.45%	100%
Users Partial Coverage	43.55%	0%
Av. nb of recommendations	3.9	-
Items coverage	0.96%	0.25%
Rate-Head	100%	100%
Rate-Tail	0%	0%
Computation time	0:01:00	0:06:00

Table 8: Ensemble of RSs.

As can be seen from our experiments, there is no unique silver bullet (so far!): for each dataset, one has to test and try a full repertoire of candidate RSs, fine tuning hyper-parameters (a topic we did not address in this paper) and selecting the best RS for the performance indicator he/she cares for. The richer the repertoire is, the more chances for the final RS to get best performances. This theoretical formalism thus provides a very powerful way to formalize and compare many RSs.

In addition, this integrated formalism enables the production of modular code, uncoupling the similarities and scoring functions computation steps. It also allows for elegant implementation of the recommendation engines, as demonstrated in our published open source code[12].

Computing the bipartite network and its projections requires significant computing resources. It can be considered as a set-up step for our recommender framework. This overhead is only worth it if one wants to produce an ensemble of RSs originating from various choices of parameters (similarity measures, neighborhoods and scoring functions) to make comparisons and select the best choice for the dataset at hand. In addition, computation of similarities is the bottleneck (since obviously it involves all pairs of users or items). However, some similarity measures (asymmetric confidence or Jaccard) are more costly than others (support, confidence and cosine) as the figures about computing time show in the various tables.

This work opens ways for future research in several directions:

- We have introduced various choices of similarity measures, neighborhood and scoring functions. Obviously, more choices can be designed and evaluated;

- Since it is easy to produce many RSs within the same framework, we could produce collections – or ensembles – of RSs working together to complement each other weaknesses. More ways for combining recommended lists will be needed to improve upon the Borda's mechanism discussed here. Inspiration from the literature on ensemble of rating-based RSs could certainly be useful.

- Since implicit and explicit social networks can be set into the same framework, further investigation is required on how to integrate implicit and explicit networks, thus producing hybrid Social recommenders;

- Recent work[13] allowing to merge user-based and item-based CF seem promising, and could certainly be framed into our formalism;

- The first phase in our formalism consists in projecting the Users and Items network. This step is computationally heavy. Hence, at the present time we cannot process extremely large datasets, such as for example in the MSD challenge. We thus intend to optimize our implementation of the Social Network projection. More generally, our code could be ported to distributed Hadoop-based environments to allow processing of larger datasets and parallel testing of the various hyper-parameters choices.

9. ACKNOWLEDGEMENTS

[12] https://bitbucket.org/danielbernardes/socialfiltering

[13] Personal communication of one of the authors [58].

This work has been partially supported by projects AMMICO and OFS (funded by BPI France). Access to public datasets and code is gratefully acknowledged.

10. REFERENCES

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

[2] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 19–28. ACM, 2009.

[3] F. Aiolli. Efficient top-n recommendation for very large scale binary rated datasets. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 273–280, New York, NY, USA, 2013. ACM.

[4] J. Aranda, I. Givoni, J. Handcock, and D. Tarlow. An online social network-based recommendation system. *Toronto, Ontario, Canada*, 2007.

[5] P. Avesani, P. Massa, and R. Tiella. A trust-enhanced recommender system application: Moleskiing. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 1589–1593. ACM, 2005.

[6] J. Bennett and S. Lanning. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35, 2007.

[7] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[8] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, 46:109–132, 2013.

[9] C. Borgelt. Frequent item set mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):437–456, 2012.

[10] D. M. Boyd and N. B. Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, 2008.

[11] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[12] D. Carmel, N. Zwerdling, I. Guy, S. Ofek-Koifman, N. Har'El, I. Ronen, E. Uziel, S. Yogev, and S. Chernov. Personalized social search based on the user's social network. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1227–1236. ACM, 2009.

[13] J.-C. de Borda. On elections by ballot. *Classics of social choice, eds. I. McLean, AB Urken, and F. Hewitt*, pages 83–89, 1995.

[14] L. M. de Campos, J. M. Fernández-Luna, and J. F. Huete. A collaborative recommender system based on probabilistic inference from fuzzy observations. *Fuzzy Sets and Systems*, 159(12):1554–1576, 2008.

[15] Z. Despotovic and K. Aberer. A probabilistic approach to predict peers' performance in p2p networks. In *Cooperative Information Agents VIII*, pages 62–76. Springer, 2004.

[16] M. Diaby, E. Viennet, and T. Launay. Toward the next generation of recruitment tools: an online social network-based job recommender system. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 821–828. ACM, 2013.

[17] M. Diaby, E. Viennet, and T. Launay. Exploration of methodologies to improve job recommender systems on social networks. *Social Network Analysis and Mining*, 4(1):1–17, 2014.

[18] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4(2):81–173, 2011.

[19] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, and K. Seada. Enhancing group recommendation by incorporating social relationship interactions. In *Proceedings of the 16th ACM international conference on Supporting group work*, pages 97–106. ACM, 2010.

[20] J. A. Golbeck. Computing and applying trust in web-based social networks. 2005.

[21] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*, pages 403–412. ACM, 2004.

[22] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral. Module identification in bipartite and directed networks. *Physical Review E*, 76(3):036102, 2007.

[23] M. Gupte and T. Eliassi-Rad. Measuring tie strength in implicit social networks. In *Proceedings of the 3rd Annual ACM Web Science Conference*, pages 109–118. ACM, 2012.

[24] J. He and W. W. Chu. *A social network-based recommender system (SNRS)*. Springer, 2010.

[25] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. IEEE, 2008.

[26] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142. ACM, 2010.

[27] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[28] J. Kim and H. Park. Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM Journal on Scientific Computing*, 33(6):3261–3281, 2011.

[29] M. W. Kim, E. J. Kim, and J. W. Ryu. A collaborative recommendation based on neural networks. In *Database Systems for Advanced Applications*, pages 425–430. Springer, 2004.

[30] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[31] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[32] D. Lemire and A. Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *SDM*, volume 5, pages 1–5. SIAM, 2005.

[33] G. Linden, B. Smith, and J. York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[34] N. N. Liu, M. Zhao, and Q. Yang. Probabilistic latent preference analysis for collaborative filtering. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 759–766. ACM, 2009.

[35] P. Lops, M. De Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

[36] L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.

[37] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 203–210. ACM, 2009.

[38] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[39] P. Massa and P. Avesani. Trust-aware collaborative filtering for recommender systems. In *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pages 492–508. Springer, 2004.

[40] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

[41] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl. Movielens unplugged: experiences with an occasionally connected recommender system. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 263–266. ACM, 2003.

[42] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.

[43] B. Ngonmang, M. Tchuente, and E. Viennet. Local community identification in social networks. *Parallel Processing Letters*, 22(01), 2012.

[44] B. Ngonmang, E. Viennet, S. Sean, F. Fogelman-Soulié, and R. Kirche. Monetization and services on a real online social network using social network analysis. In *Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*, pages 185–193. IEEE, 2013.

[45] J. O'Donovan and B. Smyth. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 167–174. ACM, 2005.

[46] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[47] B. Pradel, S. Sean, J. Delporte, S. Guérif, C. Rouveirol, N. Usunier, F. Fogelman-Soulié, and F. Dufau-Joel. A case study in a recommender system based on purchase data. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 377–385. ACM, 2011.

[48] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *The Semantic Web-ISWC 2003*, pages 351–368. Springer, 2003.

[49] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–70. ACM, 2002.

[50] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM, 2008.

[51] S. Shang, P. Hui, S. R. Kulkarni, and P. W. Cuff. Wisdom of the crowd: Incorporating social influence in recommendation models. In *Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on*, pages 835–840. IEEE, 2011.

[52] G. Shani and A. Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011.

[53] R. R. Sinha and K. Swearingen. Comparing recommendations made by online systems and friends. In *DELOS workshop: personalisation and recommender systems in digital libraries*, volume 106, 2001.

[54] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

[55] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.

[56] J. Tang, X. Hu, and H. Liu. Social recommendation: a review. *Social Network Analysis and Mining*, 3(4):1113–1133, 2013.

[57] C.-Y. Teng, Y.-R. Lin, and L. A. Adamic. Recipe recommendation using ingredient networks. In *Proceedings of the 3rd Annual ACM Web Science Conference*, pages 298–307. ACM, 2012.

[58] K. Verstrepen and B. Goethals. Unifying nearest neighbors collaborative filtering. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 177–184. ACM, 2014.

[59] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM, 2011.

[60] K. Wang and Y. Tan. A new collaborative filtering recommendation approach based on naive bayesian method. In *Advances in Swarm Intelligence*, pages 218–227. Springer, 2011.

[61] Z. Xia, Y. Dong, and G. Xing. Support vector machines for collaborative filtering. In *Proceedings of the 44th annual Southeast regional conference*, pages 169–174. ACM, 2006.

[62] B. Xu, J. Bu, C. Chen, and D. Cai. An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web*, pages 21–30. ACM, 2012.

[63] C. Yin, T. Chu, et al. Improving personal product recommendation via friendships' expansion. *Journal of Computer and Communications*, 1(05):1, 2013.

[64] R. Zheng, F. Provost, and A. Ghose. Social network collaborative filtering: Preliminary results. In *Proceedings of the Sixth Workshop on eBusiness WEB2007*, 2007.

[65] R. Zheng, D. Wilkinson, and F. Provost. Social network collaborative filtering. *Stern, IOMS Department, CeDER, Vol*, 2008.

[66] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang. Bipartite network projection and personal recommendation. *Physical Review E*, 76(4):046115, 2007.

[67] C.-N. Ziegler and G. Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7(4-5):337–358, 2005.

The Data Problem in Data Mining

Albrecht Zimmermann
LIRIS, INSA Lyon, France
albrecht.zimmermann@insa-lyon.fr

ABSTRACT

Computer science is essentially an *applied* or *engineering* science, creating tools. In Data Mining, those tools are supposed to help humans understand large amounts of data. In this position paper, I argue that for all the progress that has been made in Data Mining, in particular Pattern Mining, we are lacking insight into three key aspects: 1) How pattern mining algorithms perform quantitatively, 2) How to choose parameter settings, and 3) How to relate found patterns to the processes that generated the data. I illustrate the issue by surveying existing work in light of these concerns and pointing to the (relatively few) papers that *have* attempted to fill in the gaps. I argue further that progress regarding those questions is held back by a lack of data with varying, controlled properties, and that this lack is unlikely to be remedied by the ever increasing collection of real-life data. Instead, I am convinced that we will need to make a science of digital data generation, and use it to develop guidance to data practitioners.

1. INTRODUCTION

Computer science is basically an *applied* or *engineering* science. By this, I do not mean that all work done in our field does or should happen only in relation with a concretely defined real-life application. But rather that we use the results of other disciplines, be they mathematics, physics, or others, and develop what should be understood as *tools*, devices and algorithms that make it easier for humans to perform certain tasks.

In Data Mining, those tools come mainly in two forms: 1) *supervised* methods that learn from labeled data how to *predict* labels for unseen data or how to *characterize* predefined subsets, and 2) *unsupervised* methods. A second dimension along which to characterize them has to do with the scope of their results: a) they apply either to (almost) the entire data set – they are *global* in nature, such as classification models or clusterings, or b) being *local*, they refer only to a (non-predefined) subset of the data.

The setting where the lack of supervision and local results intersect, i.e. 2b), it often referred to as "Pattern Mining" (PM), which is the term I will use hereafter. It holds a great promise: given large amounts of data and little or no supervision, PM can find interesting, hitherto undiscovered, relationships – *patterns* – in the data. Those patterns can in turn be exploited in the domains whence the data were generated to, for instance, further research, improve logistics, or increase sales. To see why this promise is so great, one only has to consider that supervised modeling already knows one side of relationship it seeks to establish. Supervised modeling seeks explanations, unsupervised modeling hypotheses.

Pattern Mining has been an active research field at least since the publication of the seminal paper introducing the Apriori algorithm [3] for frequent itemset mining (FIM). The twenty years since have brought an ever-widening scope of the field, extending the original itemset/association rule setting to semi-structured and structured data, the transactional setting to single-instance settings such as episode and network mining, and the application of PM techniques to a variety of different fields. This widening of scope also led to a plethora of techniques, published in a number of journals and conferences.

As I will argue in detail, however, the increase in number of topics and algorithms has not been paralleled by an equal increase in understanding the strengths and in particular limitations of developed techniques, or by guidelines for their employment. And creating tools (or rather in many cases blueprints) is not enough: to fulfill PM's promise and make the most of the developed techniques, it is necessary to give potential users an idea how to actually employ those tools.

In particular, there are three large gaps in our understanding of pattern mining:

1. **We do not know how most pattern mining algorithms actually perform quantitatively!** Pattern mining algorithms are *rarely*, if ever, evaluated on additional data after they have been published. Additionally, they are rarely extensively compared against each other.

2. **We do not know how to choose good parameter settings for pattern mining algorithms!** The relationships between parameter settings and running times/memory consumption are not well-established, let alone the relationship to the interestingness of patterns.

3. **We do not know how mined patterns relate to the generative processes underlying the data!** The current working interpretations of "interesting" recur to (objective or subjective) unexpectedness, or summarization/compression of the data. This undermines the interpretation of patterns and the applicability of derived knowledge to the real-life setting whence they originated.

In the following three sections, I will illustrate the gaps in our knowledge in detail. In particular, I will pay attention to the work that *did* attempt to fill those gaps, and proposed ways of researching these issues, but also why it falls short. These problems are not limited to PM – evaluations, comparisons, and the exploration of good parameter settings also often fall short of best practices for supervised settings, and techniques that result in global models. But the problem is particularly pronounced in PM, mainly because the local results require the evaluation of individual patterns and because quality criteria are harder to define.

As I will argue, an important contributing factor is the lack of data in general, and of data with controlled, diverse characteristics and known ground truth in particular:

- Lack of data necessarily limits experimental evaluations and if no new data is added to the portfolio, no reevaluations can be performed.
- Lack of diverse characteristics means that we have seen algorithmic behavior only over a relatively narrow range of settings, and that we lack understanding of how small changes in such characteristics affects behavior.
- The lack of data of which the ground truth is know, finally, is the main factor that makes it so hard to fill in the third gap: supervised data contains a certain amount of ground truth in the labels, while this is missing for unsupervised settings, and global approaches can be evaluated by assessing the data partition, for instance.

There is a potential solution to this problem – artificial data generation – but as I will show, the track record of the PM community w.r.t. data generation is rather weak so far. This has to be remedied if we want to do more than propose tools that might or might not work. While developing generators that lead to data with differing characteristics could turn out to be relatively easy, knowing what kind of generative processes can be expected to occur in real-life data will be more challenging and such knowledge will often not be found with data miners but with real-world practitioners. Hence, I argue that we need to add a deeper understanding of data – "data science" so-to-say – to the PM research portfolio, a task that will require the collaboration with researchers and practitioners of other fields that currently is too often more statement than fact.

2. (RE)EVALUATION/COMPARISON

The first glaring problem has to do with the lack of extensive evaluation of data mining algorithms, whether in the original papers, on additional data, or in comparisons with other techniques from the field. As a result thereof, we do not have a clear idea how different algorithms will behave on certain data. We *do* know some things: dense data sets will result in more patterns and more computational effort, strict formal concept analysis on real-life data will probably not result in good output compression. But apart from that, the body of knowledge is weak.

Often, the paper that introduces a new algorithm contains the most comprehensive published experimental evaluation of that algorithm in terms of running times, memory consumption etc. Even those initial evaluations are often not very extensive, though. In the following, I will give an overview of this phenomenon in different subfields.

The seminal FIM paper used an artificial data generator to evaluate their approach on more than 45 data sets, generated by varying parameters. Notably, all those data shared a common characteristic – they were sparse. A similarly systematic evaluation can still be found in [61], yet most other early work [22; 64; 44] used far fewer data sets. FIM papers since have followed this trend with few exceptions.

Sequence mining was first formalized and proposed at roughly the same time as FIM by the same authors and in [51] twelve data sets are used for evaluation, nine of which artificial, with a similarly declining trend for follow-up work [62; 21; 45]. Sequence mining was transferred from the transactional setting to finding recurrent patterns – *episodes* – in single large data sequences in [37] and algorithms from this subfield have typically only been evaluated on a few data sets.

With a few exceptions, papers on the generalizations of sequences – trees (introduced in [63]) and graphs [25; 30; 26; 60; 24] – have followed the same trajectory, with two of my own papers [11; 68] among the worst offenders in this regard. Graph mining has also been extended to the "single large" setting and different papers [17; 31; 28] have used less than twenty data sets each from various sources.

Rarely have those algorithms been *reevaluated* on additional data beyond that used in the original papers, apart from some comparisons in the papers that proposed improvements. Even in the latter case, transitivity has often been assumed – if algorithm B has been reported to perform better than algorithm A, comparing to B is considered enough. But obviously, this only holds for the data on which that evaluation has been performed, either locking future evaluations into the same restricted data or leading to unjustified generalizations about algorithmic behavior.

The paper that most incisively demonstrated this problem is arguably the one by Zheng *et al.* [65]. As described above, the data on which Apriori was evaluated were artificially generated and follow-up techniques mainly used subsets of those data. When comparing the artificial data to real-life data at their disposal, Zheng *et al.* noticed that the latter had different characteristics. An experimental comparison of the follow-up algorithms showed that claimed improvements in the literature did not transfer to the real-life data – the improvements had been an artifact of the data generation process.

Unfortunately, that paper has remained one of a handful of exceptions. With the exception of the Frequent Itemset Mining Implementations (FIMI) workshops [19; 5], little additional work has been done on FIM. While FIMI was undoubtedly important, there were notable short-comings: the workshop took place only twice, the focus was more on the practical aspects of implementing abstract algorithms than on an assessment of the algorithms themselves, and was limited to the, relatively small, collection of data sets available.

In graph mining, [59] compared four depth-first search graph miners on new data, notably reporting results that contradict those in [24]. The authors of [40] generated and manipulated data, and most notably find that there is no strong relation between run times and the efficiency of the graph codes, as had been claimed in the literature.

Episode mining is an area in which evaluations and comparisons of algorithms are particularly rare. I am not aware of any other work except my own [67], in which I used a data generator to generate data having a range of characteristics and reported results that indicate temporal constraints have more impact than pattern semantics, contrary to claims in the literature.

Those papers show how important it is to use data with new characteristics, and to pay attention to questions of implementations and hardware when assessing the usefulness of algorithmic approaches. Yet, compared to the body of work describing algorithmic solutions, the body of work comprehensively evaluating those solutions is rather small.

3. IDENTIFYING PARAMETER SETTINGS

A second problem, somewhat related to the first, is that, for the majority of techniques, it is unclear how parameter settings should be chosen. As a rule of thumb, more lenient parameter settings will lead to longer running times but not even this relationship has been established concretely. At first sight, the situation w.r.t. this problem is better but this impression is deceiving.

Several papers established a relationship between the distribution of mined frequent, closed, and maximal itemsets and algorithmic running times [64; 49; 20; 14]. Apart from the fact that such research does not exist for structured data and patterns, those approaches are faced with the obvious problem that extensive mining, at potentially extensive running times, is needed before the relationship can be established.

An improvement consists of estimating support distributions and running times based on partial mining results, or sampled patterns, as been the approach of [34; 43; 16; 10; 9]. Those studies only traverse half the distance though – in predicting running times and output sizes – even though [9] proposes setting a frequency threshold high enough to avoid the exponential explosion of the output set. Whether a threshold setting that allows the operation to finish in a reasonable amount of time will lead to *interesting* patterns, is largely unexplored.

A notable exception concerned with mining sequences under regular expression constraints can be found in [6]. By sampling patterns fulfilling data-independent constraints under assumptions about the symbol distribution, they derive a model of background noise, and identify thresholds expected to lead to interesting results. A similar idea can be found in [38], which uses sampling and regression to arrive at *pattern frequency spectra* for FIM. By comparing analytical expressions for spectra of random data to the actually derived spectra, they also identify deviating regions, proposing to explore those. Those two papers come closest to actually giving guidance for parameter selection but are limited to frequency thresholds. Yet, the *blue print* for building up knowledge about the interplay of data characteristics and parameter settings – for instance for storage in experiment databases [54] – is there, held back by the relatively limited supply of available data sets.

Apart from the fact that it would be attractive to fix parameter settings before an expensive mining operation, this is an instance in which the unsupervised nature of pattern mining makes the task harder than for supervised settings. In the latter, validation sets can be used to assess the quality of resulting models, and parameters adjusted accordingly. There is work based on related ideas, which derive *null models* directly from the data, already found patterns, or user knowledge and use statistical testing to remove all those patterns that are not unexpected w.r.t. the null hypothesis [7; 18; 39], or uses multiple comparisons correction and hold-out sets [57]. While such approaches promise to remove all statistically unsurprising patterns, unexpected patterns are not necessarily useful ones.

4. MATCHING PATTERNS TO REALITY

This last remark hints at an important issue in PM: which patterns to consider "interesting". The field has moved on from the FIM view of (relatively) frequent and incuding (relatively) strong implications. Yet what has taken its place are (objectively or subjectively) surprisingness (see above), or effective summarization/compression (e.g. [56]). Such patterns are undoubtedly interesting and useful but they leave us with the third and in my opinion most important gap in our knowledge: it is currently mostly unclear how the patterns that are being mined by PM techniques relate to the patterns actually occurring in the data.

To be clear about this: we know what types of patterns we have defined, and we have the algorithms to find them according to specific criteria (at least in the case of complete techniques). In FIM, for instance, if there is set of items that is often bought together, it will be found. Yet so will all of its subsets, and maybe intersections with other itemsets, and we are currently lacking the knowledge to decide which of these patterns are relevant. Hence, in many cases we do not know whether we capture meaningful relationships.

Furthermore, even if we could identify the actual patterns, we would not know how those relate to the processes that generated the data in the first place. To continue with the FIM example, papers will often give the motivation behind performing such mining by stating that supermarkets can group items that are bought together close to each other, to motivate those customers who did not buy them together to do so. An alternative proposal states supermarkets should group such items far apart to motivate customers to traverse the entire store space and potentially make additional purchases.

These strategies implicitly assume two different types of customer behavior, though: in the latter case, the co-purchases are systematic and can be leveraged to generate additional business. In the former, the co-purchases are somewhat opportunistic, which also means that using the first strategy could lead to loss of business. Maybe the two types of behavior actually lead to different expressions of the pattern in the data. And maybe the layout of the supermarket at the time of purchase has an effect on this expression, for instance because it enables the opportunistic behavior. But since there exist no studies relating mined itemsets to the shopping behavior itself, i.e. the generative process of the data, it is unclear, twenty years after FIM was proposed, which of the two assumptions holds.

This has grave implications. If it is unclear how patterns relate to the underlying processes, it is also unclear how to exploit patterns in the domains that the data originated from. Given that this is the rationale of data mining, the current state of the art in PM research is therefore failing the field's main purpose: supporting real-world decision making.

Again, there exist studies that have attempted to fill this gap in a more or less systematic manner [32; 53; 52; 36; 35;

66; 48; 58; 67]. The typical approach consists of embedding explicitly defined patterns in the data (with or without noise effects), and comparing mined patterns to them to understand the relationships. Of particular interest are experiments in which the data pattern generation takes a different form from the pattern definition, such as in [35] in which Bayes' Nets were used to generate itemsets. Yet, again, there are not many of them, and their focus has often been only on a single technique.

The problem of interpretation exists for supervised and/or global approaches as well, but arguably to a lesser degree. If the goal is prediction, high accuracy is an indicator of a good result – after all, there are "black box" classification techniques but no black box PM ones. Similarly, a clustering that exhibits high intra-cluster similarity and inter-cluster dissimilarity is probably a good one – incidentally an assessment that is related to good summarization. And even if the setting is supervised and local, as in subgroup discovery, patterns that correlate strongly with the subgroup can be expected to be meaningful.

5. THE DATA PROBLEM

The preceding sections should not be read as an indictment of all PM research. Quite contrary, many of the found solutions are ingenious and elegant, and the impressive tool kit that has been amassed should enable practitioners to address many real-world problems more effectively. But faced with data, a typical practitioner will not know which tools to choose, how to set the parameters without extensive trial-and-error, and what conclusion to draw from the resulting patterns – unless we fill in the gaps.

There are several factors that influence the described situation. Some of those are related to what kind of research is rewarded, which in turn relates to publication policies. Instead of discussing those, the interpretation of which is necessarily subjective, I want to draw attention to an objective factor that I have also pointed to in each section:

Despite what one would expect given the name "Data Mining", what we lack is data!

In reaction to the work of Zheng *et al.*, the data sets they introduced were added to the benchmark data sets for FIM and reliance on the data generator from [3] was reduced. Several other data sets were added over time and the collection is currently downloadable at the FIMI website.[1] Yet the totality of this collection comprises only twelve data sets. This is a far cry from the large amount of data sets available at the UCI repository for machine learning [8] (used for predictive learning, clustering, and subgroup discovery), the UCR collection of data for time series classification and clustering [29], or even the data sets available for multi-label learning.[2]

Sequence mining is mainly performed on a small number of biological data sets. Most of the real-life data used in episode mining papers are covered by non-disclosure agreements and have therefore never entered the public domain. There are handful of tree mining data sets, mainly based on click streams or website traversal. Graph mining also makes heavy use of a small number of molecular data sets, and network mining data sets have ranged from graph encodings of UCI data to snapshots of social, citation, traffic, or biological networks.

Unless the current collections cover by pure accident all characteristics that can be encountered in the real world, even best-practice evaluations based on them will not give a complete picture of algorithms' strengths and weaknesses, making it difficult to address the first and second problem. Furthermore, even *if* we had large amounts of real-life data at our disposal, in most cases we would *not* know the ground truth of such data and therefore could not address the problem laid out in Section 4. After all, real-life data in supervised settings "only" need a label to assess whether relationships are relevant but evaluating patterns in a local unsupervised setting needs a much deeper understanding of the data. And even if we had data available of which we knew the ground truth, we would lack the knowledge about the generative processes leading to this ground truth, as described above.

Luckily for computer scientists, there is an alternative to assembling ever increasing collections of real-life data, or rather a complement: artificial data generation. This is the solution that has been chosen in exploring phenomena in the SAT solving community [47], for instance, or for evaluating another unsupervised setting, clustering [46]. The idea is also running like a thread through much of the work I have reviewed so far, whether it is artificial data used for systematically exploring the effects of data characteristics, for identifying where data deviates from random backgrounds, or for matching patterns to generating processes.

5.1 Data Generation in PM

The problem is, however, that the story of data generation in PM so far is arguably one of failure. The data generator used in [3] was discredited by Zheng *et al.*. This has repercussions since the data generators used in sequence and graph (and arguably tree) mining papers base on similar considerations. Cooper *et al.* [13], attempting to fix a second problem of that generator, ignored the first one, and introduced new artifacts. The survey undertaken in [12] lists 22 generators for network data, all of which attempt to reproduce certain numerical properties of real-life data, such as degree distribution, or clustering coefficient. With the exception of a few that attempt to model particular types of networks, the authors found that none of the proposals gets it fully right. The reference is admittedly somewhat dated but other work published since [33; 41; 15; 42] comes to the same conclusion.

Generators leading to data resembling the one already available suffer from the fact that they do not solve the problem of the data bottle neck. Approaches such as [49; 50; 55] take the output of an FIM operation and generate databases that will result in similar output. Thus, they take the existence of data as a given, as do the approaches that create null models based on the data. Furthermore, the data generated by the former is expected to result in the same mix of relevant and irrelevant patterns as the old one, and the latter mask the underlying processes.

While artificial data generation enables us to create data with a wide range of characteristics, assess the effects of different kinds of noise on the ability to recover patterns, and to simulate different generative processes, we have not used this ability to fill in the gaps in our understanding. This will need to change, and I am convinced that to do so

[1] `http://fimi.ua.ac.be/data/` – accessed 08/21/2014

[2] `http://en.sourceforge.jp/projects/sfnet_mulan/releases/` – accessed 08/21/2014

we, or at least some of us, have to become *data scientists*.

6. DATA SCIENCE

When media and non-academics refer to data miners or data analysts, the term "data scientist" is often used. But what this term implies, in my opinion, is that such a person understands the data, and **we do not**. Part of this is by design – as I wrote in the beginning, the promise of pattern mining is to find interesting patterns in a largely unsupervised manner. The naive interpretation of this promise, however, is similarly flawed as the claim that in the age of "Big Data", discovering correlation replaces understanding causation [4].[3]

To fill in the gaps in our understanding as PM researchers, data science needs to be added to our expertise. We need to develop data generators that produce varied characteristics in a controlled manner, to enable extensive experiments. Those data generators need to use generative processes to which we can map patterns back, so we start to understand how certain processes manifest in the output of the tools we develop. Concretely, this means exploring different distributions (and mixtures thereof) governing the data generation, instead of fixing a single one. It means adding varying degrees of noise to the data. And it means using generative processes that are different from the sought patterns – Bayes' Nets for itemset data, or interacting agents for network data, for instance. In fact, there exists already a tool that makes some of this possible, the KNIME data generator [2], which is however neither used widely nor systematically so far.

Once we have access to such generators, we can follow the approach of existing studies to fill in some of the gaps that currently exist. This means, for instance, evaluating and comparing algorithms over data of different density, pattern length, alphabet size, etc. It means establishing what proportion of individual patterns can be recovered under the effects of noise. It means assessing whether highly ranked itemsets represent fragments of embedded patterns, or represent subgraphs of Bayes' Nets. It also means understanding whether data that are generated by the same processes with the same parameters actually have the same characteristics, and whether they give rise to similar result sets, i.e. whether it is appropriate to transfer insights derived on one data set to another that "looks" similar. In other words, it should allow us to develop better ways of comparing data sets.

This is obviously not an exhaustive enumeration and it will take the creativity and effort of the community to get the field to that stage. But even that can only be a step on the way: we can learn how the patterns we mine relate to the patterns in the data, and in turn to the processes that generated them. How itemsets relate to agents that "shop" according to certain "behavior", for instance.

The knowledge about real-life behavior cannot come from inside our community, however. Instead, it will be found in physics [1], in engineering [27], in the social and life sciences. Once *we* have understood which method to use, how to set parameters, and how to select relevant patterns and interpret them, based on the data available, we can approach practitioners from those fields. Using their knowledge, we can generate data that are real life-like, and troubleshoot generators and evaluation methods. In all probability, some of the assumptions from those fields will turn out to be wrong but probably not more wrong than the assumptions we ourselves have made in generating data so far. And if, building on such assumptions, we find them to be wrong (or at least questionable), and feed this information back into the fields whence they originated, even better.

I have not come here to bury PM but to praise it. I am convinced that the potential of the tools that the community has developed over the last two decades is tremendous. I am, however, challenging the community to develop guidance for how to use those tools. Working closely with practitioners and giving them hands-on guidance, the modus operandi of many application papers, is a worthy endeavour but it is also time-consuming and allows for little generalization. We have to solve the data problem in data mining and we have to do it in a better-founded way than by trying to acquire additional real-life data sets. We need to make a science out of generating digital data.

Acknowledgments

I am grateful to Matthijs van Leeuwen, Arno Siebes, and Jilles Vreeken for reviewing preliminary versions of this article, giving feedback, sharpening the questions, and discussing possible answers. The author was supported by the FP7-PEOPLE-2013-IAPP project GRAISearch (Grant Agreement Number 612334) at the time of writing.

7. REFERENCES

[1] Corsika - an air shower simulation program,. https://web.ikp.kit.edu/corsika/.

[2] I. Adä and M. R. Berthold. The new iris data: modular data generators. In B. Rao, B. Krishnapuram, A. Tomkins, and Q. Yang, editors, *KDD*, pages 413–422. ACM, 2010.

[3] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Databases*, pages 487–499, Santiago de Chile, Chile, Sept. 1994. Morgan Kaufmann.

[4] C. Anderson. The end of theory: The data deluge makes the scientific method obsolete. `http://archive.wired.com/science/discoveries/magazine/16-07/pb_theory`. Accessed 08/21/2014.

[5] R. J. Bayardo Jr., B. Goethals, and M. J. Zaki, editors. *FIMI '04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Brighton, UK, November 1, 2004*, 2004.

[6] J. Besson, C. Rigotti, I. Mitasiunaite, and J.-F. Boulicaut. Parameter tuning for differential mining of string patterns. In *ICDM Workshops*, pages 77–86. IEEE Computer Society, 2008.

[7] T. D. Bie. Maximum entropy models and subjective interestingness: an application to tiles in binary databases. *Data Min. Knowl. Discov.*, 23(3):407–446, 2011.

[3] A claim that incidentally experiences tremendous push-back.

[8] C. Blake and C. Merz. UCI repository of machine learning databases, 1998.

[9] M. Boley, T. Gärtner, and H. Grosskreutz. Formal concept sampling for counting and threshold-free local pattern mining. In *SDM*, pages 177–188. SIAM, 2010.

[10] M. Boley and H. Grosskreutz. A randomized approach for approximating the number of frequent sets. In *ICDM*, pages 43–52. IEEE Computer Society, 2008.

[11] B. Bringmann and A. Zimmermann. Tree^2 - Decision trees for tree structured data. In A. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama, editors, *9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 46–58. Springer, 2005.

[12] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1), 2006.

[13] C. Cooper and M. Zito. Realistic synthetic data for testing association rule mining algorithms for market basket databases. In J. N. Kok, J. Koronacki, R. L. de Mántaras, S. Matwin, D. Mladenic, and A. Skowron, editors, *PKDD*, volume 4702 of *Lecture Notes in Computer Science*, pages 398–405. Springer, 2007.

[14] F. Flouvat, F. D. Marchi, and J.-M. Petit. A new classification of datasets for frequent itemsets. *J. Intell. Inf. Syst.*, 34(1):1–19, 2010.

[15] A. Freno, M. Keller, and M. Tommasi. Fiedler random fields: A large-scale spectral approach to statistical network modeling. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *NIPS*, pages 1871–1879, 2012.

[16] F. Geerts, B. Goethals, and J. V. den Bussche. Tight upper bounds on the number of candidate patterns. *ACM Trans. Database Syst.*, 30(2):333–363, 2005.

[17] S. Ghazizadeh and S. S. Chawathe. Seus: Structure extraction using summaries. In S. Lange, K. Satoh, and C. H. Smith, editors, *Discovery Science*, volume 2534 of *Lecture Notes in Computer Science*, pages 71–85. Springer, 2002.

[18] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas. Assessing data mining results via swap randomization. *TKDD*, 1(3), 2007.

[19] B. Goethals and M. J. Zaki, editors. *FIMI '03, Frequent Itemset Mining Implementations, Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 19 December 2003, Melbourne, Florida, USA*, volume 90 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

[20] K. Gouda and M. J. Zaki. Genmax: An efficient algorithm for mining maximal frequent itemsets. *Data Min. Knowl. Discov.*, 11(3):223–242, 2005.

[21] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M. Hsu. Freespan: frequent pattern-projected sequential pattern mining. In R. Ramakrishnan, S. J. Stolfo, R. J. Bayardo, and I. Parsa, editors, *KDD*, pages 355–359. ACM, 2000.

[22] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In W. Chen, J. F. Naughton, and P. A. Bernstein, editors, *SIGMOD Conference*, pages 1–12. ACM, 2000.

[23] J. Han, B. W. Wah, V. Raghavan, X. Wu, and R. Rastogi, editors. *Fifth IEEE International Conference on Data Mining*, Houston, Texas, USA, Nov. 2005. IEEE.

[24] J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraphs in the presence of isomorphism. In *ICDM*, pages 549–552. IEEE Computer Society, 2003.

[25] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In D. A. Zighed, H. J. Komorowski, and J. M. Zytkow, editors, *PKDD*, volume 1910 of *Lecture Notes in Computer Science*, pages 13–23. Springer, 2000.

[26] A. Inokuchi, T. Washio, K. Nishimura, and H. Motoda. A fast algorithm for mining frequent connected subgraphs. Technical report, IBM Research, 2002.

[27] E. F. V. J. J. Down. A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3):245–255, March 1993.

[28] U. Kang, C. E. Tsourakakis, and C. Faloutsos. Pegasus: A peta-scale graph mining system. In W. Wang, H. Kargupta, S. Ranka, P. S. Yu, and X. Wu, editors, *ICDM*, pages 229–238. IEEE Computer Society, 2009.

[29] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana. The UCR time series classification/clustering homepage, 2011.

[30] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In N. Cercone, T. Y. Lin, and X. Wu, editors, *ICDM*, pages 313–320. IEEE Computer Society, 2001.

[31] M. Kuramochi and G. Karypis. Finding frequent patterns in a large sparse graph*. *Data Min. Knowl. Discov.*, 11(3):243–271, 2005.

[32] S. Laxman, P. S. Sastry, and K. P. Unnikrishnan. Discovering frequent episodes and learning hidden markov models: A formal connection. *IEEE Trans. Knowl. Data Eng.*, 17(11):1505–1517, 2005.

[33] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In J. Huai, R. Chen, H.-W. Hon, Y. Liu, W.-Y. Ma, A. Tomkins, and X. Zhang, editors, *WWW*, pages 695–704. ACM, 2008.

[34] L. Lhote, F. Rioult, and A. Soulet. Average number of frequent (closed) patterns in bernouilli and markovian databases. In Han et al. [23], pages 713–716.

[35] M. Mampaey and J. Vreeken. Summarizing categorical data by clustering attributes. *Data Min. Knowl. Discov.*, 26(1):130–173, 2013.

[36] M. Mampaey, J. Vreeken, and N. Tatti. Summarizing data succinctly with the most informative itemsets. *TKDD*, 6(4):16, 2012.

[37] H. Mannila and H. Toivonen. Discovering frequent episodes in sequences. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD'95)*, pages 210–215. AAAI Press, 1995.

[38] A. U. Matthijs van Leeuwen. Fast estimation of the pattern frequency spectrum.

[39] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[40] S. Nijssen and J. Kok. Frequent subgraph miners: runtimes don't say everything. In T. Gärtner, G. Garriga, and T. Meinl, editors, *Proceedings of the Workshop on Mining and Learning with Graphs,*, pages 173–180, 2006.

[41] G. K. Orman, V. Labatut, and H. Cherifi. Qualitative comparison of community detection algorithms. In H. Cherifi, J. M. Zain, and E. El-Qawasmeh, editors, *DICTAP (2)*, volume 167 of *Communications in Computer and Information Science*, pages 265–279. Springer, 2011.

[42] G. K. Orman, V. Labatut, and H. Cherifi. Towards realistic artificial benchmark for community detection algorithms evaluation. *IJWBC*, 9(3):349–370, 2013.

[43] P. Palmerini, S. Orlando, and R. Perego. Statistical properties of transactional databases. In H. Haddad, A. Omicini, R. L. Wainwright, and L. M. Liebrock, editors, *SAC*, pages 515–519. ACM, 2004.

[44] J. Pei, J. Han, and R. Mao. Closet: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[45] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Prefixspan: Mining sequential patterns by prefix-projected growth. In D. Georgakopoulos and A. Buchmann, editors, *ICDE*, pages 215–224. IEEE Computer Society, 2001.

[46] Y. Pei and O. Zaïane. A synthetic data generator for clustering and outlier analysis. Technical report, 2006.

[47] D. M. Pennock and Q. F. Stout. Exploiting a theory of phase transitions in three-satisfiability problems. In *AAAI/IAAI, Vol. 1*, pages 253–258, 1996.

[48] B. A. Prakash, J. Vreeken, and C. Faloutsos. Efficiently spotting the starting points of an epidemic in a large graph. *Knowl. Inf. Syst.*, 38(1):35–59, 2014.

[49] G. Ramesh, W. Maniatty, and M. J. Zaki. Feasible itemset distributions in data mining: theory and application. In *PODS*, pages 284–295. ACM, 2003.

[50] G. Ramesh, M. J. Zaki, and W. Maniatty. Distribution-based synthetic database generation techniques for itemset mining. In *IDEAS*, pages 307–316. IEEE Computer Society, 2005.

[51] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In P. M. G. Apers, M. Bouzeghoub, and G. Gardarin, editors, *EDBT*, volume 1057 of *Lecture Notes in Computer Science*, pages 3–17. Springer, 1996.

[52] N. Tatti and J. Vreeken. Discovering descriptive tile trees - by mining optimal geometric subtiles. In P. A. Flach, T. D. Bie, and N. Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part I*, volume 7523 of *Lecture Notes in Computer Science*, pages 9–24. Springer, 2012.

[53] N. Tatti and J. Vreeken. The long and the short of it: summarising event sequences with serial episodes. In Q. Yang, D. Agarwal, and J. Pei, editors, *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 462–470. ACM, 2012.

[54] J. Vanschoren, H. Blockeel, B. Pfahringer, and G. Holmes. Experiment databases - A new way to share, organize and learn from experiments. *Machine Learning*, 87(2):127–158, 2012.

[55] J. Vreeken, M. van Leeuwen, and A. Siebes. Preserving privacy through data generation. In N. Ramakrishnan and O. Zaïane, editors, *ICDM*, pages 685–690. IEEE Computer Society, 2007.

[56] J. Vreeken, M. van Leeuwen, and A. Siebes. Krimp: mining itemsets that compress. *Data Min. Knowl. Discov.*, 23(1):169–214, 2011.

[57] G. I. Webb. Discovering significant patterns. *Machine Learning*, 68(1):1–33, 2007.

[58] G. I. Webb and J. Vreeken. Efficient discovery of the most interesting associations. *Transactions on Knowledge Discovery from Data*, 8(3):15–1, 2014.

[59] M. Wörlein, T. Meinl, I. Fischer, and M. Philippsen. A quantitative comparison of the subgraph miners mofa, gspan, ffsm, and gaston. In A. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama, editors, *PKDD*, pages 392–403. Springer, 2005.

[60] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *ICDM*, pages 721–724. IEEE Computer Society, 2002.

[61] M. J. Zaki. Scalable algorithms for association mining. *IEEE Trans. Knowl. Data Eng.*, 12(3):372–390, 2000.

[62] M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.

[63] M. J. Zaki. Efficiently mining frequent trees in a forest. In *KDD*, pages 71–80. ACM, 2002.

[64] M. J. Zaki and C.-J. Hsiao. Charm: An efficient algorithm for closed itemset mining. In R. L. Grossman, J. Han, V. Kumar, H. Mannila, and R. Motwani, editors, *SDM*. SIAM, 2002.

[65] Z. Zheng, R. Kohavi, and L. Mason. Real world performance of association rule algorithms. In *KDD*, pages 401–406, 2001.

[66] A. Zimmermann. Objectively evaluating condensed representations and interestingness measures for frequent itemset mining. *Journal of Intelligent Information Systems*, pages 1–19, 2013.

[67] A. Zimmermann. Understanding episode mining techniques: Benchmarking on diverse, realistic, artificial data. *Intell. Data Anal.*, 18(5):761–791, 2014.

[68] A. Zimmermann and B. Bringmann. Ctc - correlating tree patterns for classification. In Han et al. [23], pages 833–836.

New Research Directions in Knowledge Discovery and Allied Spheres

Anisoara Nica
Research and Development
Sybase, An SAP Company
Waterloo, ON, Canada
anisoara.nica@sybase.com

Fabian M. Suchanek
Otto Hahn Research Group "Ontologies"
Max Planck Institute for Informatics
Saarbrücken, Germany
fabian@suchanek.name

Aparna S. Varde
Department of Computer Science
Montclair State University
Montclair, NJ, USA
vardea@montclair.edu

ABSTRACT

The realm of knowledge discovery extends across several allied spheres today. It encompasses database management areas such as data warehousing and schema versioning; information retrieval areas such as Web semantics and topic detection; and core data mining areas, e.g., knowledge based systems, uncertainty management, and time-series mining. This becomes particularly evident in the topics that Ph.D. students choose for their dissertation. As the grass roots of research, Ph.D. dissertations point out new avenues of research, and provide fresh viewpoints on combinations of known fields. In this article we overview some recently proposed developments in the domain of knowledge discovery and its related spheres. Our article is based on the topics presented at the doctoral workshop of the ACM Conference on Information and Knowledge Management, CIKM 2011.

Keywords

Ranking, Text Mining, Extreme Web, ETL, Pattern Recognition, Resource Monitoring, Version Control, KNN, Semantic Web, Main Memory Database, Data Warehousing, Database Analytics

1. INTRODUCTION

Knowledge discovery is an interdisciplinary field of research, which encompasses diverse areas such as data mining, database management, information retrieval, and information extraction. This inspires doctoral candidates to pursue research in and across these related disciplines, with the core contributions of their dissertation being in one or more of these areas. In this article, we review some of the directions that the researchers of tomorrow pursue. We provide a report of the research challenges addressed by students in the Ph.D. workshop PIKM 2011. This workshop was held at the ACM Conference on Information and Knowledge Management, CIKM 2011. The CIKM conference and the attached workshop encompass the tracks of data mining, databases and information retrieval, thus providing an excellent venue for dissertation proposals and early doctoral work in and across different spheres of knowledge discovery. This workshop was the fourth of its kind after three successful PIKM workshops in 2007 [15, 16], 2008 [11, 14] and 2010 [9, 10]. The PIKM 2011 [8] attracted submissions from several countries around the globe. After a review by a PC comprising 19 experts from academia and industry worldwide, 9 full papers were selected for oral presentation and 4 short papers for poster presentation. The program was divided into 4 sessions: data mining and knowledge management; databases; information retrieval; and a poster session with short papers in all tracks.

The first highlight of the PIKM 2011 was a keynote talk on "Extreme Web Data Integration" by Prof. Dr. Felix Naumann from the Hasso Plattner Institute in Potsdam, Germany [8]. This talk addressed the integration and querying of data from the Semantic Web at large scale, i.e., from vast sources such as DBpedia, Freebase, public domain government data, scientific data, and media data such as books and albums. It discussed the challenges related to the heterogeneity of Web data (even inside the Semantic Web), common ontology development, and multiple record linkage. It also highlighted the problems of Web data integration in general, such as identification of good quality sources, structured data creation, standardization-related cleaning, entity matching, and data fusion.

We now furnish a review comprising a summary and critique of the dissertation proposals presented at this workshop, discussing new directions of research in data mining and related areas. We follow the thematic structure of the workshop with 3 topic areas: knowledge discovery, database research, and information retrieval. The knowledge discovery issues surveyed in this article include areas as diverse as pattern recognition in time-series, resource monitoring with knowledge-based models, version control under uncertainty, and random walk k-nearest-neighbors (k-NN) for classification. The database research problems presented here entail aggregation for in-memory databases, evolving extract-transform-load (E-ETL) frameworks, schema and data versioning, and automatic regulatory compliance support. The information retrieval themes involve paradigms such as user interaction with polyrepresentation, ranking with entity relationship (ER) graphs, online conversation mining, sub-topical document structure, and cost optimization in test collections.

The workshop also issues a best paper award to the most exciting dissertation proposal, as determined by the PC of the workshop. This year's award went to the proposal "Ranking Objects by Following Paths in Entity-Relationship Graphs" [6], in the Information Retrieval track.

The rest of this article is organized as follows. Sections 2, 3, and 4 discuss the different tracks of PIKM, i.e., knowledge discovery, database research, and information retrieval, respectively. In Section 5, we summarize the hot topics of current research, and compare them with the topics of the previous PIKM workshops.

2. KNOWLEDGE DISCOVERY

The topics surveyed here are those with main contributions in data mining and knowledge discovery, although some of them overlap with the other two thematic tracks, namely, databases and information retrieval.

2.1 Pattern Recognition in Evolving Data

Many devices today, such as mobile phones, modern vehicular equipment and smart home monitors, contain integrated sensors.

These sensors need to capture information with reference to context (e.g., abnormal motor behavior in vehicles) in order to enable the device to adapt to change and cater to users. This entails dealing with temporal data that is evolving in the respective environment. Spiegel et al. [13] addressed this problem. They proposed efficient methods to recognize contextual patterns in continuously evolving temporal data. Their approach incorporated a machine learning paradigm with a three step process: extracting features to construct robust models capturing important data characteristics; determining homogenous intervals and point of change by segmentation; and grouping the time series segments into their respective subpopulation by clustering and classification. This problem was considered interesting by the audience especially due to its application in smart phones where the proposed approach is useful in detecting patterns such as changing product prices to provide better responses to queries.

2.2 Resource Monitoring with KM Models

Abele et al. [1] focused on a knowledge engineering problem in the manufacturing domain. More specifically, their problem was on computerizing the monitoring of resource consumption in complex industrial production plants, a task that usually involves tremendous time and manual effort. They proposed a semi-automated method for monitoring through knowledge based modeling with sensors, reasoners, annotations and rule engines, easily adaptable to changes. Their modeling approaches included object-oriented models with UML and domain models in the Semantic Web, with specific use of a data format for plant engineering called AutomationML. Advantages of their monitoring approach were reduction in manual effort, saving of time and resources, application independence and flexibility. Their research was particularly appreciated due to the manner in which they dealt with a domain-specific problem in an application independent manner by proposing knowledge based models that adequately encompassed ontology and logic through formalisms.

2.3 Version Control under Uncertainty

Collaborative work on documents poses the problem of version control in the presence of uncertainty. Ba et al. [2] tackled this issue with much attention to XML documents. They proposed a version-control paradigm based on XML data integration to evaluate data uncertainty and automatically perform conflict resolution. They outlined the main features of versioning systems and defined a version-space formalism to aid in data collaboration and in the management of uncertainty. In their proposed solution, they incorporated aspects such as XML differencing with respect to trees, delta models for operations like insertions, moves and updates, directed acyclic graphs of version derivations and construction of probabilistic XML documents that minimize uncertainty. This work was found highly appealing due to its contributions to data mining, databases and IR since it addressed the important problem of uncertainty in knowledge discovery, proposed models based on database theoretical concepts and applied these within the context of XML in information retrieval.

2.4 Random Walk k-NN for Classification

The challenging problem of multi-label classification formed the focus of the work by Xia et al. [19]. In this problem, an instance belonged to more than one class as predicted by the classifier and the number of potential class labels could be exponential, posing a challenge in predicting the target class. The authors proposed an approach to solve this problem by exploiting the benefits of k nearest neighbors and random walks. They first constructed a link graph based on k-NN, and then executed a random walk on the graph, so as to obtain a probability distribution of class label sets for the target instance. Further, they determined a classification threshold based on minimizing the Hamming Loss that computed the average binary classification error. Although this work did not show any significant experimentation, and thus drew criticism from the audience, it provided a complexity analysis with true and predicted class labels that was found acceptable. The authors also provided good motivation for their work by addressing real-world applications such as functional genomics, semantic scene classification and text categorization.

3. DATABASE RESEARCH PROBLEMS

The work presented at PIKM 2011 in the area of database research includes topics in database analytics, main memory databases, evolution of resources using ETL, and schema and data versioning.

3.1 Aggregation Strategies for Columnar In-memory Database Systems

Some of the hottest topics in current database research include in-memory database systems, columnar database systems, self-managing and self-adapting database systems in the presence of mixed workloads. These topics, and, in general, issues related to large in-memory database systems were discussed by Müller and Plattner in [7]. The main goal of the proposed work is to design an adaptive engine for aggregation materialization for database analytics. The paper addresses problems related to aggregation materialization identifying relevant cost factors for deciding when the materialization is cost efficient, and also what characteristics a cost model used by this type of adaptive engines must have. The authors provided an excellent motivation for this research by discussing recent trends in database usage where online transactional processing and online analytical processing are all reunified in one single database.

3.2 Managing Evolving ETL Processing

Traditional data warehouse systems employ an ETL process that integrates the external data sources into the data warehouse. One of the hardest problems in ETL research is the frequent changes in the schema of the external data sources. Wojciechowski [18] took up the challenge of automating the process of adapting and evolving the ETL process itself to the changes in the structural schemas of the external data sources. The proposed E-ETL framework brings together, in a unique way, techniques from materialized view adaptation, schema and data evolution, versioning of schema and data [17], and multiversion in data warehouse systems. The author presented the current implementation of the E-ETL system and future work on developing an appropriate language for defining ETL operations, as well as extensions to the current set of changes at the external data sources that E-ETL can detect and adapt to.

3.1 Schema and Data Versioning Systems

The topic of schema and data versioning systems, which was briefly referenced in [18] in the context of evolvable ETL processing, was the main topic of the paper by Wall and Angryk [17]. The authors investigate two different approaches to schema and data versioning: one approach is based on storing the minimal set of changes from the base schema and data to each branch, a sandbox, representing a particular deviation from the base schema The queries run against the branch are mapped accordingly to the base schema and its data, as well as to the branch schema and data to correctly reflect the set of changes in the branch. Another approach is to create copies of the modified tables into the branch, and propagate changes done in the base schema and data to these copies. The most interesting parts of this work are related to investigation into the qualitative and quantitative differences between the two techniques. The authors described the design of the ScaDaVer system meant to be used in assessing and testing different approaches to schema and data versioning. The topic and the motivation for this research are well established in the database area with new importance being brought to by the SaaS systems where multiple instances of the database can be operational in the same time.

3.2 Automatic Regulatory Compliance

A unique research on using semantic web technologies to support the management of the compliance systems was presented by Sapkota et al. in [12]. Compliance Management systems, although using computer-assisted processes, still lack one of the fundamental requirements, which is the automation towards updating the system when the regulations change. The paper proposed a Semantic Web methodology to automate these processes. The proposed techniques include automatic extraction and modeling of regulatory information, and mapping regulations to organizational internal processes. The authors describe the implementation of the proposed methodology which has been started using the Pharmaceutical industry as a case study, and is being applied to the Eudralex European regulation for good manufacturing practice in the pharmaceutical industry.

4. INFORMATION RETRIEVAL THEMES

The third thematic track of the PIKM workshop was concerned with topics in information retrieval. It comprised 2 poster papers and 3 full papers, including the best paper award winner (see Section 4.5).

4.1 User Interaction with Polyrepresentation

Zellhoefer et al. [20] introduce a new interactive information retrieval model. They present a prototypical GUI-based system that allows users to interactively define and narrow down the documents they are interested in. The novelty of the proposed approach lies in the inspiration from the cognitively motivated principle of polyrepresentation. The principle's core hypothesis is that a document is defined by different representations such as low-level features, textual content, and the user's context. Eventually, these representations can be used to form a cognitive overlap of features in which highly relevant documents are likely to be contained. In their work, the authors link this principle to a quantum logic-based retrieval model, which enhances its appeal. The work also addresses the issue of information need drifts, i.e., of adjustments of the information needs of the user. This gives the work a refreshingly practical angle.

4.2 Online Conversation Mining

Inches et al. [5] address the problem of data mining in online instant messaging services, twitter messages and blogs. The texts in these new areas of the social Web exhibit unique properties. In particular, the documents are short, user generated, and noisy. The authors investigate two different but related aspects of the content of these colloquial messages: the topic identification and the author identification tasks. They develop a framework in which social-Web specific features, such as emoticons, abbreviations, shoutings, and burstiness are systematically extracted from the documents. These features are used to build up a model of topic representation and author representation. The paper at the workshop described work in progress, with the author identification, and the synthesis of the features still to be explored.

4.3 Sub-topical document structure

In text segmentation, a document is decomposed into constituent subtopics. Ganguly et al. [3] analyze how text segmentation can help for information retrieval. This can happen, for example, by computing the score of a document as a combination of the retrieval scores of its constituent segments, or by exploiting the proximity of query terms in documents for ad-hoc search. Text segmentation can also help for question answering (QA), where retrieved passages from multiple documents are aggregated and presented as a single document to a searcher. Text segmentation can also help segmenting the query, if the query is a long piece of text. This is particularly important for patent prior art search tasks, where the query is an entire patent.

4.4 Cost Optimization in Test Collections

Information retrieval systems are usually evaluated by measuring the relevance of the retrieved documents on a test collection. This relevance has to be assessed manually. Since this is usually a costly process, Hosseini et al [4] consider the problem of optimally allocating human resources to construct the relevance judgments. In their setting, there is a large set of test queries, for each of which a large number of documents need to be judged, even though the available budget only permits to judge a subset of them. In their work, the authors propose a framework that treats the problem as an optimization problem. The authors design optimization functions and side constraints that ensure not only that the resources are allocated efficiently, but also that new, yet unseen systems can be evaluated with the previous relevance judgments. It also takes into account uncertainty that is due to human errors. This way, the proposal aims to tackle holistically a problem that is of principal importance in the area of information retrieval in general.

4.5 Ranking with ER graphs

The area of Information Retrieval is no longer restricted to text documents. It can equally well be applied to entity-relationship graphs or RDF knowledge bases. Kahng et al. [6] explore this idea by looking at paths in entity-relationship graphs. They put forward two ideas: First, an entity-relationship graph can represent not just factual information, but also other, additional, heterogeneous information. This allows treating tasks that have traditionally been seen as orthogonal within one single model. Second, the paper proposes to take into account the schema of the graph, and in particular the labels along the edges of paths. The paper shows that this allows treating tasks that have traditionally been seen as different, such as information retrieval and item recommendation, in the same model. This contribution earned the work the best paper award of the PIKM 2011.

5. CONCLUSIONS

In this article, we have looked at the area of knowledge discovery from the viewpoint of Ph.D. students. We have presented some promising research proposals, which treat not just the area of knowledge discovery but also the neighboring fields of database management and information retrieval.

PIKM 2011 was the fourth Ph.D. workshop in a series of such workshops. Looking back, we see that PIKM 2007 concentrated on topics such as fuzzy clustering, linguistic categorization, online classification, rule-based processing and collaborative knowledge management frameworks. In 2008, the focus was on social networking, text mining and speech information retrieval. In 2010, the research areas tilted towards security, quality and ranking, getting more interdisciplinary. In PIKM 2011, three new topics emerged: mining in social media [5], mining in the Semantic Web [6], and main memory databases [7]. We see this as a proof of the growing attraction of these domains. In addition, one theme that caught particular attention across multiple areas was the evolution of resources over time, be it in the area of ETL processing [18], in the area of XML [2], or in database versioning [17]. We also see an overarching topic of process management in general, in the sense of resource monitoring [1] and regulatory compliance [12], as well as in the sense of human cost optimization when producing IR test collections [4].

After four successful Ph.D. workshops in Information and Knowledge Management, the PIKMs in 2007, 2008, 2010 and 2011, we hope to continue these events in future conferences. We believe that such workshops benefit not just Ph.D. students, but also the research community as a whole, since Ph.D. thesis proposals point out new research avenues and provide fresh viewpoints from the researchers of tomorrow.

6. REFERENCES

[1] L. Abele, M. Kleinsteuber, and H. Thorbjoern. Resource monitoring in industrial production with knowledge-based models and rules. *PIKM 2011*, pp. 35 – 43.

[2] M. L. Ba, T. Abdessalem, and P. Senellart. Towards a version control model with uncertain data. *PIKM 2011*, pp. 43 – 50.

[3] D. Ganguly, J. Leveling, and G. J. Jones. Utilizing sub-topical structure of documents for information retrieval. *PIKM 2011*, pp. 75 – 78.

[4] M. Hosseini, I. Cox, and N. Milic-Frayling. Optimizing the cost of information retrieval test collections. *PIKM 2011*, pp. 79 – 82.

[5] G. Inches and F. Crestani. Online conversation mining for author characterization and topic identification. *PIKM 2011*, pp. 19 – 26.

[6] M. Kahng, S. Lee, and S.-G. Lee. Ranking objects by following paths in entity-relationship graphs. *PIKM 2011*, pp. 11 – 18.

[7] S. Müller and H. Plattner. Aggregation strategies for columnar in-memory databases in a mixed workload. *PIKM 2011*, pp. 51-57.

[8] A. Nica and F. M. Suchanek, editors. *Proceedings of the 4th Ph.D. Workshop in CIKM, PIKM 2011, 20th ACM Conference on Information and Knowledge Management, ACM CIKM 2011*, Glasgow, UK.

[9] A. Nica, F. M. Suchanek, and A. S. Varde. Emerging multidisciplinary research across database management systems. *SIGMOD Record*, 39(3):33–36, 2010.

[10] A. Nica and A. S. Varde, editors. *Proceedings of the Third Ph.D. Workshop in CIKM, PIKM 2010, 19th ACM Conference on Information and Knowledge Management, ACM CIKM 2010*, Toronto, Canada.

[11] P. Roy and A. S. Varde, editors. *Proceedings of the Second Ph.D. Workshop in CIKM, PIKM 2008, 18th ACM Conference on Information and Knowledge Management, ACM CIKM 2008*, Napa Valley, CA.

[12] K. Sapkota, A. Aldea, D. A. Duce, M. Younas, and R. Bãnares-Alćantara. Towards semantic methodologies for automatic regulatory compliance support. *PIKM 2011*, pp. 83 – 86.

[13] S. Spiegel, B.-J. Jain, E. W. D. Luca, and S. Albayrak. Pattern recognition in multivariate time series. *PIKM 2011*, pp. 27-33.

[14] A. S. Varde. Challenging research issues in data mining, databases and information retrieval. *SIGKDD Explorations*, 11(1):49–52, 2009.

[15] A. S. Varde and J. Pei. *Proceedings of the First Ph.D. Workshop in CIKM, PIKM 2007, Sixteenth ACM Conference on Information and Knowledge Management, ACM CIKM 2007*, Lisbon, Portugal.

[16] A. S. Varde and J. Pei. Advances in information and knowledge management. *SIGIR Forum*, 42(1):29–35, 2008.

[17] B. Wall and R. Angryk. Minimal data sets vs. synchronized data copies in a schema and data versioning system. *PIKM 2011*, pp. 67 – 73.

[18] A. Wojciechowski. E-ETL: Framework for managing evolving ETL processes. *PIKM 2011*, pp. 59-65.

[19] X. Xia, X. Yang, S. Li, C. Wu, and L. Zhou. RW.KNN: A proposed random walk KNN algorithm for multi-label classification. *PIKM 2011*, pp. 87 – 90.

[20] D. Zellhoefer and I. Schmitt. A user interaction model based on the principle of polyrepresentation. *PIKM 2011*, pp. 3 – 10.

About the authors:

Anisoara Nica holds a Ph.D. in Computer Science and Engineering from the University of Michigan, Ann Arbor, USA, 1999, with the dissertation in the areas of information integration systems and data warehousing. She is currently Distinguished Engineer in the SQL Anywhere Research and Development team of Sybase, An SAP Company, in Waterloo, Canada. Her research interests and expertise are focused on database management systems, in particular query processing and query optimization, data warehousing, distributed, mobile, and parallel databases. Her work experience includes Lawrence Berkeley National Laboratory and International Computer Science Institute in Berkeley. Dr. Nica holds eight patents and has several other patents pending. She has published more than 25 research articles, and she reviews for NSERC, ACM SIGMOD, IEEE ICDE, ACM CIKM.

Fabian Suchanek is the leader of the Otto Hahn Research Group "Ontologies" at the Max Planck Institute for Informatics in Germany. In his Ph.D. thesis, Fabian developed inter alia the YAGO-Ontology, earning him a honorable mention of the SIGMOD dissertation award. His interests include information extraction, automated reasoning, and ontologies in general. Fabian has published around 30 scientific articles, and he reviews for conferences and journals such as ACL, EDBT, WSDM, WWW, TKDE, TODS, AIJ, JWS, and AAAI.

Aparna Varde, Computer Science Faculty at Montclair State University (New Jersey), has a Ph.D. from WPI (Massachusetts) with a dissertation in the data mining area. Her interests include scientific data mining and text mining with projects in green IT, nanoscale analysis, markup languages, terminology evolution and collocation, overlapping AI & DB areas. She has 45 publications and 2 trademarks. Her students are supported by grants from PSE&G, NSF, Roche & Merck. She has served as a panelist for NSF; journal reviewer for TKDE, VLDBJ, DKE, KAIS and DMKD; and PC member at ICDM, SDM and EDBT conferences.