Infinitesimal Reasoning in Information Retrieval and Trust-Based Recommendation Systems

by

Maria Chowdhury

Bsc. Engineering, Bangladesh University of Engineering and Technology, 1993

MSc. Engineering, Bangladesh University of Engineering and Technology, 2000

A Dissertation Submitted in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Computer Science

University of Victoria

Infinitesimal Reasoning in Information Retrieval and Trust-Based Recommendation Systems

by

Maria Chowdhury

University of Victoria, BC

Bsc. Engineering, Bangladesh University of Engineering and Technology, 1993

MSc. Engineering, Bangladesh University of Engineering and Technology, 2000

Supervisory Committee

Dr. Alex Thomo. Supervisor Main, Supervisor

(Department of Computer Science, University of Victoria)

Dr. Bill Wadge. Member One, Co-Supervisor

(Department of Computer Science, University of Victoria)

Dr. Venkatesh Srinivasan. Departmental Member

(Department of Computer Science, University of Victoria)

Dr. Afzal Suleman. Outside Member

(Mechanical Engineering Department, University of Victoria)

Supervisory Committee

Dr. Alex Thomo. Supervisor Main, Supervisor

(Department of Computer Science, University of Victoria)

Dr. Bill Wadge. Member One, Co-Supervisor

(Department of Computer Science, University of Victoria)

Dr. Venkatesh Srinivasan. Departmental Member

(Department of Computer Science, University of Victoria)

Dr. Afzal Suleman. Outside Member

(Mechanical Engineering Department, University of Victoria)

ABSTRACT

We propose preferential and trust-based frameworks for Information Retrieval and Recommender Systems, which utilize the power of *Hyperreal Numbers*.

In the first part of our research, we propose a preferential framework for Information Retrieval which enables expressing preference annotations on search keywords and document elements, respectively. Our framework is flexible and allows expressing preferences such as "A is infinitely more preferred than B," which we capture by using *hyperreal numbers*. Due to widespread use of XML as a standard for representing documents, we consider XML documents in this research and propose a consistent preferential weighting scheme for nested document elements. We show how to naturally incorporate preferences on search keywords and document elements into an IR ranking process using the well-known TF-IDF (Term Frequency - Inverse Document Frequency) ranking measure.

In the second part of our research we propose a novel recommender system which enhances user-based collaborative filtering by using a trust-based social network. Again, we use hyperreal numbers and polynomials for capturing natural preferences in aggregating opinions of trusted users. We use these opinions to “help” users who are similar to an active user to come up with recommendations for items for which they might not have an opinion themselves. We argue that the method we propose reflects better the real life behaviour of the people. Our method is justified by the experimental results; we are the first to break a stated “barrier” of 0.73 for the mean absolute error (MAE) of the predicted ratings. Our results are based on a large, real life dataset from Epinions.com, for which, we also achieve a prediction coverage that is significantly better than that of the state-of-the-art methods.

Contents

List of Figures

ACKNOWLEDGEMENTS

I am very much grateful to:

my supervisors Alex Thomo and Bill Wadge, for giving me the expert guidance and encouragement that enabled me to accomplish my work.

Dean Office, Faculty of Engineering, University of Victoria for funding me with a Scholarship.

my Supervisory committee members for showing great interest in my work.

DEDICATION

I appreciate the mental support of my daughter Tanha Kabir and my friend Shohreh Hadian to achieve my degree.

Chapter 1

Introduction

The exponential growth of Internet is leading us to a world of *information abundance*. Processing of this huge available information is cumbersome and also frustrating as most of the times we end up with unnecessary information.

Naturally, the information importance for different people can be very different. A piece of information may be very important for one person but useless for another. Also the importance of a particular piece of information might be different at different situations. What is needed is a mechanism enabling a person to express his/her preferences in order to filter out unnecessary information and achieve his/her desired information.

On the other hand, we observe the significant reliance of people on other people's recommendations to choose the right item from a set of numerous choices. This reliance or trust on other people's recommendations varies. In real life we have different degrees of trust for different people. As a result we need a mechanism to formulate the trust value for different recommendations.

To deal with the above problems, we have introduced intuitive frameworks for reasoning based on qualitative and quantitative preferences for enhancing Information Retrieval and Trust-based Recommender Systems. What is instrumental to these frameworks is the use of hyperreal numbers. This thesis is organized into two parts. In the following we briefly describe each one.

Preferential Infinitesimals for Information Retrieval.[1] In this part of the thesis, we propose a framework for preferential information retrieval by incorporating in the document ranking process preferences given by the user or the system administrator. Namely, in our proposed framework, the user has the option of weighting the search keywords, whereas the system administrator has the option of weighting structural elements of the documents. We address both facets of preferential weighting by using hyperreal numbers, which form a superset of the real numbers, and in our context, serve the purpose of specifying natural preferences of the form "*A* is infinitely more preferred than *B*."

Trust-Based Infinitesimals for Enhanced Collaborative Filtering.[2] In this part of the thesis, we propose a novel recommender system which enhances user-based collaborative filtering by using a trust-based social network. Our main idea is to use infinitesimal numbers and polynomials for capturing natural preferences in aggregating opinions of trusted users. We use these opinions to "help" users who

[1] [1] and [2] encompass this part of thesis.

[2] [3] encompasses this part of thesis.

are similar to an active user to come up with recommendations for items for which they might not have an opinion themselves. We argue that the method we propose reflects better the real life behaviour of the people. Our method is justified by the experimental results; we are the first to break a stated “barrier” of 0.73 for the mean absolute error (MAE) of the predicted ratings. Our results are based on a large, real life dataset from Epinions.com, for which, we also achieve a prediction coverage that is significantly better than that of the state-of-the-art methods.

Chapter 2

Preferential Infinitesimals for Information Retrieval

2.1 Introduction

In this chapter we introduce a new framework for preferential Information Retrieval. Specifically, we propose to annotate the search keywords and document elements by hyperreal numbers in order to capture both quantitative and qualitative preferences.

Keyword Preferences. To illustrate preferences on keywords, suppose that a user wants to retrieve documents on research and techniques for "music-information-retrieval." Also, suppose that the user is a fan of Google technology. As such, this user would probably give to a search engine the keywords:

music-information-retrieval, google-search, google-ranking.

It is interesting to observe that if the user specifies these keywords in Google, then she gets a list of only *three*, low quality, pages. What happens is that the true, highly informative pages about “music-information-retrieval” are lost (or insignificantly ranked) in the quest of trying to serve the “google-search” and “google-ranking” keywords. Unfortunately, in Google and other search engines, the user cannot explicitly specify her real preferences among the specified keywords. In this example, what the user needs is a mechanism for saying that “music-information-retrieval” is of primary importance or *infinitely* more important than “google-search” and “google-ranking,” and thus, an informative page about “music-information-retrieval” should be retrieved and highly ranked even if it does not relate to Google technologies.

Structural Preferences. The other facet of using preferential weights is for system administrators to annotate structural parts of the documents in a given corpus. In practice, most of the documents are structured, and often, certain parts of them are more important than others. While our proposed ideas can be applied on any corpus of structured documents, due to the wide spread of XML as a standard for representing documents, we consider in this research XML documents which conform to a given schema (DTD)[1]. In the same spirit as for keyword preferences, we will use hyperreal weights to denote the importance of different elements in the schema and documents.

To illustrate preferences on structural parts of documents, suppose that we have a corpus of documents representing research papers, and a user is searching for a

[1] Document Type Definition

specific keyword. Now, suppose that the keyword occurs in the *title* element of one paper and in the *references* element of another paper. Intuitively, the paper having the keyword in the *title* should be ranked higher than the paper containing the keyword in the *references* element as the title of a paper usually bears more representative and concise information about the paper than the reference entries do. In fact, one could say that terms in the title (and abstract) are *infinitely* more important than terms in the references entries as the latter might be there completely incidental.

While weighting of certain parts of documents has been considered and advocated in the folklore (cf. [4, 5]), to the best of our knowledge there is no work dealing with inferring a consistent weighting scheme for nested XML elements based on the weights that a system administrator gives to DTD elements. As we explain in Section 2.4, there are tradeoffs to be considered and we present a solution that properly normalizes the element weights producing values which are consistent among sibling elements and never greater than the normalized weight of the parent element, thus respecting the XML hierarchy.

Contributions. Specifically, our contributions in this research are as follows.

1. We propose using hyperreal numbers (see [6]) to capture both "quantitative" and "qualitative" user preferences on search keywords. The set of hyperreal numbers includes the real numbers which can be used for expressing "quantitative" preferences such as, say "A is twice more preferred than B," as well as *infinitesimal* numbers, which can be used to express "qualitative" preferences

such as, say "A is infinitely more preferred than B." We argue that without such qualitative preferences there is no guarantee that an IR system would not override user preferences in favor of other measures that the system might use.

2. We extend the ideas of using hyperreal numbers to annotating XML (DTD) schemas. This allows system administrators to preferentially weight structural elements in XML documents of a given corpus. We present a normalization method which produces consistent preferential weights for the elements of any XML document that complies to an annotated DTD schema.

3. We adapt the well-known TF-IDF ranking in IR systems to take into consideration the preferential weights that the search keywords and XML elements can have. Our extensions are based on symbolic computations which can be effectively computed on expressions containing hyperreal numbers.

4. We present (in the appendix) illustrative practical examples which demonstrate the usefulness of our proposed preference framework. Namely, we use a full collection of speeches from the Shakespeare plays, and a diverse XML collection from INEX ([7]). In both these collections, we observed a clear advantage of our preferential ranking over the ranking produced by the classical TF-IDF method. We believe that these results encourage incorporating both quantitative and (especially) qualitative preferences into other ranking methods as well.

Organization. The rest of the chapter is organized as follows. In Section 2.2, we give an overview of hyperreal numbers and their properties. In Section 2.3, we present hyperreal preferences for annotating search keywords. In Section 2.4, we propose annotated DTDs for XML documents and address two problems for consistent weighting of document elements. In Section 2.5, we show how to extend the TF-IDF ranking scheme to take into consideration the hyperreal weights present in the search keywords and document elements. Section 2.6, we present experimental results.

2.2 Hyperreal Numbers

Hyperreal numbers were introduced in calculus to capture "infinitesimal" quantities which are infinitely small and yet not equal to zero. Formally, a number ϵ is said to be *infinitely small* or *infinitesimal* (cf. [6]) iff -a< ϵ <a for every positive *real* number a. Hyperreal numbers contain all the real numbers and also all the infinitesimal numbers. There are principles (or axioms) for hyperreal numbers (cf. [6]) of which we mention:

Extension Principle.

1. The real numbers form a subset of the hyperreal numbers, and the order relation $x < y$ for the real numbers is a subset of the order relation for the hyperreal numbers.

2. There exists a hyperreal number that is greater than zero but less than every positive real number.

3. For every real function f, we are given a corresponding hyperreal function f^*

which is called the *natural extension* of f.

Transfer Principle. Every real statement that holds for one or more particular real functions holds for the hyperreal natural extensions of these functions.

In short, the Extension Principle gives the *hyperreal* numbers and the Transfer Principle enables carrying out computation on them. The Extension Principle says that there does exist an infinitesimal number, for example ϵ. Other examples of hyperreals numbers, created using ϵ, are: ϵ^3, $100\epsilon^2$+ 51ϵ, $\epsilon/300$.

For $a, b, r, s \in \mathbb{R}^+$ and $r > s$, we have $a\epsilon^r < b\epsilon^s$, regardless of the relationship between a and b.

If $a\epsilon^r$ and $b\epsilon^s$ are used for example to denote two preference weights, then an object annotated by $a\epsilon^r$ is "infinitely less preferred" than an object annotated by $b\epsilon^s$, even though a might be much bigger than b, i.e. coefficients a and b are insignificant when the powers of ϵ are different. On the other hand, when comparing two preferential weights of the same power, as for example $a\epsilon^r$ and $b\epsilon^r$, the magnitudes of coefficients a and b become important. Namely, $a\epsilon^r \leq b\epsilon^r$ $(a\epsilon^r > b\epsilon^r)$ iff $a \leq b$ $(a > b)$.

2.3 Keyword Preferences

We propose a framework where the user can preferentially annotate the keywords by *hyperreal numbers.*

Using hyperreal annotations is essential for reasoning in terms of "infinitely more important," which is crucially needed in a scenario with numerous documents. This is because preference specification using only real numbers suffers from the possibility of producing senseless results as those preferences can get easily absorbed by other measures used by search engines. For instance, continuing the example given in the Introduction,

music-information-retrieval, google-search, google-ranking,

suppose that the user, dismayed of the poor result from Google, containing only three low quality pages, changes the query into[2]

music-information-retrieval OR google-search OR google-ranking.

It is interesting to observe that if the user specifies this (modified) query in Google, then what she gets is a list of *many* web-pages (documents)! These pages are ranked by their Google-computed importance which is by far biased toward general pages about "google-search" and "google-ranking" rather than "music-information-retrieval." The true pages about "music-information-retrieval" are simply buried under tons of other pages about "google-search" and "google-ranking" that are highly ranked, but contain "music-information-retrieval" either incidentally or not at all. Unfortunately, in Google and other search engines, the user cannot explicitly specify her real preferences among the specified keywords. In this example, what the user needs is a mechanism

[2] This second query style corresponds more closely than the first to what is known in the folklore as the popular "free text query:" a query in which the terms of the query are typed freeform into the search interface (cf. [4, 5]).

for saying that “music-information-retrieval” is of primary importance or infinitely more important than “google-search” and “google-ranking.”

But, let us suppose for a moment that Google would allow users to specify preferences expressed by real numbers. Now, imagine the user who is trying to convey that her “first and foremost” preference is for documents on “music-information-retrieval” rather than general documents about Google technology. For this, the user specifies that *music-information-retrieval* is 100 times more important than *google-search*. After all, “100 times more important” seems quite convincing in colloquial talking! However, what would happen if, according to the score computed by the search engine, general documents about *google-search* were in fact 1000 times more important than documents about *music-information-retrieval*? If the user preference levels were used to simply boost the computed document score by the same factor, then still, documents about *google-search* would be ranked higher than documents about *music-information-retrieval*. What the user would experience in this case is an “indifferent” search engine with respect to her preferences.

The solution we propose is to use hyperreal numbers for expressing preferential weights. In order to always have an effective comparison of documents with respect to a user query, we will fix an infinitesimal number, say ϵ, and build expressions on it. By the Extension Principle, such a number does exist. Now, we give the following definition.

> An *annotated free text query* is simply a set of keywords (terms) with preference weights which are polynomials of ϵ.

For all our practical purposes it suffices to consider only polynomials with coefficients in $\mathbb{R}^+$. For example, $3 + 2\epsilon + 4\epsilon^2$.

By making this restriction we are able to perform symbolic (algorithmic) computations on expressions using ϵ. All such expressions translate into operations on polynomials with real coefficients for which efficient algorithms are known (we will namely need to perform polynomial additions, multiplications and divisions[3]).

Let us illustrate our annotated queries by continuing the above example. The user can now give

$$\textit{music-information-retrieval, google-search} : \epsilon,\ \textit{google-ranking} : \epsilon^2$$

to express that she wants to find documents on Music Information Retrieval and she is interested in the Google technology for retrieving and ranking music. However, by leaving intact the *music-information-retrieval* and annotating *google-search* by ϵ and *google-ranking* by ϵ^2, the user makes her intention explicit that a document on *music-information-retrieval* is infinitely more important than any document on simply *google-search* or *google-ranking.* Furthermore, in accord with the above user expression, documents on *music-information-retrieval* and/or *google-search* are infinitely more important than documents on simply *google-ranking.* Of course, among documents on Music Information Retrieval, those which are relevant to Google search

[3] The division is performed by first factoring the highest power of ϵ. For example, $(6 + 3\epsilon + 3\epsilon^2)/(4 + 2\epsilon + 3\epsilon^2)$ is first transformed into $(6\epsilon^{-2} + 4\epsilon^{-1} + 3)/(3\epsilon^{-2} + 2\epsilon^{-1} + 4)$, and then we perform the division as we would do for $(6x^2 + 4x + 3)/(3x^2 + 2x + 4)$. Observe that, as ϵ is infinitely small, ϵ^{-1} is *infinitely big.*

and Google ranking are more important.

We note that our framework also allows the user to specify "soft" preference levels. For example, suppose that the user changes her mind and prefers to have both *google-search* and *google-ranking* in the same "hard" preference level as determined by the power of infinitesimal ϵ. However, she still prefers, say "twice more," *google-search* over *google-ranking*. In this case, the user gives

$$\textit{music-information-retrieval, google-search} : 2\epsilon,\ \textit{google-ranking} : \epsilon.$$

2.4 Preferentially Annotated XML Schemas

In this section, we consider the problem of weighting the structural elements of documents in a corpus with the purpose of influencing an information retrieval system to take into account the importance of different elements during the process of document ranking. Due to the wide spread of XML as a standard for representing documents, we consider in this research XML documents which conform to a given schema (DTD). In the same spirit as in the previous section, we will use hyperreal weights to denote the importance of different elements in the schema and documents.

While the idea of weighting the document elements is old and by now part of the folklore (cf. [5]), to the best of our knowledge, there is no work that systematically studies the problem of weighting XML elements. The problem becomes challenging when elements can possibly be nested inside other elements which can be weighted as well, and one wants to achieve a consistent weight normalization reflecting the true preferences of a system administrator. Another challenging problem, as we explain in Subsection 2.4.4, is determining the right mapping of weights from the elements of a DTD schema into the elements of XML documents.

2.4.1 Hyperreal weights

In our framework, the system administrator is enabled to set the importance of various XML elements/sections in a DTD schema. For example, she can specify that the *keywords* elements of documents in an XML corpus, with "research activities" as the main theme, is more important than than a section, say on *related work*. Intuitively, an occurrence of a search term in the *keywords* section is way more important than an occurrence in the *related work* section as the occurrence in the latter might be completely incidental or only loosely related to the main thrust of the document.

Thus, in our framework, we allow the annotation of XML elements by weights being, as in the previous section, polynomials of a (fixed) infinitesimal ϵ.

2.4.2 DTDs

Let Σ be the (finite) tag alphabet of a given XML collection, i.e. each tag is an element of Σ. Then, a DTD D is a pair (d, r) where d is a function mapping Σ-symbols to regular expressions on Σ and r is the root symbol (cf. [8]).

A *valid* XML document complying to a DTD $D = (d, s)$ can be viewed as a tree, whose root is labeled by r and every node labeled, say by a, has a sequence of children

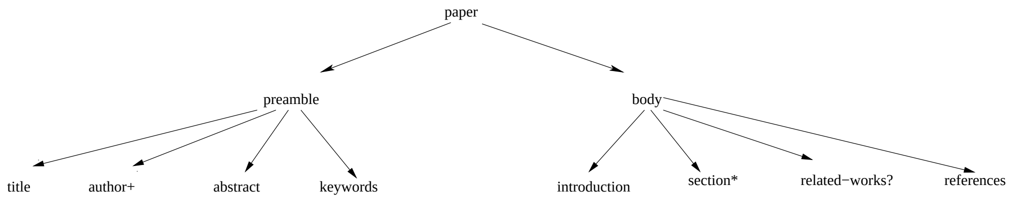


Figure 2.1: Tree structure of example DTD.

whose label concatenation, say $bc \dots x$, is in $L(d(a))$.

A simple example of a DTD defining the structure of some XML research documents is the following:

$$\begin{aligned} \text{paper} &\rightarrow \text{preamble body} \\ \text{preamble} &\rightarrow \text{title author}^{+} \text{ abstract keywords} \\ \text{body} &\rightarrow \text{introduction section}^{*} \text{ related-work? references} \end{aligned}$$

where

'+' implies "one or more," '$*$' implies "zero or more" and '?' implies "zero or one" occurrences of an element.

In essence, a DTD D is an extended context-free grammar, and a valid XML document with respect to D is a parse tree for D.

2.4.3 Annotated DTDs

To illustrate annotated DTDs, let us suppose that the system administrator wants to express that in the *body* element, the *introduction* is twice more important than a *sec-*

tion, and both are infinitely more important than *related-work* and *references*, with the latter being infinitely less important than the former, we would annotate the rule for *body* as follows: body → (introduction : 2) (section : 1)* (related-work : ϵ)? (references : ϵ^2).

Further annotations, expressing for example that the *preamble* element is three times more important than the *body* element, and in the *preamble*, the *keywords* element is 5 times more important than *title* and 10 times more important than the rest, would lead to having the following annotated DTD:

$$\begin{aligned}
\text{paper} &\rightarrow (\text{preamble} : 3)\ (\text{body} : 1)\\
\text{preamble} &\rightarrow (\text{title} : 2)\ (\text{author} : 1)^+\ (\text{abstract} : 1)\ (\text{keywords} : 10)\\
\text{body} &\rightarrow (\text{introduction} : 2)\ (\text{section} : 1)^*\ (\text{related-work} : \epsilon)?\ (\text{references} : \epsilon^2).
\end{aligned}$$

Since an annotated element can be nested inside other elements, which can be annotated as well, the natural question that now arises is: How to compute the actual weight of an element in a DTD? One might be tempted to think that the actual weight of an element should obtained by multiplying its (annotation) weight by the weights of all its ancestors. However by doing that, we could get strange results as for example a possibly increasing importance weight as we go deep down in the XML element hierarchy.

What we want here is "an element to never be more important than its parent." For this, we propose normalizing the importance weights assigned to DTD elements. There are two ways for doing this. Either divide the weights of a rule by the sum of the rule's weights, or divide them by the maximum weight of the rule. In the first way, the weight of the parent will be divided among the children. On the other hand, in the second way, the weight of the most important child will be equal to the weight of the parent.

The drawback of the first approach is that the more children there are, the lesser their weight is. Thus, we opt for the second way of weight normalization as it better corresponds to the intuition that nesting in XML documents is for adding structure to text rather than hierarchically dividing the importance of elements.

For example, in the above DTD, for the children of *preamble*, we normalize dividing by the greatest weight of the rule, which is 10. Normalizing in this way the weights of all the rules, we get

$$\begin{aligned}
\text{paper} &\rightarrow (\text{preamble} : 1)\ (\text{body} : 1/3) \\
\text{preamble} &\rightarrow (\text{title} : 1/5)\ (\text{author} : 1/10)^{+}\ (\text{abstract} : 1/10)\ (\text{keywords} : 1) \\
\text{body} &\rightarrow (\text{introduction} : 1)\ (\text{section} : 1/2)^{*}\ (\text{related-work} : \epsilon/2)?\ (\text{references} : \epsilon^2/2).
\end{aligned}$$

After such normalization, for determining the actual weight of an element, we multiply its DTD weight by the weights of all its ancestors. For example, the weight of a *section* element is $(1/3) \cdot (1/2)$.

As mentioned earlier, under this weighting scheme, the most important child of a parent has the same importance as the parent itself. Thus, for instance, element *introduction* has the same importance $(1/3)$ as its parent *body*. Note that the weight normalization can of course be automatically done by the system, while we annotate using numbers that are more comfortable to write.

2.4.4 Weighting Elements of XML Documents

In the previous section, we described how to compute the weight of an element in a DTD. However, the weight of an element in an XML document depends not only on the DTD, but also on the particular structure of the document. This is because the same element might occur differently nested in different valid XML documents. For example, if we had an additional rule, section → (title : 1) (text : 1/2), in our annotated DTD, then, given a valid XML document, the weight of a *title* element depends on the particular nesting of this element. Namely, if the nesting is

$$\langle \text{paper} \rangle \langle \text{preamble} \rangle \langle \text{title} \rangle \ldots \langle /\text{title} \rangle \ldots \langle /\text{preamble} \rangle \ldots \langle /\text{paper} \rangle$$

then the normalized weight of the *title* element is 1/5. On the other hand, if the nesting is

$$\langle \text{paper} \rangle \ldots \langle \text{body} \rangle \langle \text{section} \rangle \langle \text{title} \rangle \ldots \langle /\text{title} \rangle \ldots \langle /\text{section} \rangle \ldots \langle /\text{body} \rangle \langle /\text{paper} \rangle$$

then the normalized weight of the *title* element is $(1/3) \cdot (1/2) \cdot 1 = 1/6$.

In general, in order to derive the correct weight of an element in an XML document, we need to first build the element tree of the document. This will be a parse tree for the context-free grammar corresponding to the DTD. For each node a of this tree with children $bc \dots x$, there is a unique rule $a \to r$ in the DTD such that word $bc \dots x$ is in $L(r)$.

Naturally, we want to assign weights to a's children b, c, $\dots$, x based on the weights in annotated expression r. Thus, the question becomes how to map the weights assigned to the symbols of r to the symbols of word $bc \dots x$.

Since b, c, $\dots$, x occur in r, this might seem as a straightforward matter. However, there is subtlety here arising from the possibility of ambiguity in the regular expression. For example, suppose the (annotated) expression r is $(b:1+c:1)^*(b:2)(b:3)^*$, and element a has three children labeled by b. Surely, bbb is in $L(r)$, but what label should we assign to each of b's? There are three different ways of assigning weights to these b's: $(b:1)(b:1)(b:2)$, $(b:1)(b:2)(b:3)$, and $(b:2)(b:3)(b:3)$.

However, according to the SGML standard (cf. [9]), the only allowed regular expressions in the DTD rules are those for which we can uniquely determine the correspondence between the symbols of an input word and the symbols of the regular expression. These expressions are called "1-unambiguous" in [9].

For such an expression r, given a word $bc \dots x$ in $L(r)$, there is a unique mapping of word symbols b, c, $\dots$, x to expression symbols. Thus, when r is annotated with symbol weights, we can uniquely determine the weights for each of the b, c, $\dots$, x word symbols.

Based on all the above, we can state the following theorem.

Theorem 1. *If T is a valid XML tree with respect to an annotated DTD D, then* *based on the weight annotations of D, there is a unique weight assignment to each node of T.*

Now, given an XML document, since there is unique path from the root of an XML document to a particular element, we have that

Corollary 1. *Each element of a valid XML document is assigned a unique weight.*

The unique weight of an element is obtained by multiplying its local node weight with the weights of the ancestor nodes on the unique path connecting the element with the document root.[4]

2.5 Preferential Term Weighting and Document Scoring

Early scoring schemes were based on the Boolean model in which only the mere occurrence of terms in documents really matters. The next step was to consider the intuition that a document with more occurrences of a query term is more relevant to the query. The most popular measure reflecting this intuition is the *term frequency* (TF), which is computed as the normalized frequency of a term occurring in a document.

Formally, let V (vocabulary) be the set of distinctive terms in a collection C of documents. Denote by m and n the cardinalities of V and C respectively. Let t_i be

[4] All weights are considered being normalized.

term in V and d_j a document in C. Suppose that t_i occurs f_{ij} times in d_j. Then, the normalized term frequency of t_i in d_j is

$$tf_{ij} = \frac{f_{ij}}{max\{f_{1j}, \ldots, f_{mj}\}},$$

where the maximum is in fact computed over the terms that appear in document d_j.

Considering now XML documents whose elements are weighted based on annotated DTDs, we have that *not* all occurrences of a term "are created equal." For instance, continuing the example in Section 2.4, an occurrence of a term t_i in the *keywords* element of a document is 5 times more important than an occurrence (of t_i) in the *title*, and infinitely more important than an occurrence in the *related-work* element.

Hence, we refine the TF measure to take the importance of XML elements into account. When an XML document conforms to an annotated DTD, each element e_k will be accordingly weighted, say by w_k.

Suppose that term t_i occurs f_{ijk} times in element e_k of document d_j. Now, we define the normalized term frequency of t_i in d_j as

$$tf_{ij} = \frac{\sum_k w_k f_{ijk}}{max\{\sum_k w_k f_{1jk}, \ldots, \sum_k w_k f_{mjk}\}}.$$

For example, suppose that t_i occurs

- once in the *keywords* element,

- twice in the *abstract* element,
- three times in the *section* elements,
- four times in the *related-work* element, and
- twice in the *references* element

of document d_j. Then, the numerator of the tf_{ij} fraction will be

$$1 \cdot 1 \cdot 1 + 1 \cdot (1/10) \cdot 2 + (1/3) \cdot (1/2) \cdot 3 + (1/3) \cdot (\epsilon/2) \cdot 4 + (1/3) \cdot (\epsilon^2/2) \cdot 2$$
$$= 1.7 + (2/3) \cdot \epsilon + (1/3) \cdot \epsilon^2.$$

The other popular measure used in Information Retrieval is the *inverse document frequency* (IDF) which is used jointly with the TF measure. IDF is based on the fraction of documents which contain a query term. The intuition behind IDF is that a query term that occurs in numerous documents is not a good discriminator, or does not bear to much information, and thus, should should be given a smaller weight than other terms occurring in few documents. The weighting scheme known as TF*IDF, which multiplies the TF measure by the IDF measure, has proved to be a powerful heuristic for document ranking, making it the most popular weighting scheme in Information Retrieval (cf. [10, 4, 5]).

Formally, suppose that term t_i occurs n_i times in a collection of n elements. Then, the *inverse document frequency* of t_i is defined to be

$$idf_i = \log \frac{n}{n_i}.$$

IDF has a natural explanation from an information theoretic point of view. If we consider a term t_i as a "message" and $p_i = \frac{n_i}{n}$ as the probability of receiving message t_i, then, in Shannon's information theory [11], the information that the message carries is quantified by

$$I_i = -\log p_i,$$

which coincides with the IDF measure. The connection is clear; terms occurring in too many documents do not carry too much information for "discriminating" documents ([12]). On the other hand, terms that occur in few documents carry more information and hence have more discriminative power.

In XML Information Retrieval, considering each XML element that contains text as a mini-document, we can compute multiple IDF scores for a given term. Note that here, we restrict ourselves to *textual* elements only, i.e. those elements that contain terms. For instance, in the above example, *introduction* is a textual element, while *body* is not.

Depending on the importance weight of each textual element, the IDF scores should be appropriately weighted. Intuitively, in the above example, the IDF score of a term with respect to the *related-work* elements is infinitely less important than the IDF score of the term with respect to say *introduction* elements.

Formally, let E be the set of textual element-weight pairs (e_h, w_h) extracted from

XML document collection C. This set is finite because C is finite, and for each element in an XML document, there is a unique weight assigned to it (see Corollary 1).

For a textual element-weight pair (e_h, w_h), let n_h be the total number of such elements in the XML documents in collection C. Suppose that a term t_i occurs in n_{hi} of these e_h elements (of weight w_h). Then, we define the IDF of t_i with respect to these elements as

$$idf_{hi} = \log \frac{n_h}{n_{hi}}.$$

Next, we define the IDF score of a term t_i with respect to the whole document collection as

$$idf_i = \frac{\sum_h w_h \cdot idf_{hi}}{\sum_h w_h}.$$

This is the weighted average of IDF scores computed for each textual element-weight pair (e_h, w_h).

Finally, the TF*IDF weighting scheme combines the term frequency and inverse document frequency, producing a composite weight for each term in each document. Namely, the TF*IDF weighting scheme assigns to term t_i a weight in document d_j given by

$$tf\,idf_{ij} = tf_{ij} \times idf_i\,.$$

In the vector space model, every document is represented by a vector of weights which are the TF*IDF scores of the terms in the document. For the other terms in vocabulary V that do not occur in a document, we have a weight of zero.

Similarly, a query q can be represented as a vector of weights with non-zero weights for the terms appearing in the query. The weights are exactly those hyperreal numbers specified by the user multiplied by the IDF scores of the terms.

Now, we want to rank the documents by computing their similarly score with respect to a query q. The most popular similarity measure is the *cosine similarity*, which for a document d_j with weight vector $\mathbf{w}_j$ and a query q with weight vector $\mathbf{w}_q$ is

$$cosine(\mathbf{w}_j, \mathbf{w}_q) = \frac{\langle \mathbf{w}_j, \mathbf{w}_q \rangle}{||\mathbf{w}_j|| \times ||\mathbf{w}_q||} = \frac{\sum_{i=1}^{m} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{m} w_{ij}^2} \times \sqrt{\sum_{i=1}^{m} w_{iq}^2}},$$

where m is the cardinality of vocabulary V.

The above formula naturally combines the query preference weights, XML element weights, and Information Retrieval measures. Note that, we can in fact rank documents using instead the square of the cosine similarity. Thus, we only need to compare fractions of polynomial expressions based on the (fixed) infinitesimal ϵ. As such, these expressions allow for an algorithmic (symbolic) comparison procedure for ranking XML documents.

Finally, the query can be a complete document in its own. Such queries are of the type: Find all the documents which are similar to a given document. We derive weights for the elements of the query document in exactly the same manner as described in Section 2.4. The vector of weights for the query document is computed as for any other document in the collection. Then, this vector is compared against the vectors of the documents in the collection by computing the cosine similarity as described above.

2.6 Experiments

Here, we describe experiments to evaluate our framework and to illustrate our ideas. For this purpose, we implemented a system incorporating our proposed framework and compared its ranking effectiveness with that of a system that ranks using the classical TF-IDF measure.

Our main research question is:

Does our preferential IR improve users' search experience compared to a traditional IR?

Here we provide practical evidence that our preferential IR does indeed perform better than a traditional IR.

As described in the previous sections, we annotated XML schema elements and search keywords in order to mark their importance in ranking the documents. We designed our experiments for both document retrieval and element retrieval. We used the following corpora as test-beds.

Corpus I On-line Internet Shakespeare Edition of the English Department ([13]), University of Victoria for element retrieval. This corpus consists of all the

Shakespeare plays in XML format. The elements of interest are the speeches which total more than 33,000. For this corpus we consider all the speeches to be of the same importance, and thus, only search keyword preferences are in fact relevant for this corpus in influencing the ranking process.

Corpus II An INEX (INitiative for the Evaluation of XML retrieval) (cf. [7]) corpus. INEX is a collaborative initiative that provides reference collections (corpora). For evaluating our method, we have chosen a collection named "*topic-collection*" with numerous XML documents of moderate size. The topics of documents vary from *climate change* to *space exploration*. We preferentially annotated the DTD of this collection and gave many preferentially annotated search queries, some of which we show in this section.

2.6.1 Queries and Results for Corpus I

For the On-line Internet Shakespeare Edition, we created many search queries and observed that for all of them the highly ranked *speech* elements were much more relevant than the *speech* elements which were highly ranked by a traditionally implemented IR system. Here, due to space constraints, we only present two representative examples of search queries.

Q1. *romeo, iuliet*: ϵ, *loue*: ϵ^2. This query says that the user is mostly interested in the keyword *'romeo'* and then *'iuliet'* and least interested in *'loue'* (love).

Preferential IR Result for Q1. The *speech* element which was the top ranked by our preferential IR system is:

> <s> The excellent Tragedie And Ile informe you how these things fell out. Iuliet here slaine was married to that Romeo, Without her Fathers or her Mothers grant: The Nurse was priuie to the marriage. The balefull day of this vnhappie marriage, VVas Tybalts doomesday: for which Romeo VVas banished from hence to Mantua. He gone, her Father sought by soule constraint To marrie her to Paris: but her Soule (Loathing a second Contract) did refuse To giue consent; and therefore did she vrge me Hither to finde a meanes she might auoyd What so her Father sought to force her too Or els all desperately she threatned Euen in my presence to dispatch of her selfe. Then did I giue her, (tutord my mine arte) A potion that should make her seeme as dead: And told her that I would with all post speed Send hence to Mantua for her Romeo, That he might come and take her from the Toombe, But he that had my Letters (Frier Iohn) Seeking a Brother to associate him, VVhereas the sicke infection remaind, VVas stayed by the Searchers of the Towne. But Romeo vnderstanding by his man, That Iuliet was deceasde, returnde in post Vnto Verona for to see his loue. VVhat after happened touching Paris death, Or Romeos is to me vnknowne at all. But when I came to take the Lady hence, I found them dead, and she awakt from sleep: VVhom faine I would haue taken from the tombe, VVhich she refused seeing Romeo dead. Anone I heard the watch and then I fled, VVhat after happened I am ignorant of. And if in this ought haue miscaried By of Romeo and Iuliet. By me, or by my meanes let my old life Be sacrificd some houre before his time. To the most strickest rigor of the Law. </s>

Traditional IR Result for Q1. The *speech* element which was the top ranked by the traditional IR system is:

> <s> Consider what you first did sweare vnto: To fast, to study, and to see no woman: Flat treason gainst the kingly state of youth. Say, Can you fast? your stomacks are too young: And abstinence ingenders maladies. And where that you haue vowd to studie (Lordes) In that each of you haue forsworne his Booke. Can you still dreame and poare and thereon looke. For when would you my Lord, or you, or you, Haue found the ground of Studies excellence, Without the beautie of a womans face? From womens eyes this doctrine I deriue, They are the Ground, the Bookes, the Achadems, From whence doth spring the true Promethean fire. Why vniuersall plodding poysons vp The nimble spirites in the arteries, As motion and long during action tyres The sinnowy vigour of the trauayler. Now for not looking on a womans face, You haue in that forsworne the vse of eyes: And studie too,

> the causer of your vow. For where is any Authour in the worlde, Teaches such beautie as a womas eye: Learning is but an adiunct to our selfe, And where we are, our Learning likewise is. Then when our selues we see in Ladies eyes, With our selves. Do we no likewise see our learning there? O we haue made a Vow to studie, Lordes, And in that Vow we haue forsworne our Bookes: For when would you (my Leedge) or you, or you? In leaden contemplation haue found out Such fierie Numbers as the prompting eyes, Of beautis tutors haue inritcht you with: Other slow Artes intirely keepe the braine: And therefore finding barraine practizers, Scarce shew a haruest of their heauie toyle. But called Loues Labor's lost. But Loue first learned in a Ladies eyes, Liues not alone emured in the braine: But with the motion of all elamentes, Courses as swift as thought in euery power, And giues to euery power a double power, Aboue their functions and their offices. It addes a precious seeing to the eye: A Louers eyes will gaze an Eagle blinde. A Louers eare will heare the lowest sound. When the suspitious head of theft is stopt. Loues feeling is more soft and sensible, Then are the tender hornes of Cockled Snayles. Loues tongue proues daintie, Bachus grosse in taste, For Valoure, is not Loue a Hercules? Still clyming trees in the Hesperides. Subtit as Sphinx, as sweete and musicall, As bright Appolos Lute, strung with his haire. And when Loue speakes, the voyce of all the Goddes, Make heauen drowsie with the harmonie. Neuer durst Poet touch a pen to write, Vntill his Incke were tempred with Loues sighes: O then his lines would rauish sauage eares, And plant in Tyrants milde humilitie. From womens eyes this doctrine I deriue. They sparcle still the right promethean fier, They are the Bookes, the Artes, the Achademes, That shew, containe, and nourish all the worlde. Els none at all in ought proues excellent. Then fooles you were, these women to forsweare: Or keeping what is sworne, you will proue fooles, For Wisedomes sake, a worde that all men loue: Or for Loues sake, a worde that loues all men. Or for Mens sake, the authour of these Women: Or Womens sake, by whom we Men are Men. Lets vs once loose our othes to find our selues, Or els we loose our selues, to keepe our othes: It is Religion to be thus forsworne. For A pleasant conceited Comedie: For Charitie it selfe fulfilles the Law: And who can seuer Loue from Charitie. </s>

One can easily observe that the first speech element is clearly more relevant to the given query than the second element which is in fact quite relevant to word "loue" but not at all to the first two query keywords. We see here that the traditional TF-IDF measure has essentially ignored the first two keywords in favor of the third one just because the latter occurs too frequently in the shown document.

In the following, we show the second search query and the top-ranked speech elements for our preferential system as well as for the traditional one. For this query, similarly as for the first query, we observe that the result of the preferential system is better than that of the traditional system.

Q2. *henry, death*: ϵ, *king*: ϵ^2. This query says that the user is mostly interested in the keyword *'henry'* and then *'death'* and least interested in *'king'*.

Preferential IR Result for Q2. The *speech* element which was the top ranked by our preferential IR system is:

> <s> Which whiles it lasted, gaue King Henry light. O Lancaster! I feare thy ouerthrow, More then my Bodies parting with my Soule: My Loue and Feare, glew'd many Friends to thee, And now I fall. Thy tough Commixtures melts, Impairing Henry, strength'ning misproud Yorke; And whether flye the Gnats, but to the Sunne? And who shines now, but Henries Enemies? O Phoebus! had'st thou neuer giuen consent, That Phaeton should checke thy fiery Steeds, Thy burning Carre neuer had scorch'd the earth. And Henry, had'st thou sway'd as Kings should do, Or as thy Father, and his Father did, Giuing no ground vnto the house of Yorke, They neuer then had sprung like Sommer Flyes: I, and ten thousand in this lucklesse Realme, Hed left no mourning Widdowes for our death, And thou this day, had'st kept thy Chaire in peace. For what doth cherrish Weeds, but gentle ayre? And what makes Robbers bold, but too much lenity? Bootlesse are Plaints, and Curelesse are my Wounds: No way to flye, no strength to hold out flight: The Foe is mercilesse, and will not pitty: For at their hands I haue deseru'd no pitty. The ayre hath got into my deadly Wounds. </s>

Traditional IR Result for Q2. The *speech* element which was the top ranked by the traditional IR system is:

> <s> King. So, if a Sonne that is by his Father sent about Merchandize, doe sinfully miscarry vpon the Sea; the im- putation of his wickednesse, by your rule, should be im- posed vpon his Father that sent him: or if a Seruant, vn- der his Masters command, transporting a summe of Mo- ney, be assayled by Robbers, and dye in many irreconcil'd Iniquities; you may call the businesse of the Master the

author of the Seruants damnation: but this is not so: The King is not bound to answer the particular endings of his Souldiers, the Father of his Sonne, nor the Master of his Seruant; for they purpose not their death, when they purpose their seruices. Besides, there is no King, be his Cause neuer so spotlesse, if it come to the arbitre- ment of Swords, can trye it out with all vnspotted Soul- diers: some (peraduenture) haue on them the guilt of premeditated and contriued Murther; some, of beguiling Virgins with the broken Seales of Periurie; some, making the Warres their Bulwarke, that haue before go- red the gentle Bosome of Peace with Pillage and Robbe- rie. Now, if these men haue defeated the Law, and out- runne Natiue punishment; though they can out-strip men, they haue no wings to flye from God. Warre is his Beadle, Warre is his Vengeance: so that here men are punisht, for before breach of the Kings Lawes, in now the Kings Quarrell: where they feared the death, they haue borne life away; and where they would bee safe, they perish. Then if they dye vnprouided, no more is the King guiltie of their damnation, then hee was be- fore guiltie of those Impieties, for the which they are now visited. Euery Subiects Dutie is the Kings, but euery Subiects Soule is his owne. Therefore should euery Souldier in the Warres doe as euery sicke man in his Bed, wash euery Moth out of his Conscience: and dying so, Death is to him aduantage; or not dying, the time was blessedly lost, wherein such preparation was gayned: and in him that escapes, it were not sinne to thinke, that making God so free an offer, he let him out- liue that day, to see his Greatnesse, and to teach others how they should prepare. </s>

2.6.2 Queries and Results for Corpus II

The DTD defining the structure of this XML corpus is as follows:

inex_topic → title mmtitle* castitle* description narrative

We preferentially annotated this DTD as follows:

inex_topic → (title:1) (mmtitle:1/10)* (castitle:1/100)* (description: ϵ) (narrative: ϵ^2).

We had numerous runs on our system with preferentially annotated queries. As an example, a preferentially annotated query is as follows.

Q1. *Norway, climate*: ϵ*, information*: ϵ^2, where the user is looking for climate information for Norway. The query says that the user is primarily interested in keyword *Norway*, next *climate* and then, the least important, *information.*

Preferential IR Result for Q1. The top-ranked document is:

<?xml version=”1.0” encoding=”ISO-8859-1” ? >

<!DOCTYPE inex_topic (View Source for full doctype...) >

− <inex_topic topic_id=”447” ct_no=”56”>

<title>Climate in Norway< /title>

<castitle> //article[about(., climate) and about(.,Norway)]< /castitle>

<description>Find information about the climate in Norway in summer.< /description>

<narrative>I would like to travel to Norway in july, but I have no idea about the weather. i don’t know which clothes to put in my bag. To be relevant, a paragraph or a document should let me know the mean average temperature in this season and the precipitation level, or just give me an information like continental climate or polar climate...< /narrative>

< /inex_topic>

Traditional IR Result for Q1. The top-ranked document is:

<?xml version=”1.0” encoding=”ISO-8859-1” ? >

− <inex_topic topic_id=”494” ct_no=”144”>

<title>ontology< /title>

<castitle> //title[about(.,ontology)]< /castitle>

<description>Find information about ontology.< /description>

<narrative>An ontology is typically a hierarchical data structure containing all the relevant entities and their relationships and rules within that domain (e.g., a domain ontology). However, computational ontology does not have to be hierarchical at all. The computer science usage of the term ontology is derived from the much older usage of the term ontology in philosophy. For it plays a very important role in information extraction, entity recognition etc., I would like to learn more information about the introduction of it and how it works. Besides, I expect to find relevant information as

elements in larger documents that deal with ontology e.g., the title of documents contains the term ontology. To be relevant, the document should contain the conception and description about ontology, something detailed about the uses of ontology as well. Information such as catalog or about specified domain without general discussion of it is not relevant.< /narrative>

< /inex_topic>

It is obvious that the top-ranked document of our preferential system is way more relevant than the top-ranked document of the traditional system. A similar observation applies to the other query examples which we give in the following.

Q2. *hurricane, information*: ϵ, where the user is primarily interested in keyword *hurricane*, and then *information*.

Preferential IR Result for Q2. The top-ranked document is:

<?xml version=”1.0” encoding=”ISO-8859-1” ? >

− <inex_topic topic_id=”530” ct_no=”23”>

<title>Hurricane satellite image< /title>

<castitle> //figure[about(.,hurricane)]< /castitle>

<mmtitle> //figure[about(.,hurricane) and about(.,src:www.katrina-hurricane.biz/images/katrina-hurricane-pic3.jpg)]< /mmtitle>

<description>Find images of hurricanes taken from satellites, similar to one image from the web.< /description>

<narrative>Because I need, for a report at school on meteorological events, to have views of hurricanes taken from satellites with clues on the size of the hurricane. The images can be in greyscale or colours and we have to see the ground or at least the shape of the coasts.< /narrative>

< /inex_topic>

Traditional IR Result for Q2. The top-ranked document is:

<?xml version=”1.0” encoding=”ISO-8859-1” ? >

− <inex_topic topic_id=”494” ct_no=”144”>

<title>ontology< /title>

<castitle> //title[about(.,ontology)]< /castitle>

<description>Find information about ontology.< /description>

<narrative>An ontology is typically a hierarchical data structure containing all the relevant entities and their relationships and rules within that domain (e.g., a domain ontology). However, computational ontology does not have to be hierarchical at all. The computer science usage of the term ontology is derived from the much older usage of the term ontology in philosophy. For it plays a very important role in information extraction, entity recognition etc., I would like to learn more information about the introduction of it and how it works. Besides, I expect to find relevant information as elements in larger documents that deal with ontology e.g., the title of documents contains the term ontology. To be relevant, the document should contain the conception and description about ontology, something detailed about the uses of ontology as well. Information such as catalog or about specified domain without general discussion of it is not relevant.< /narrative>

< /inex_topic>

Q3. *space, news*: ϵ, where the user is primarily interested in keyword *space*, and then *news*.

Preferential IR Result for Q3. The top-ranked document is:

<?xml version=”1.0” encoding=”ISO-8859-1” ? >

− <inex_topic topic_id=”415” ct_no=”5”¿

<title>space history astronaut cosmonaut engineer< /title>

<castitle> //article[about(.,space history)]//section[about(., astronaut cosmonaut engineer)]< /castitle>

<description>Find the names of the 25 five most important people involved in the space exploration.< /description>

<narrative>The aim is to write a 10 pages report on the big names in the space exploration. The relevant documents should talk about at least one of the, say 25, most important people who were

involved in the space exploration. Documents about one astronaut/cosmonaut who should not be personally mentioned in a 10 page report are not relevant. A relevant document should trace by itself an history of space exploration with mention of the big names, or be a document on one of these big names. So the context is space history and in this context I am looking for names of either astronauts, cosmonauts and/or engineers.< /narrative>

< /inex_topic>

Traditional IR Result for Q3. The top-ranked document is:

<?xml version=”1.0” encoding=”ISO-8859-1” ? >

− <inex_topic topic_id=”481” ct_no=”116”>

<title>asia ”news channel”< /title>

<castitle> //article[about(., ”news channel” + asia)]< /castitle>

<description>Find articles about any of the Asian news channel.< /description>

<narrative>The TV channels which are dedicated for News alone are gaining enormous popularity. The query is aimed at finding news channels which are from asian countries. For a document to be relevant, it should include the name of the news channel along with an asian country name.If it includes more information about the news channel it will be considered more relevant.Worldwide News channels like BBC and CNN are considered as irrelevant to the query.< /narrative>

< /inex_topic>

2.7 Conclusions

We have introduced an IR framework based on hyperreal numbers for expressing preferences on search keywords and XML nested elements. For this framework, we have shown how to extend the well-known IR ranking scheme, TF-IDF, to take into account the expressed preferences. Experimentally, we have given evidence that incorporating preferences in the ranking process of an IR system according to our proposed method is effective; a system using our extended TF-IDF measure ranks better than a system using the classical version of the TF-IDF measure.

Chapter 3

Trust-Based Infinitesimals for Enhanced Collaborative Filtering

3.1 Introduction

One of the key innovations in on-line marketing is the creation of recommender systems for suggesting new interesting items to users (or buyers). In essence a recommender system (RS) tries to predict the ratings that users would give to different items. The recommendations should be of good quality, otherwise the users would soon loose the confidence on the RS and consider it just another spamming annoyance. On the other hand, the system should be able to recommend a good range of items to the users, not just few ones. These two desiderata, the quality of the predictions, and the coverage of recommendations are often two conflicting goals. Typically, the better the quality of predictions is, the worse the coverage gets, and vice versa. In this research we propose a novel recommender system which has at the same time both high quality of predictions as well as great item coverage.

The quality of predictions is usually measured by the Mean Absolute Error (MAE), which is computed by trying to predict the existing, real user ratings (after they are hidden) and then compute the differences of the predictions from these real ratings. On the other hand, the coverage is estimated as the percentage of the existing ratings that the RS is able to approximate.

Regarding MAE, there exists a belief that there is some inherent "magic barrier" below which MAE cannot go. As the seminal work by Herlocker et. al. ([14]) puts it, the recommender systems working with ratings in a scale from one to five hit a MAE barrier of 0.73, and achieving a MAE below that is very difficult due to "the natural variability" of humans when rating items.

In this research, we are the first to break the above barrier. We achieve this by using the power of the underlying Epinions social network expressed by trust statements between the users. Furthermore, we do not improve the MAE by compromising the coverage. In fact our coverage is significantly better than the coverage of the state-of-the-art methods.

Motivation and Main Idea. On-line social networks have become a very important part of decision making and other activities in our daily lives. We use these systems to communicate with each other, to make new friends, to buy or sell products on-line, to collect reviews of products, to play games etc. On-line social networks such as *Orkut, Facebook, LinkedIn, MySpace*, etc have a huge number of users and there is a tremendous increase in the number of users every day.

We believe that looking for recommendations is one of the most important uses of social networks. In this research we introduce a new recommender system which leverages the power of a social network created by users who issue trust statements with respect to other users. The trust statements, over time, create a precious web of trust, which, as we show, can be used to significantly enhance the quality of recommendations.

Notably, our system blends together collaborative filtering and trust-based reasoning. Collaborative filtering (CF) identifies similar users based on the product ratings that the users have issued over time. The similarity between two users is typically determined by calculating the Pearson correlation between the users' rating vectors. Then, the recommendation to a user u for an item i is generated by averaging the ratings of the similar users for item i.

To illustrate, given a user, say Bob, and an item, say HP laptop, in order to generate a recommendation for HP laptop to Bob, CF will find the users who are similar to Bob, say Alice and Jon, and then average their ratings for HP laptop.

The problem is: "What if Alice or Jon, or both, do not have a rating for the HP laptop?" Clearly in such cases the system would suffer from data sparsity and the quality of recommendations will degrade.

The intuition behind our solution is to make "Alice" and/or "Jon" (in this example) "come up" with an opinion about HP laptop by using the available trust-based social network. Based on how friendship connections are evaluated in real life, we propose that a user aggregates first the opinions of his/her (immediate) friends, and considers the opinions of the friends-of-friends only if the friends are unable to provide some opinion. If the latter happens, then the opinion of a “second degree” friend who is trusted by many “first degree” friends should be more important than the opinion of some other “second degree” friend who is trusted by fewer “first degree” friends. This idea can be naturally generalized to more than one or two levels of friendship connections.

We believe that this reasoning reflects better the people’s real life behaviour; we trust our friends “infinitely” more than the friends-of-friends, and we trust them in turn “infinitely” more that the friends-of-friends-of-friends, whom, after all, in real life, we might not even know at all.

Contributions. Specifically, our contributions in this research are as follows.

1. We present a method to inject the power of a trust-based social network into Collaborative Filtering.

2. We propose the idea of having ratings which are “infinitely” more important than other ratings. We capture this by using infinitesimal numbers and polynomials. In this way we obtain a framework in which one can elegantly set qualitative preferences or semantics for a recommender system.

3. We present a recursive formula for aggregating user opinions based on our framework of rating polynomials. The aggregated opinion given to a user u for an item i is concisely expressed as a Hadamard division of two infinitesimal polynomials.

4. We present a detailed experimental evaluation of our system and show that it significantly outperforms state-of-the-art methods both with respect to the quality of recommendations as well as their coverage.

Organization. The rest of the chapter is organized as follows. In Section 3.2 we give an overview of related works. In Section 3.3 we present an outline of our method. In Section 2.2 we present the hyperreal numbers which include both the real and infinitesimal numbers. In Section 3.4 we give the main data structures used by our method. In Section 3.5 we present our hybrid CF-and-trust-based recommendation method. In Section 3.6 we show the results of our evaluation. Finally, Section 3.7 concludes the chapter.

3.2 Related Works

Using trust networks for recommender systems has been identified as a promising direction for improving the quality of recommendations. Some important works on trust-based recommender systems are [15, 16, 17, 18, 19, 20]. They study different aspects of trust-based recommenders, such as computing or inferring the trustworthiness of the users, devising effective trust metrics, applying trust-based recommendation techniques in specific domains etc.

In [21], Massa and Avesani present a deep comparative study of trust-based recommender systems vs. a classical recommender system based on Collaborative Filtering. We believe that [21] is seminal in that it shows that recommendations based on local trust neighborhoods are significantly better than recommendations based on global reputation systems such as Google's PageRank [22]. Notably, Massa and Avesani derive these results on a large, real-life dataset, which they collected from the Epinions.com site. We also use this dataset[1] for evaluating our system. Furthermore, they propose studying the effectiveness of recommender systems not only on all users and items, but also on several well-chosen, critical categories of users and items.

Evaluating recommender systems in a reliable way is certainly very important. The most authoritative work on the evaluation of recommender systems is by Herlocker et. al. [14]. It is there that the stated barrier of 0.73 on MAE is mentioned. This value is based on their experiments as well as experiments performed by other works. Another metric, that they (as well as Massa and Avesani in [21]) suggest, is the recommendations coverage which is the fraction of ratings that the system is able to predict after the ratings are hidden. We evaluate our method using both MAE and recommendations coverage.

3.3 Outline of our method

The ratings of a set of users for a set of items can be visualized as a matrix M with users as rows and items as columns. The rating that a user u has given for an item i is the value $M[u, i]$. Of course this matrix is very sparse; most of the entries are zero because typically each user has rated only a handful of items. As such, the user-item

[1] Actually, what is available is a slightly different version which can be downloaded from www.trustlet.org

matrix is never explicitly built, but this does not hinder the similarity computations of Collaborative Filtering.

The ratings of a user u are all the non-zero entries in the corresponding matrix row. Given an *active user* (row) u_a for which we want to generate recommendations, the (user-based) Collaborative Filtering method works by computing the similarity of row u_a with each other row in the matrix. Only the non-zero entries are considered. Typically, the Pearson correlation is used for determining the similarity between two rows, u_1 and u_2. Let the pairs of non-zero ratings for u_1 and u_2 be $(x_1, y_1), \ldots, (x_n, y_n)$, i.e. u_1 and u_2 have n items in common which they have *both* rated. The Pearson correlation for u_1 and u_2 is computed as

$$p_{1,2} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2}\sqrt{n \sum x_i)^2 - (\sum (y_i)^2}}.$$

The Person correlation has been shown to perform well in practice for indentifying users who have similar tastes to a given active user.

By computing the Pearson correlation of the active user u_a against the other users in the database, a set of *similar* users is identified. Let $S = \{u_1, \ldots, u_m\}$ be this set. Then, the recommendation that u_a gets for an item i is a weighted average of the ratings that the users in S have for i. Specifically, if we denote by $u_k[i]$ the rating of user u_k for item i, then the recommendation is

$$r_{a,i} \quad = \quad \frac{\sum_{u_k \in S \wedge u_k[i] \neq 0} u_k[i] \cdot p_{a,k}}{\sum_{u_k \in S \wedge u_k[i] \neq 0} p_{a,k}}. \tag{3.1}$$

The **problem** is:

> When $u_k[i]$ is zero, user u_k, who is similar to u_a, cannot participate in the computation of recommendation $r_{a,i}$, and this harms the quality of the recommendation.

To alleviate this problem we will use the available trust network to derive an opinion for a user u_k on item i. Our desiderata in using a trust network are as follows.

1. The opinions of (trusted) friends at distance d (in the trust graph) matter *infinitely* more than the opinions of friends at distance $d+1$. In other words, friends at distance $d+1$ start to matter only if the friends at distance d cannot come up with an opinion on item i.

2. If f_1 and f_2 are two friends of u_k at distance $d+1$ in the trust graph, and f_1 is trusted by more friends at distance d than f_2, then the opinion of f_1 should be more important than the opinion of f_2. In other words, even if the friends at distance d cannot come up with an opinion on item i, they can still influence the weighting of the opinions of the friends at distance $d+1$. This is recursive. As such, the friends at distance d can in fact influence the weighting of the opinions of friends at distance $d+k$, for $k \geq 1$.

In order to capture these desiderata we propose a framework based on infinitessimal numbers and polynomials.

For our purposes in this part we fix an infinitesimal number, say ϵ, and consider polynomials (or finite series) of the powers of ϵ. Such polynomials are compared using the above rules. Namely, given two polynomials p and q we first compare their terms of the lowest ϵ power. If the coefficients of these terms are equal, then we continue by comparing the terms of the second lowest ϵ power, and so on. For example, $3\epsilon + 4\epsilon^2 + 10\epsilon^3$ is smaller than $3\epsilon + 5\epsilon^2 + 6\epsilon^3$ because although their coefficients of the lowest ϵ power are equal (value 3), the coefficients of the next lowest ϵ power, ϵ^2, are not, namely the coefficient in the first polynomial is 4 and thus, smaller than 5, which is the corresponding coefficient in the second polynomial. In this example, the third terms of the polynomials are not used in the comparision. They would have been used only if all the coefficients of the previous terms were respectively equal.

Given two polynomials $p = a_1\epsilon^{r_1} + \ldots + a_n\epsilon^{r_n}$ and $q = b_1\epsilon^{r_1} + \ldots + b_n\epsilon^{r_n}$, where $a_1, \ldots, a_n \in \mathbb{R}$, $b_1, \ldots, b_n \in \mathbb{R}$ and non-zero, and $r_1, \ldots, r_n \in \mathbb{N}$, the Hadamard quotient of p by q is defined as

$$p/\!\!/q = (a_1/b_1)\epsilon^{r_1} + \ldots + (a_n/b_n)\epsilon^{r_n}.$$

We will use the Hadamard quotient of polynomials when deriving the rating predictions later on in the chapter.

3.4 Data Structures

A social network is a social structure made of nodes that are tied by one or more specific types of interdependency, such as values, visions, ideas, friendship, etc; resultant is a graph-based structure.

In this research we consider social networks based on trust statements that users have issued for other users. We call such a social network a trust graph and denote it by TG. This is a directed graph with nodes representing users, and edges representing trust statements. Specifically an edge (u, u') going from (user) node u to node u' expresses the fact that user u trusts user u'.

The other structure that is involved is a nested hash table representing the sparse matrix of the user ratings. This hash table contains user-item-rating triples (u, i, r), which are first hashed with respect to u and then i. We denote this set of user ratings by UR.

Given a (user) node u in TG, we denote the set of its immediate neighbor nodes by N_u.

3.5 Trust-Based Recommendation Polynomials

Now suppose that a user u is asked for an opinion on item i. We introduce a recursive formula which computes this opinion based on the opinions, or ratings, for i of the neighbors of u in trust graph TG. Our formula captures the intuition that for a user u the opinions of neighbors at distance d are infinitely more important than the opinions of neighbors at distance $d+1$. Thus, the opinions of u's immediate neighbors in TG are inifinitely more important than the opinions of the neighbors-of-neighbors, and so on.

User u can already have an opinion (rating) about product i. We consider the personal opinions of users to be infinitely more important than the opinions of their neighbors. When user u has already an opinion for item i, it might be tempting to think that u does not need a recommendation for i. However we argue that providing trust-based recommendations, even in such cases, is nevertheless useful. These recommendations have the potential to *influence* or *calibrate* the opinion of a user for an item against some other, competing item, for which the user had the *same* opinion initially. For example suppose that a user had tried in the past both an HP laptop and an LG one, and had created the same opinion (rating), say 4, for both of them. Now this user wants to buy an HP or LG laptop, but does not have a definitive "winner" in her head. If she gets from the neighbors the recommendations of 2 and 5 for these products, respectively, then she might better prefer the LG laptop as opposed to the HP one.

Let $o_{u,i}$ be the opinion (or rating) user u has for item i. If u does not know or has not created yet an opinion about i, then this value is 0.

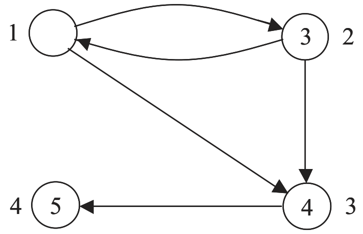


Figure 3.1: Example of a trust graph. The numbers outside the nodes are the node ids. The numbers inside the nodes are the ratings that the users have given to some item. These ratings correspond to the same item under consideration.

Our formula for computing/calibrating u's opinion on item i is

$$q_{u,i} \quad = \quad o_{u,i} + \epsilon \cdot \sum_{v \in N_u} q_{v,i} \tag{3.2}$$

where ϵ is the infinitessimal number that we fixed in Section 2.2, and N_u is the set of u's immediate neighbors in TG. We compute the above for each user u and each item i. This is "one pass" or "one iteration" over the data. We repeat this procedure several times. Initially, the $q_{u,i}$ values are equal to the $o_{u,i}$ values. After each iteration the opinions get updated with the (aggregated) opinion of neighbors, then the opinion of the neighbors-of-neighbors, and so on.

Example 1. *Consider the trust graph given in Figure 3.1. There are four users represented by the nodes 1, 2, 3, and 4. These node ids are shown in the exterior of the circles representing the nodes. In this example we consider one item only. The ratings that the users have given to this item are shown in the interior of the nodes. Node 1 does not have a rating for the item on focus.*

In the following computations we have kept only the subscript for the user and *dropped the subscript for the item as we are considering only a single item in this example.*

Initially, the q-opinions are the same as the corresponding user ratings, i.e. $q_1 = o_1 = 0$, $q_2 = o_2 = 3$, $q_3 = o_3 = 4$, *and* $q_4 = o_4 = 5$.

Now, formula 3.2, gives rise to the following equations for nodes 1, 2, 3, and 4, respectively.

$$\begin{aligned} q_1 &= o_1 + \epsilon \cdot (q_2 + q_3) \\ q_2 &= o_2 + \epsilon \cdot (q_1 + q_3) \\ q_3 &= o_3 + \epsilon \cdot q_4 \\ q_4 &= o_4. \end{aligned}$$

After the first iteration, the q-opinion values are

$$\begin{aligned} q_1 &= 0 + \epsilon \cdot (3 + 4) = 7\epsilon \\ q_2 &= 3 + \epsilon \cdot (0 + 4) = 3 + 4\epsilon \\ q_3 &= 4 + 5\epsilon \\ q_4 &= 5, \end{aligned}$$

whereas, after the second iteration, the q-opinion values are

$$\begin{aligned}
q_1 &= 0 + \epsilon \cdot (3 + 4\epsilon + 4 + 5\epsilon) = 7\epsilon + 9\epsilon^2 \\
q_2 &= 3 + \epsilon \cdot (7\epsilon + 4 + 5\epsilon) = 3 + 4\epsilon + 12\epsilon^2 \\
q_3 &= 4 + 5\epsilon \\
q_4 &= 5.
\end{aligned}$$

□

As it can be observed, the q-opinions are polynomials of the powers of ϵ with real numbers as coefficients. As such they can be compared, and thus ranked, using the rules described in Section 2.2. Namely, if there is more than one item, then when comparing the respective recommendation polynomials, the coefficients of an ϵ power become relevant only when the coefficients for all the lesser powers of ϵ are respectively the same in the polynomials that are being compared.

It can be verified that the immediate neighbors of a (user) node contribute their ratings to the coefficent of ϵ; the neighbors-of-neighbors contribute their ratings to the coefficient of ϵ^2, and so on. In general, a neighbor at distance d contributes to the coefficient of ϵ^d.

Also, it can be verified that if a neighbor at distance $d + 1$ is a direct neighbor of k neighbors at distance d, then this neighbor will contribute k times its opinion (rating) in the calculation of the coefficient of ϵ^{d+1}. Thus, the second of the desiderata in Section 3.3 is satisfied.

Several iterations of applying formula 3.2 give us polynomials which can be compared and thus ranked. What is needed next is to normalize the coefficients of these polynomials, in order for the coefficients to be in the proper rating scale (for instance, from one to five).

For this, during an iteration, we not only calculate formula 3.2, but also calculate the following formula for accumulating the counts of nodes that contribute to the coefficients of ϵ.

$$c_{u,i} \quad = \quad e_{u,i} + \epsilon \cdot \sum_{v \in N_u} c_{v,i} \tag{3.3}$$

where $e_{u,i}$ is a 0/1 integer which is 1 if there exists a rating for item i in node u, and 0 otherwise.

Finally, the coefficients of the $q_{u,i}$ polynomials are normalized by taking their Hadamard quotient with the corresponding $c_{u,i}$ polynomials.

Example 2. *Consider the trust network of the previous example (see Figure 3.1). As before we have kept only the subscript for the user and dropped the subscript for the item as there is only a single item that we are considering.*

Initially, $c_1 = e_1 = 0$ *and* $c_2 = e_2 = 1$, $c_3 = e_3 = 1$, *and* $c_4 = e_4 = 1$.

Formula 3.3, gives rise to the following equations for nodes 1, 2, 3, and 4, respec-

tively.

$$\begin{aligned} c_1 &= \epsilon \cdot (c_2 + c_3) \\ c_2 &= 1 + \epsilon \cdot (c_1 + c_3) \\ c_3 &= 1 + \epsilon \cdot c_4 \\ c_4 &= 1. \end{aligned}$$

After the first pass, we have

$$\begin{aligned} c_1 &= 0 + \epsilon \cdot (1 + 1) = 2\epsilon \\ c_2 &= 1 + \epsilon \cdot (0 + 1) = 1 + \epsilon \\ c_3 &= 1 + \epsilon \\ c_4 &= 1, \end{aligned}$$

whereas, after the second pass, we have

$$\begin{aligned} c_1 &= 0 + \epsilon \cdot (1 + \epsilon + 1 + \epsilon) = 2\epsilon + 2\epsilon^2 \\ c_2 &= 1 + \epsilon \cdot (2\epsilon + 1 + \epsilon) = 1 + \epsilon + 3\epsilon^2 \\ c_3 &= 1 + \epsilon \\ c_4 &= 1. \end{aligned}$$

Finally, the normalized polynomials are obtained by taking the Hadamard quotients *of the q_u polynomials with the corresponding c_u polynomials. We have*

$$\begin{aligned}
q_1 /\!\!/ c_1 &= (7\epsilon + 9\epsilon^2) /\!\!/ (2\epsilon + 2\epsilon^2) = 3.5\epsilon + 4.5\epsilon^2 \\
q_2 /\!\!/ c_2 &= (3 + 4\epsilon + 12\epsilon^2) /\!\!/ (1 + \epsilon + 3\epsilon^2) = 3 + 4\epsilon + 4\epsilon^2 \\
q_3 /\!\!/ c_3 &= (4 + 5\epsilon) /\!\!/ (1 + \epsilon) = 4 + 5\epsilon \\
q_4 /\!\!/ c_4 &= 5 /\!\!/ 1 = 5.
\end{aligned}$$

□

If only a single real number is needed as output for the q-opinion of a user u on some item i, then the coefficient of the smallest ϵ power in the normalized polynomial $q_{u,i} /\!\!/ c_{u,i}$ can be considered. For example, if this polynomial is $3.5\epsilon + 4.5\epsilon^2$, then 3.5 is produced, whereas if the polynomial is $3\epsilon^2 + 2\epsilon^3$, then 3 is produced.

We note that in practice the polynomials are more useful as a whole because as such they enable a more fine grained comparision of recommendations for different items than just the approximation by their most important coefficient. Thus, we can substitute directly these polynomials for $u_k[i]$'s in formula (3.1) and then perform symbolic computations with polynomials generating in the end recommendations which are polynomials as opposed to single numbers. However, we use the above most-important coefficient approximation in order to be able to compare with other methods which only produce single numerical predictions for user/item ratings.

3.6 Evaluation

In this section we present the evaluation of our system. We have used the same dataset that Massa and Avesani [21] have collected for testing their trust-based recommendation methods (http://www.trustlet.org).

3.6.1 Dataset

The dataset has been crawled from the *Epinions.com* Web site. Epinions is a site where the users can write and read reviews about items such as movies, cars, books, etc. The users also assign numeric ratings to the items of interest. These ratings are in a scale from 1 (min) to 5 (max).

What is interesting about *Epinions.com* is that the users can create their web of trust which is *accessible*, and thus, the dataset we consider contains a trust network that we can use. To the best of our knowledge, this is the only dataset which provides both user ratings, as well as a trust network.

The dataset has the following specific characteristics:

1. There are

   (a) 45,819 users,

   (b) 139,738 items,

   (c) 664,823 ratings, and

   (d) 487,183 trust statements.

2. The number of customers who trust someone is 33,961.

3. The number of users who have rated at least one item is 40,163.

3.6.2 Evaluation Metrics

There are several types of metrics that have been used to evaluate recommendation systems. One metric we use is the well known Mean Absolute Error (MAE). This is the ubiquitous metric in most of the works in recommender system and the easiest to interpret for measuring the effectiveness of prediction. MAE measures the deviation of predictions from the true user-specified ratings. The technique for calculating MAE is *leave-one-out*, i.e. the true user rating is hidden and the prediction is calculated.

For each rating-prediction pair (r, p), the absolute error $|r - p|$ is calculated. The procedure is repeated for all the rating-prediction pairs and the average is finally calculated. Formally,

$$\text{MAE} \quad = \quad \frac{\sum_{(r,p)\in P} |r-p|}{|P|}, \tag{3.4}$$

where P is the set of rating-prediction pairs and $|P|$ is the cardinality of this set.

If we qualify a rating-prediction pair as $(r_{u,i}, p_{u,i})$ to express by the subscripts the particular user and item for the pair, and also introduce a 0/1 variable $a_{u,i}$ which is 1 when the system is able to predict a value for user u and item i, and 0 otherwise, then formula 3.4 is equivalently reexpressed as

$$\text{MAE} = \frac{\sum_{u \in U} \sum_{i \in I_u} |r_{u,i} - p_{u,i}|}{\sum_{u \in U} \sum_{i \in I_u} 1}, \tag{3.5}$$

where U is the set of all users, whereas I_u is the set of items for which user u has provided a rating and the system was able to produce a prediction. The denumerator of the fraction can also be expressed as $\sum_{u \in U} |I_u|$.

Clearly, the lower the MAE, the more accurate is the system.

Another metric which is argued to be very important in the study of recommender systems (cf. Herlocker et. al.) is the *Coverage*. The coverage is the fraction of ratings, which after being hidden, the system is able to produce a prediction. This metric is important because the recommender systems are not always able to give recommendations for a given user and a given item.

We also measure the *rank accuracy* (RA) of the considered methods. The RA is the ability to generate an ordering of recommendations that matches how the user would have ordered his/her opinions (ratings) [14]. As a rank accuracy metric we consider the Spearman ρ (cf. [14]), which is computed as the Pearson correlation, except that ratings and predictions are first transformed into ranks.

Since the dataset we consider is significantly large, as proposed by Massa and Avesani in [21], we also study in detail the following segments or views on the data:

1. *Heavy raters*, who provided more than 10 ratings;

2. *Opinionated users*, who provided more than 4 ratings and whose standard de-

viation of ratings is greater than 1.5;

3. *Black sheep users*, who provided more than 4 ratings and for which the average distance of their rating on item i with respect to mean rating of item i is greater than 1;

4. *Cold start users*, who provided from 1 to 4 ratings;

5. *Controversial items*, which received ratings whose standard deviation is greater than 1.5.

6. *Niche items*, which received less than 5 ratings;

Evidently, the performance numbers obtained on these segments of the data provide better insights regarding the strengths of various methods.

3.6.3 Experimental Results

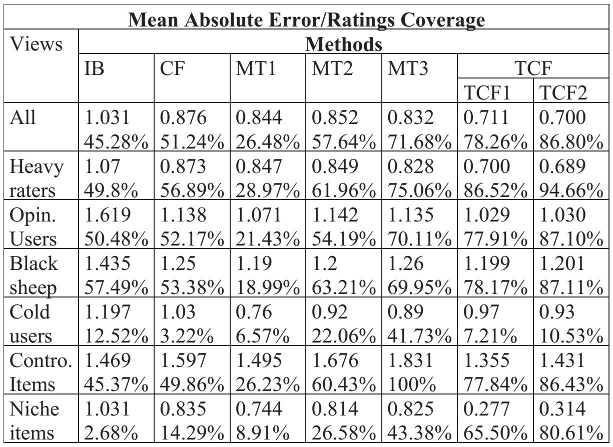
In this section we present our experimental results comparing our recommendation method, TCF[2], with the following methods.

1. *Mole Trust* (MT) introduced in [19]: the method is instantiated with *trust horizon* one, two, and three[3], and these instantiations are denoted by MT1, MT2, and MT3, respectively;

2. User-based collaborative filtering (*CF*) as descibed before, and

[2] **T**rust-based Enhanced **C**ollaborative **F**iltering

[3] The trust horizon is parameter of Mole Trust which specifies the depth of neighbors in the trust graph that are considered by the method.



Mean Absolute Error/Ratings Coverage							
Views	Methods						
	IB	CF	MT1	MT2	MT3	TCF	
						TCF1	TCF2
All	1.031 45.28%	0.876 51.24%	0.844 26.48%	0.852 57.64%	0.832 71.68%	0.711 78.26%	0.700 86.80%
Heavy raters	1.07 49.8%	0.873 56.89%	0.847 28.97%	0.849 61.96%	0.828 75.06%	0.700 86.52%	0.689 94.66%
Opin. Users	1.619 50.48%	1.138 52.17%	1.071 21.43%	1.142 54.19%	1.135 70.11%	1.029 77.91%	1.030 87.10%
Black sheep	1.435 57.49%	1.25 53.38%	1.19 18.99%	1.2 63.21%	1.26 69.95%	1.199 78.17%	1.201 87.11%
Cold users	1.197 12.52%	1.03 3.22%	0.76 6.57%	0.92 22.06%	0.89 41.73%	0.97 7.21%	0.93 10.53%
Contro. Items	1.469 45.37%	1.597 49.86%	1.495 26.23%	1.676 60.43%	1.831 100%	1.355 77.84%	1.431 86.43%
Niche items	1.031 2.68%	0.835 14.29%	0.744 8.91%	0.814 26.58%	0.825 43.38%	0.277 65.50%	0.314 80.61%

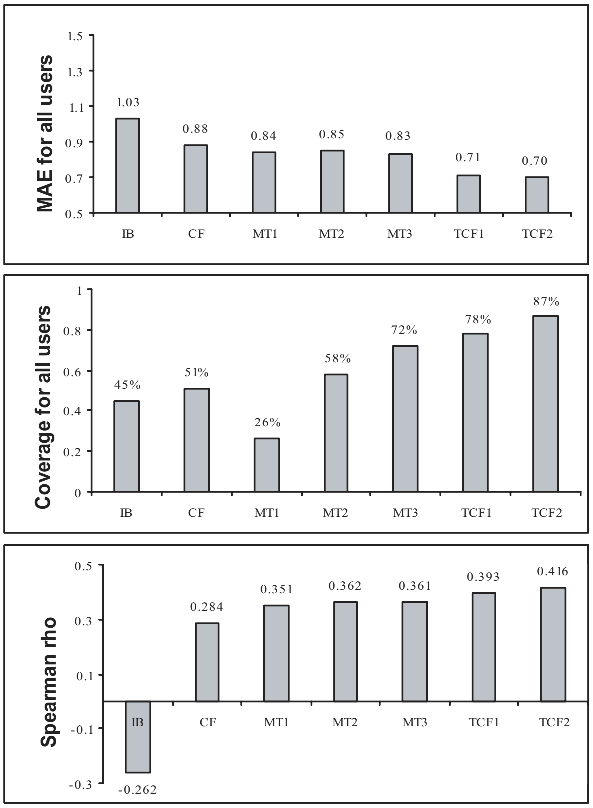


Figure 3.2: [Top] MAE and Coverage for the different methods. The performance numbers for our TCF method are given in the two rightmost columns of each table. [Bottom] Graphs for MAE, Coverage, and Spearman ρ when considering all users and items.

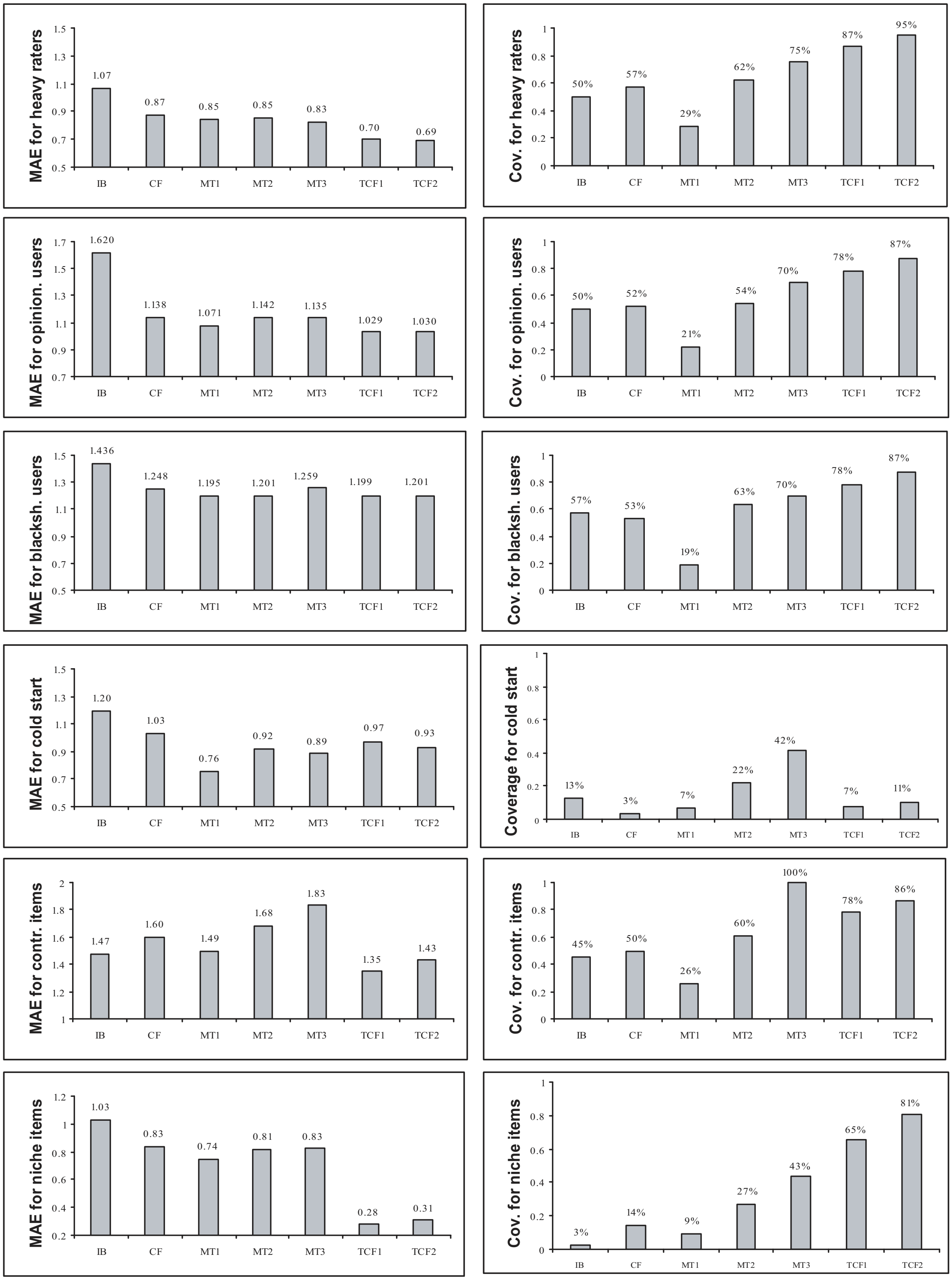


Figure 3.3: MAE and Coverage for the specified data segments.

3. Item-based (*IB*) collaborative filtering, which instead of similar users, finds similar items and then computes the recommendation as a weighted average of the ratings for the similar items (cf. [23]).

Specifically, we present results for our method when performing one and two iterations or passes through the data. We do not perform more iterations because the results reach a fixed point after the second iteration. The results are presented in figures 3.2 and 3.3.

In Figure 3.2 [Top] we show error and coverage numbers for the different methods we consider.

In Figure 3.2 [Bottom] we graph the performance numbers for the different methods when considering all the users and items. As it can be seen, our TCF method instantiations (TCF1 and TCF2) significantly outperform the other methods in terms accuracy, coverage, and Spearman ρ. Namely, our MAE values are 0.71 and 0.70 for TCF1 and TCF2, respectively, which not only are smaller than the MAE values for the other methods, but also are below the stated barrier of 0.73 in Herlocker et. al. paper [14]. The MAE values for our TCF method when performing two or three iterations are very close to each other. The coverage however is better for TCF when performing two iterations.

In Figure 3.3, we graph the performance numbers for the different methods when considering different views of users and items. Note that we consider these views when computing the error and coverage, not when computing the predictions. Our TCF method performs better for every view except for cold start users. MoleTrust with trust horizon 1 (MT1) gives a better accuracy by severely sacrificing the ratings coverage. As we go to a greater trust horizon for MoleTrust, and have MT2 and MT3, the coverage improves, but the accuracy decreases.

Interestingly, for cold start users, if we consider a naive recommender, which, for a given item i, always gives the simple average of ratings for i over all users, then the MAE is 0.86 and the coverage is 93%, thus being better than any other method for acceptable levels of coverage. Our conclusion with respect to cold start users is that one does not need to use elaborated methods, but just rely on the *most simple method* considering the aggregate opinion of all the users.

Another interesting feature of our method is that we can increase the coverage by performing more iterations without sacrificing accuracy. This in contrast to MoleTrust which suffers decreased accuracy (higher MAE and MAUE) as the trust horizon increases. In MoleTrust, the trust horizon represents a tradeoff between accuracy and coverage, but this is not so for our TCF method.

3.7 Conclusions

We have presented a novel method for utilizing trust-based networks for enhancing collaborative filtering in recommendation systems. Our main idea is to use infinitesimal polynomials for representing the trustworthiness of users when aggregating their opinions for producing recommendations. When aggregating recommendations from trusted neighbors, these polynomials enforce our belief that the opinions of immediate (distance-one) neighbors are infinitely more important than the opinions of distance-two neigbors which in turn are more important than the opinions of distance-three neighbors, and so on. This way of treating opinions is a departure from previous approaches which consider a less drastic aggregation of trust neighbors' opinions.

The results justify our proposed method. We are the first to report a MAE of 0.7 on a large, real life, Epinions dataset, for which the next best MAE values are above 0.8 (for reasonable coverage). We also break the MAE barrier of 0.73, previously reported in the literature. We achieve this without sacrificing our ratings prediction coverage which is also better than that of the other methods.

We believe that our proposed framework based on infinitesimal numbers and polynomials opens the way for expressing special qualitative preferences in recommender systems, and it is an important step towards harnessing the power of social networks.

Bibliography

[1] M. Chowdhury, A. Thomo, and W. W. Wadge, "Preferential infinitesimals for information retrieval," in *Artificial Intelligence Applications and Innovations*, pp. 113–125, 2009.

[2] M. Chowdhury, A. Thomo, and W. W. Wadge, "Enhanced information retrieval with preferential hyperreals," *Submitted to Engineering Intelligent Systems*, 2009.

[3] M. Chowdhury, A. Thomo, and W. W. Wadge, "Trust-based infinitesimals for enhanced collaborative filtering," in *the 15th International Conference on Management of Data (COMAD'09)*, 2009.

[4] B. Liu, "Web data mining: Exploring hyperlinks, contents and usage data.," *Springer*, vol. Berlin Heidelberg, 2007.

[5] C. D. Manning, P. Raghavan, and H. Schutze, "Introduction to information retrieval.," 2008.

[6] J. H. Keisler, *Foundations of Infinitesimal Calculus.* http://www.math.wisc.edu/∼keisler/foundations.html: On-line Edition,

2007.

[7] S. Malik, A. Trotman, M. Lalmas, and N. Fuhr, "Overview of inex 2006.," *Fifth Workshop of the INitiative for the Evaluation of XML Retrieval*, pp. 1–11, 2007.

[8] G. J. Bex, F. Neven, T. Schwentick, and K. Tuyls, "Inference of concise dtds from xml data.," pp. 115–126, 2006.

[9] A. Bruggemann-Klein and D. Wood., "One-unambiguous regular languages.," *Inf. Comput.*, vol. 140, no. 2, pp. 229–253, 1998.

[10] S. Robertson, "Understanding inverse document frequency: On theoretical arguments for idf.," *J. of Documentation*, vol. 60, pp. 503–520, 2004.

[11] C. E. Shannon, "A mathematical theory of communication.," *The Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[12] A. N. Aizawa, *An Information-Theoretic Perspective of TF-IDF measures.*, vol. 39. Inf. Process. Manage., 2003.

[13] "On-line internet shakespeare edition," *English Department, University of Victoria.* http://internetshakespeare.uvic.ca/index.html.

[14] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[15] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems," in *CoopIS/DOA/ODBASE (1)* (R. Meersman and Z. Tari, eds.), vol. 3290 of *Lecture Notes in Computer Science*, pp. 492–508, Springer, 2004.

[16] J. O'Donovan and B. Smyth, "Trust in recommender systems," in *IUI* (R. S. Amant, J. Riedl, and A. Jameson, eds.), pp. 167–174, ACM, 2005.

[17] A. J. Golbeck, *Computing and applying trust in web-based social networks.* PhD thesis, College Park, MD, USA, 2005. Chair-Hendler,, James.

[18] P. Avesani, P. Massa, and R. Tiella, "A trust-enhanced recommender system application: Moleskiing," in *SAC* (H. Haddad, L. M. Liebrock, A. Omicini, and R. L. Wainwright, eds.), pp. 1589–1593, ACM, 2005.

[19] P. Massa and P. Avesani, "Trust metrics on controversial users: Balancing between tyranny of the majority and echo chambers," *International Journal on Semantic Web and Information Systems*, vol. 3, no. 1, pp. 39–64, 2007.

[20] C.-N. Ziegler, *Towards Decentralized Recommender Systems.* Saarbrcken, Germany: Verlag Dr. Mller, May 2008.

[21] P. Massa and P. Avesani, "Trust-aware recommender systems," in *RecSys* (J. A. Konstan, J. Riedl, and B. Smyth, eds.), pp. 17–24, ACM, 2007.

[22] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," 1998.

[23] T. Segaran, *Programming Collective Intelligence*. O'Reilly, 2007.