Profile Diversity in Search and Recommendation

Maximilien Servajean, Esther Pacitti, Sihem Amer-Yahia, Pascal Neveu

Profile Diversity in Search and Recommendation*

Maximilien Servajean
INRIA & LIRMM
University of Montpellier
France
servajean@lirmm.fr

Esther Pacitti
INRIA & LIRMM
University of Montpellier
France
pacitti@lirmm.fr

Sihem Amer-Yahia
CNRS, LIG
sihem.amer-yahia@imag.fr

Pascal Neveu
INRA/SupAgro
Montpellier, France
pn@supagro.inra.fr

ABSTRACT

We investigate profile diversity, a novel idea in searching scientific documents. Combining keyword relevance with popularity in a scoring function has been the subject of different forms of social relevance [2, 6, 9]. Content diversity has been thoroughly studied in search and advertising [4, 11], database queries [16, 5, 8], and recommendations [17, 10, 18]. We believe our work is the first to investigate profile diversity to address the problem of returning highly popular but too-focused documents. We show how to adapt Fagin's threshold-based algorithms to return the most relevant and most popular documents that satisfy content and profile diversities and run preliminary experiments on two benchmarks to validate our scoring function.

Categories and Subject Descriptors

H.4 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Performance, Experimentation

Keywords

Recommendation, diversity, top-k

1. INTRODUCTION

Cross-discipline scientific domains have been growing thanks to the various calls for funding of different government agencies and to the adoption of collaborative tools. Several large projects now involve sizable laboratories of biologists, computer scientists, chemists and statisticians. In cross-discipline domains, users belonging to different communities produce various scientific material that they own, share, or endorse. In that context, we are interested in querying and recommending scientific material in the form of documents.

Table 1: Example of the need of cross-disciplinary researches

Undiversified Profiles		
Documents	Communities	Disciplines
Short-term responses of leaf growth rate to water defic...	Ecophysiologist community	Biologist discipline
Drought and Abscisic Acid Effects on Aquaporin Content...	Ecophysiologist community	Biologist discipline
Control of leaf growth by abscisic acid: hydraulic or non-hydraulic processes...	Ecophysiologist community	Biologist discipline
The importance of the anthesis-silking interval in breeding for drought tolerance in tropical maize...	Ecophysiologist community	Biologist discipline
Diversified Profiles		
Short-term responses of leaf growth rate to water defic...	Ecophysiologist community	Biologist discipline
A Multiscale Model of Plant Topological Structures...	Modeling community	Computer scientists discipline
Drought and Abscisic Acid Effects on Aquaporin Content...	Ecophysiologist community	Biologist discipline
Computational analysis of flowering in pea (Pisum sativum)...	Modeling community	Computer scientists discipline

Such documents cover various topics such as models for plant phenotyping, statistics on specific kinds of plants, or biological experiments. In this paper, we investigate diversity when searching scientific documents.

The ability to search scientific documents helps scientists gather and share knowledge on the same topic that is endorsed by other scientists. Each user belongs to a well-known discipline (e.g. computer science, biology, mathematics, etc.). Within a discipline a user belongs to one or more communities which reflect specializations of a discipline. For instance in the biology discipline, examples of communities are geneticists, ecophysiologists and plant breeders. The profile of a user is therefore a combination of her discipline and communities. In such a context, searching documents requires the careful design of an appropriate relevance function. We consider the example of plant phenotyping research where various disciplines and communities are involved. When an ecophysiologist u submits a query q ="plant model" (similar to q="model" as everyone works in the plant area), u might want documents containing details

*Work conducted within the Institut de Biologie Computationnelle and partially funded by the labex NUMEV and the CNRS project Mastodons.

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

on experiments by other ecophysiologists or those describing models of plant behavior shared by computer scientists. Table 1 shows two possible result lists. The list at the top is based on finding documents relevant to q that have diverse content. As we can see, that list only contains documents owned or shared by ecophysiologists. Since u is also interested in computer models, the list of results in the bottom part of the table would be more appropriate since it returns documents endorsed by users having different profiles.

Traditionally, diversity is achieved along one axis that is content. Content diversity alleviates the risk of returning highly-relevant but too-similar documents. In this work, we advocate *profile diversity* to address the problem of returning highly popular but too-focused documents. We design a scoring function that combines query relevance, content diversity to alleviate document similarity in query results, document popularity to account for profile endorsements, and finally, profile diversity to expose users to documents owned and shared by different communities. Combining keyword relevance with popularity in a scoring function has been the subject of different forms of social relevance [2, 6, 8]. Content diversity has been thoroughly studied in search and advertising [4, 11], database queries [16, 5, 8], and recommendations [17, 10, 19]. We believe our work is the first to investigate profile diversity in searching scientific documents.

In summary, we make the following contributions.

1. We introduce profile diversity for scientific document search as a complement to traditional content diversity. Profile diversity combines the discipline and communities to which a user belongs.

2. We propose an adaptation of Fagin's threshold-based algorithms to return the most relevant and most popular documents that satisfy content and profile diversities.

3. To validate our scoring function, we ran experiments that use two benchmarks: a realistic benchmark with scientists and TREC'09.

This paper is organized in the following way. Section 2 provides some background on document search and recommendation in the context of on-line scientific communities and presents the problem definition. Section 3 describes our general scoring function, $DivRSci$, based on probabilistic diversification. Next, Section 4 presents all algorithms necessary for $DivRSci$, and shows in details our contributions for profile diversification. In Section 5, we present the performance evaluation behavior of $DivRSci$ compared to other approaches, using in two benchmarks. Section 6 is concerned with the related work, and finally Section 7 concludes and provides directions for future work.

2. BACKGROUND

We focus on online scientific communities where users aim to query and have recommendations of inter-community and inter-disciplinary documents shared by other scientists. Our approach is generic, however to facilitate the understanding of our concepts and model we take into account plant phenotyping research that clearly requires inter-community and inter-disciplinary research.

For scientific document recommendation, it is essential to understand the sense of inter-community and inter-disciplinary research. In general, a user belongs to a well known discipline (*e.g.* computer science, biology, mathematics, etc.). Within a discipline a user belongs to one or more communities which reflects specializations of a discipline. For instance in the biology discipline, examples of communities are geneticists, ecophysiologists and plant breeders. Inter-community research refers to the fact that users research interests involves different communities of one discipline. For instance a geneticist may be interested in specific research results of the ecophysiologists community to understand the genetic behavior of some plants. Inter-disciplinary research refers to the fact that users research interests involves different disciplines. For instance, a biologist can query for mathematical tools that can model a plant behavior. In both inter-community and inter-disciplinary research, users would benefit from discovering new and diversified research trends coming from different communities or disciplines.

In our context, we choose a *content based* join to a *collaborative filtering* recommendation approach where users profiles - or alternatively user research interests - are defined based on the documents DS_i the user u_i stores. Thus, we assume a set of users $U = \{u_1, ..., u_n\}$. Each user u_i shares some of his documents $D_i = \{d_1, ..., d_m\}$ (or contents) with his friends, such that D_i is a subset of his DS_i. A document d can be shared by 1 to n users. Each time a document is chosen to be shared and copied, a new replica (or copy) of d is produced. In our context, a replica refers to the fact that different users have the same instance of a document in their work-space. Thus, each document d is associated with a *degree of replication* that expresses the number of replicas of d among U. Notice that the degree of replication can be related to the document popularity.

Documents are represented based on the vector space model [14]. By using $tf-idf$ a document is represented by a list of keywords $k_1, ..., k_z$, and the vector represents the weight of each distinct keyword given the document and the whole corpus. A user *profile* $profile_i$ expresses his interests based on DS_i. Queries are expressed by a list of keywords $k_1, ..., k_z$. Users' profiles and queries are also represented based on the vector model.

Problem Statement: Given U, DS, D and a keyword query q submitted by some user u the problem we address is to propose a new scoring function to recommend the $top-k$ most relevant documents among D to favor the inter-community, inter-disciplinary research and diversity requirements presented above. We assume that the k documents are in a sorted order list L in descending relevance order.

The intuition of our approach is that guarantees of inter-community and inter-disciplinary recommendation can be achieved by diversifying the documents and related users profiles in L. Therefore to produce L we identify four recommendation requirements with respect to the relevancy of a document d_i:

1. The similarities of d_i and q.

2. Content Diversification with respect to the documents already chosen in L.

3. The popularity of d_i.

4. Profile diversification with respect to the profiles of the users that owns the documents already chosen in L. Those profiles should be either similar to u (for

inter-community recommendation) or similar to q (for inter-disciplinary recommendation).

3. SCORING MODEL

Several methods have been proposed for diversification [18, 17, 5, 7, 1]. However, they only address requirements 2 discussed in the previous section. Our goal is to introduce profile diversification (*i.e.* requirement 4), taking into account a probabilistic diversification model because it provides more guarantees for inter-disciplinary and inter-community recommendation, as we show in our experiments in section 5.

3.1 Probabilistic Diversification

In the domain of information retrieval, given D and a query q, the computation of the top-k diversified documents is known to be NP-hard problem. Following [7, 5], $div(d_i|\{d_1, ..., d_{i-1}\})$ is defined as the diversification probability of d_i (*i.e.* brings novelty to the user u) with respect to the previously chosen documents in L (i.e. $\{d_1, ..., d_{i-1}\}$). In this model, the diversity can be expressed using the notion of redundancy. The redundancy $red_c(d_i, d_j)$ is computed by comparing the similarity between d_i and d_j. Angel and Koudas [5] strictly defines the diversity probability as $1 - red(d_i|d_1, ..., d_{i-1})$. Based on the hypothesis that the redundancy between documents d_i and d_j is independent of its redundancy with the other documents [12, 5, 7], the probabilistic diversification score is defined as:

$$1 - red(d_i|d_1, ..., d_{i-1}) = \prod_{d_j \in \{d_1, ..., d_{i-1}\}} 1 - red(d_i, d_j) \quad (1)$$

3.2 DivRSci Scoring Function

To address the 4 requirements presented in section 2, we propose the *DivRSci* score that evaluates the relevancy of a document given a query q:

$$score_{DivRSci}(d, u, q) = rel(d, q). div_c(d|\{d_1, ..., d_{i-1}\}).div_p(u_d|\{u_{d_1}, ..., u_{d_{i-1}}\}) \quad (2)$$

$rel(d, q)$ defines the probability that d will answer the query q. It can be defined as the similarity measure between d and q (*e.g.* cosine, jaccard, etc.)[15]. This addresses requirements 1.

$div_c(d|\{d_1, ..., d_{i-1}\})$ is a straightforward application of equation 1 and addresses requirement 2.

$div_p(u_d|\{u_{d_1}, ..., u_{d_{i-1}}\})$ is the profile diversification score of document d and takes into account the document's popularity (requirement 3) and the diversification among trusted users (requirement 4). More precisely, we evaluate for each user in U holding a replica of d, a trust and a diversification score (requirement 4) with respect to L.

The trust $trust(v_n, u, q)$ is a value which indicates the confidence the user u can have in the user v. Such information can be computed in many ways (*e.g.* social friendship, localization, previous recommendation, etc.). In the following we consider that the trust takes into account the relevance of the user v, given u and q. The relevance indicates if v is either similar to u (*i.e.* inter-community recommendation) or to q. (*i.e.* inter-disciplinary recommendation). We define the user's relevance in equation 3.

$$rel_{trust}(v, u, q) = \alpha * sim(u, v) + (1 - \alpha) * sim(v, q) \quad (3)$$

More formally, we propose the user profile diversification score defined in Equation 4. Recall that the profile diversification score also takes into account the popularity of the document d_i (requirement 3), that is why we need $\frac{1}{N}$. Notice that $\frac{1}{N}$ is also used for normalization.

$$div_p(u_d|\{u_{d_1}, ..., u_{d_{i-1}}\}) = \frac{1}{N}. \sum_{v_n \in u_{d_i}} \Bigg[rel_{trust}(v, u, q). \prod_{v_m \in \{u_{d_1}, ..., u_{d_{i-1}}\}} (1 - red_p(v_m|v_n) \Bigg] \quad (4)$$

4. ALGORITHMS

In this section we present in details the algorithms involved in *DivRSci*. For sake of clarity, in section 4.1, we present the extended version of the algorithm related to the probabilistic model we adopt [5] adapted for *DivRSci*. In section 4.2, we show the performance degradation brought by the profile diversification aspect of *DivRSci* and we propose a new threshold condition that is best suited to profile diversification. Finally in section 4.3 we propose a new algorithm to compute profile diversification.

4.1 Preliminaries

In [5], the authors propose an algorithm (called DAS) used to implement the following scoring function:

$$rel(d, q).(1 - red(d_i|d_1, ..., d_{i-1})) \quad (5)$$

DAS is a threshold based algorithm. Given a query q and a set of documents D, a threshold algorithm operates over a set of inverted indexes:

$$\begin{aligned} &w_i \Rightarrow < d_a, sc_a >, < d_b, sc_b >, ..., < d_n, sc_n > \\ &... \\ &w_m \Rightarrow < d_e, sc_e >, ..., < d_n, sc_n > \end{aligned} \quad (6)$$

where w_i is a word, d_a a document and sc_a the score of the document with respect to the word w_1 (*i.e.* $sc_a = sim(w_1, d_a)$). The documents are sorted in decreasing order of sc. Notice that the set of indexes used by the threshold algorithm depends on the query q. For instance, if $q = \{w_i, w_m\}$ then the inverted indexes will be the ones of w_i and w_m. Finally the algorithm stops when the threshold condition δ is satisfied. δ is computed based on the inverted indexes:

$$\delta = f(s_1, s_2, ..., s_n) \quad (7)$$

where f defines a specific measure (*e.g.* cosine, etc.) and s_i is the last sorted access on the w_i index. For instance, given a set of inverted indexes $\{w_i, w_j\}$, if we want to retrieve the top-1 document. The stop condition will be satisfied if the score of a document d is superior or equal to $\delta = f(s_i, s_j)$.

The goal of *DivRSci* is to find an optimal list L of k documents such that we can't find a better list L given u and q and our scoring function. That is, given L and a document $d_i \in L$, where $i \in \{1, ..., k\}$, we can't find any document $d_j \notin \{d_1, ..., d_{i-1}, d_i\}$ that would have a better score than d_i at the i^{th} place in L.

We propose *DAS_DivRSci* as an implementation solution (see Algorithm 1) that uses a new threshold condition suited for profile diversification. Notice that div_p (line 4), δ' (line 5) and line 9 are specific features related to *DivRSci*.

Algorithm 1: DAS_*DivRSci*

Input: index,query,user,k
Output: the top-k most relevant documents wrt. to our scoring function.

1 $L \Leftarrow$;
2 **while** $size(L) < min(k, size(corpus))$ **do**
3   $d \Leftarrow index.nextSortedAccess()$;
4   $d.score = rel(d,q).div_c(d|\{d_1,...,d_{i-1}\}).div_p(u_d|\{u_{d_1},...,u_{d_{i-1}}\})$;
5   **add** d **to** candidates;
6   **if** *the best candidate's score is higher than* δ' **then**
7   **add** best candidate **to** L;
8   **Update** the score of the other candidates;
9   **Update** $\prod_{d_j \in \{d_i,...,d_i-1\}} max\_div_c(d_j)$ **and** $prod_{d_j \in \{d_i,...,d_i-1\}} max\_div_p(d_j)$ using the best candidate;

The algorithm runs until L reaches k documents (line 2). From line 3 to 5, the algorithm performs a sorted access to get the next document, then it computes its score (*i.e.* $score_{DivRSci}$, formula 2) and inserts it into a candidates' list. The *candidates list* contains each document that has already been analyzed but that can't be inserted in L yet because the algorithm can still find documents with better diversity score. Notice that a document's score is not fixed until it has been added to L. At line 6, *DivRSci* analyses if the best candidates has a score higher than the threshold δ'. In other words, it analyses if there isn't any better document in the indexes. In that case, *DivRSci* inserts the best document in L and update the score of the other candidates (line 7 & 8). Line 9 will be explained in more details in the next subsection motivated by the new threshold score proposal.

4.2 DivRSci Threshold

As presented in formula 7, the threshold δ is evaluated using the document's score in the indexes $\{w_1, ..., w_n\}$. In *DivRSci*, div_c and div_p are always smaller than 1. Notice that while the number of documents in L grows, the content diversification score and the profile diversification score become to get smaller for any given document $d_i \notin L$. For instance, to retrieve 3 diversified documents (using our benchmark, $U = 50$ users, $D = 300$ documents), *DivRSci* needs about 175 sorted accesses in average. In the worst case, the whole index is used to find these 3 documents. Thus, δ is no longer appropriate.

We propose to use a new threshold δ' with respect to our scoring function to optimize the number of sorted accesses:

$$\delta' = f(s_1, s_2, ..., s_n).f_{div_c}(d_i, \{s_1, s_2, ..., s_n\}).f_{div_p}(d_i, \{s_1, s_2, ..., s_n\}) \quad (8)$$

Where each part of the threshold corresponds to a part of our scoring function (i.e. *DivRSci*). Notice that to compute f_{div_c} and f_{div_p} we need additional information because the indexes $\{s_1, ..., s_n\}$ are not sufficient. Thus, we define 4 primitives:

1. max_div_c: returns the maximum content diversity score between d_i and the documents that follow d_i in $\{s_1, s_2, ..., s_n\}$.
2. max_div_p: returns the maximum profile diversity score between d_i and the documents that follow d_i in $\{s_1, s_2, ..., s_n\}$.
3. max_trust: returns the maximum trust score of the users that share the document in $\{s_1, s_2, ..., s_n\}$. Notice that the part of the trust score that depends on the user u that submitted the query is evaluated as equal to 1.
4. max_rep: returns the maximum number of replicas of any documents in $\{s_1, s_2, ..., s_n\}$.

We now define f_{div_c} and f_{div_p}:

$$f_{div_c}(d_i, \{s_1, s_2, ..., s_n\}) = \prod_{d_j \in \{d_i,...,d_i-1\}} max\_div_c(d_j) \quad (9)$$

$$f_{div_p}(d_i, \{s_1, s_2, ..., s_n\}) = \frac{max\_rep}{N}.max\_trust.\prod_{d_j \in \{d_i,...,d_i-1\}} max\_div_p(d_j) \quad (10)$$

Lemma 1. *The content diversity score of a given document d_i is inferior or equal to f_{div_c}*

Lemma 2. *The profile diversity score of a given document d_i is inferior or equal to f_{div_p}*

The demonstration is straightforward.

Notice that $\prod_{d_j \in \{d_i,...,d_i-1\}} max\_div_c(d_j)$ and $\prod_{d_j \in \{d_i,...,d_i-1\}} max\_div_p(d_j)$ can be updated at each iteration without recomputing the overall formulas 9 and 10. In Algorithm 1, (line 9) *DivRSci* updates their values with respect to the last document inserted in L.

We now present an example to compare δ with our new threshold. Due to lack of space and for simplicity, we simplify the *DivRSci* scoring function by removing the trust and the popularity related to div_p:

$$div_p = \sum_{v_n \in u_{d_i}} \left[\prod_{v_m \in \{u_{d_1},...,u_{d_{i-1}}\}} (1 - red_p(v_m|v_n)) \right] \quad (11)$$

Not surprisingly, removing $\frac{1}{N}$ and rel_{trust} from the *DivRSci* scoring function, enables the definition of a simpler threshold, δ'', that is quite simpler to compute compared to δ', but that keeps the same general behavior:

$$\delta'' = lastSA.\prod_{d \in L} max\_div_c(d).\prod_{d \in L} max\_div_p(d) \quad (12)$$

In more details, table 2 shows a running case in which *DivRSci* is built L using δ''. We show that the number of sorted accesses would have been largely superior if we've used δ. The input is a built index of documents based on a query. The first column *step* corresponds to a whole iteration in algorithm 1 (line 3 to 9). The second column *sorted accessed* indicates the sorted access done at the given step (line 3 of algorithm 1) on the index of the input. The columns *max div$_c$* and *max div$_p$* indicate that the document's we've just done the sorted access on (*e.g.* document

Table 2: **Example of the effect of the threshold on the number of sorted accesses.**

Step	Sorted Access	$rel(d,q)$	max div_c	max div_p	Final Score	δ	δ'	L	C
1	A	0.90	0.85	0.45	0.9	0.9	0.345	A	-
2	B	0.88	0.84	0.46	0.238	0.88	0.34	A	B
3	C	0.87	0.95	0.65	0.34	0.87	0.33	A,C	B
...									
n	Z	0.55			0.0023	0.55			

A for step 1) can't be more diverse than the value indicated, with respect to all other indexed documents still not accessed. L is the list of results and C the list of candidates. The columns δ and δ'' indicates the value of the thresholds at the given step.

On step 1, *DivRSci* performs a sorted access on A. As it's the first document, the diversification score is 1 and the final score of the document is $rel(d,q) = 0.9$.
On step 2, *DivRSci* performs a sorted access on B. The final score of B is 0.238 due to its diversification score with respect to A. Notice that δ'' (which is inferior to δ) has a value of 0.34 which is superior to B's score. It means that we may find a better document.
Then, on step 3, *DivRSci* performs a sorted access on document C. The final score of this document (with respect to A) is 0.34 which is superior or equal to δ''. We can assume that there will not be any better document in the index. Therefore C is inserted in L. Notice that δ is equal to 0.87, and *DivRSci* couldn't have inserted C in L at this step by using δ. Furthermore, we can see that at step n, δ is equal to 0.55 which is still superior to C's score and is not satisfying the stop condition. This confirm the fact that the proposal of div_p for *DivRSci* introduces important complexity and our new threshold approach provides important performance improvement.

4.3 DivRSci Profile Diversification

In this section, we present how we compute div_p (Algorithm 1, line 4).

Algorithm 2 presents a possible way to compute div_p. From line 1 to 7, it computes for each user holding a replica of the document d a trust and a diversification score. On line 3, it evaluates the trust score of v_n with respect to u and to q. Then, from line 4 to 6 it evaluates the diversification score of v_n with respect to the users that hold a document already inserted in L.

Finally, on line 7 it combines the trust and the diversification score and adds the computed value to the global profile diversification score. Line 8 normalizes the value of div_p and takes into account the popularity of d_i.

Thus, the number of iterations is strictly equal to:

$$|U_{d_i}|.|U_{[d_1,\ldots,d_{i-1}]}|$$

and the complexity of the function, in the worst case is $O(n^2)$, where n is equal to the total number of users. Recall that the profile redundancy score between two documents also takes into account the trust score which depends on the u submitting the query. Therefore the profile diversification can't be precomputed because a specific index would be necessary for each user.

Algorithm 2: Profile Diversification Score Computing

Input: $List[d_1,...,d_{i-1}]$,User u,Query q,Document d_i
Output: The profile diversity score of d_i wrt. q, u and $[d_1,...,d_{i-1}]$

`/* the documents are indexed based on sim(d,q). */`

1 $profDiv \Leftarrow 0$;
2 **for** v_n **in** U_{d_i} **do**
3     $t \Leftarrow trust(u, v_n, q)$;
4     $div \Leftarrow 1$;
5   **for** v_m **in** $U_{[d_1,\ldots,d_{i-1}]}$ **do**
6         $div \Leftarrow div.red(v_n, v_m)$;
7     $profDiv \Leftarrow profDiv + t.div$;
8 $profDiv \Leftarrow \frac{profDiv}{N}$;

5. PERFORMANCE EVALUATION

In this section, we provide an experimental evaluation of *DivRSci* to assess the quality of recommendations, content diversification, profile diversification and of the algorithm efficiency. We have conducted a set of experiments using a self-built benchmark and using TREC'09. In section 5.1 we first describe the experimental setup. Then, in section 5.2, we discuss the results.

5.1 Experimental Setup

Our self-built benchmark is composed of a set of 50 users. They are scientists in the domain of plant phenotyping from different localities (e.g. Australia, England, France, etc.). They belong to 4 main disciplines (*i.e.* ecophysiologists, geneticist, mathematician, computer scientists). Each discipline contains about 4 communities. The users share documents related to their research with respect to different disciplines and communities. Our benchmark is composed of 300 documents, 92% of these documents have a degree of replication of 1, 3% of them have a degree of 2, 2% have a degree of 3 and 2% have a degree of replication of 4. All users submit queries that are 1/3 inter-disciplinary and 2/3 inter-community. They can be classified in two categories:

1. unspecific queries (*i.e.* queries with very few keywords such as "plant" or "plant model").

2. specific queries (*i.e.* queries with lot of keywords such as "FSPM structure function plant model").

Each category of query represents 50% of the total number of queries which is 300.

In addition to our self-built benchmark we also show that using a well known large-scale benchmark (*i.e.* TREC'09 in our case) produces comparable results. From TREC'09, we take 15000 documents and 1500 specific queries. 50%

of these queries are inter-disciplinary and 50% are inter-community. We consider 1000 users. We built the users profile by clustering the documents using k-means. Each cluster corresponds to a community. We obtained 30 communities. By considering that a discipline is a set of communities that are similar to each other, we expect to have 8 disciplines. In our scenario, the documents are replicated ranging from 1 to 200 copies.

In the following, we present the four scores we compared in our experiments:

1. *Simple top-k*: we only retrieve the documents that optimize $rel(d, q)$.

2. *DAS*: we retrieve the documents that optimize $rel(d_i, q).(1 - red(d_i|d_1, ..., d_{i-1}))$.

3. *Trusted DAS*: we retrieve the documents that optimize *DAS* score and that are shared by the most trusted users - with respect to the trust we defined in section 2.

4. *DivRSci*: we retrieve the documents that optimize our scoring function.

To understand the behavior of the scores, we analyze the following metrics:

1. The content diversity: $\sum_{d_i \in L} \sum_{d_j \in L} 1 - red(d_i, d_j)$

2. The profile diversity: $\sum_{u_i \in U_L} \sum_{u_j \in U_L} 1 - red(u_i, u_j)$

3. The average relevance of the documents in L:
$$avg_{d_i \in L}(sim(d_i, q))$$

4. The average relevance of the users involved in L:
$$avg_{u_i \in U_L}(\alpha.sim(u, u_i) + (1 - \alpha).sim(u_i, q))$$

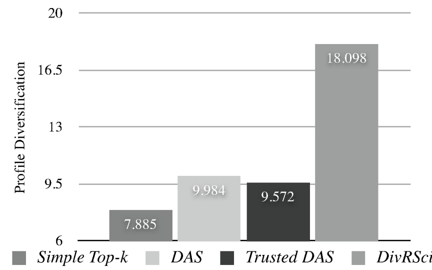
5. The cost to retrieve documents in number of sorted accesses by comparing several scores:

5.2 Experiments

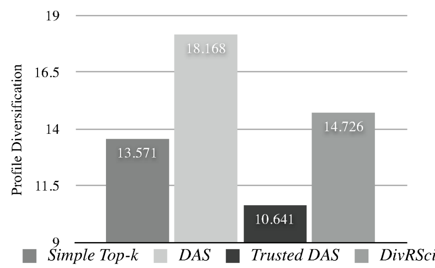
5.2.1 Scoring Function

Figure 1 compares the behavior of our scores to understand the degree of diversification of the chosen users profiles in L. In Figure 1a we executed unspecific queries. In Figure 1b we executed specific queries.

We discuss and analyze the expected profile diversification behavior with respect to our inter-disciplinary and inter-community requirements. Notice that given an unspecific query q_1="plant model", most users in U should be able to answer it because in some way they are all involved in plant research. Notice that unspecific queries enable inter-disciplinary recommendation, and by diversifying users profiles, more disciplines will be involved in the recommendation results (*i.e.* L) and the profile diversification measure should be high. In the case of specific queries such as q_2 ="FSPM structure function plant model", less users will be able to answer it because less users are involved in these researches as it is a subset of plant model researches. Notice that specific queries enable inter-communities recommendation, and by diversifying users profiles more communities of the same discipline will be involved in the recommendation



(a) the users submit unspecific queries.



(b) the users submit specific queries.

Figure 1: profile diversification depending on the top-k algorithm.

results (*i.e.* L) and the profile diversification measure should be low.

Not surprisingly Figures 1a and 1b show that the *simple top-k* and *DAS* have exactly the opposite behavior compared to the expected one. Their profile diversification measure double from 9.5 to 18 and 7 to 14 respectively (Figure 1a and Figure 1b) instead of decreasing. Moreover, we can see that by adding the trust score to *DAS* (*i.e. trusted DAS*), we resolved this issue by only inserting in L trusted users. Notice that, the trust score reduces considerably the profile diversification degree of *trusted DAS*. In *DivRSci*, we introduced a profile diversification score and a trust score. Therefore, *DivRSci* is able to compute a diversified list of users in L that has a coherent behavior with respect to the expected one. In Figure 2, we analyze if the behavior of

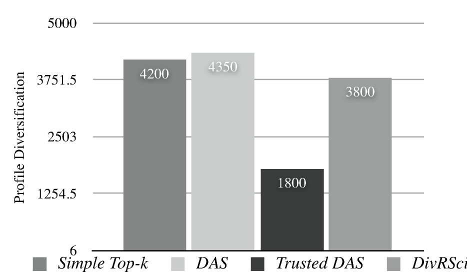
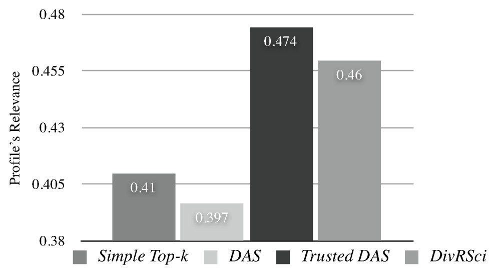
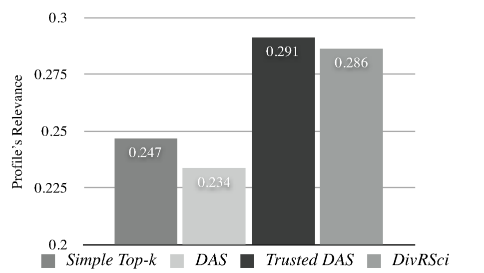


Figure 2: profile diversification with specific queries in TREC depending on the top-k algorithm.

the four scores is similar in the TREC'09 based benchmark. We only present profile diversification results due to a lack of space. All users submit only specific queries and we measure the profile diversification. As we can see, the different scores follow the same trend as the one of Figure 1b. However, the profile diversification is much higher due to the fact that with TREC'09, the number of replicas is much higher than in our self-built benchmark. This result shows that as the degree of replication globally increases the degree of diversification also increases. The goal of Figure 3 is to check if



(a) the users submit unspecific queries.



(b) the users submit precise and specific queries.

Figure 3: Average relevance of the users in L depending on the top-k algorithm.

the "profiles" in L are relevant given our recommendation requirement 4 (*i.e.* given u and q). As shown in Figure 3a and Figure 3b, since *simple top-k* and *DAS* does not have a trust score, this yields to a worse profile relevance. In the other hand, *DivRSci* profile diversification score is a compromise between the trust and the profile diversification of the users in L. Therefore, *DivRSci* is expected to have a relevance inferior to a scoring function that does not diversify the users such as *trusted DAS*. For instance, if $U = \{u_1, u_2, u_3\}$ where $rel(u_1) = rel(u_2) = 10$ and $rel(u_3) = 9$, *trusted DAS* will keep u_1 and u_2 in L. But if u_1 and u_2 have exactly the same profiles then, *DivRSci* will remove one of them and put u_3 instead. Notice, however, that *DivRSci* still have very good profile relevance results.

Finally, we constructed a feedback method using [13] to evaluate the list L quality taking in account *simple top-k*, *DAS*, *Trusted DAS* and *DivRSci*. The feedback was generally positive with more than 70% of satisfaction. The principal favored argument was the possibility to retrieve relevant inter-community and inter-disciplinary documents.

5.2.2 Threshold Efficiency

In this experiment, we show the effect of a complex scoring function and of the threshold on the number of sorted accesses. Table 3 resumes the experiment of running *DAS* and *DivRSci* with the threshold δ and δ' on our self-built benchmark. We first executed *DAS* with the threshold δ. *DAS* only diversifies the document's content. Obviously, it has the best results in term of sorted accesses. In second, we executed *DivRSci* with the threshold δ. Not surprisingly, the number of sorted accesses is very high because δ is not

Table 3: Number of sorted accesses depending on the scoring function and on the threshold to compute the top-3 documents.

$Score_{threshold}$	number of sorted accesses
DAS_δ	10
$DivRSci_\delta$	175
$DivRSci_{\delta'}$	30

suitable as discussed in section 4.2. Finally we executed *DivRSci* with the threshold δ'. The results are 6 times better than *DivRSci* with δ.

6. RELATED WORK

Content diversity has been studied in Web search, database queries, and recommendations. Diversifying Web search results and recommendations aims to achieve a compromise between relevance and result heterogeneity. In [12], the authors adopt an axiomatic approach to diversity that aims to address user intent. They show that no diversification function can satisfy all axioms together and illustrate that with concrete examples. In [4], taxonomies are used to sample search results in order to reduce homogeneity. In the database context [16, 8], solutions have proposed to post-process structured query results, organizing them in a decision tree [8] for easier navigation or merging ranked lists [16] for faster processing. In [3], a hierarchical notion of diversity in databases is introduced, and efficient top-k processing algorithms are developed. In recommendations [19, 10, 17], results are typically post-processed using pair-wise item similarity in order to generate a list that achieves a balance between accuracy and diversity. For example, in the recommender systems world, the approach in [19] defines an intra-list similarity which relies on mapping items to taxonomies to determine topics or using item features such as author and genre. The method is based on an exhaustive post-processing algorithm which operates on a top-N list to compute the top-K results (N > K). In contrast, in [10], diversity is formulated as a set-coverage problem. Finally, [11] introduces diversity in the framework of sponsored search ads, proposing algorithms for the selection of ads that intend to increase heterogeneity while not significantly reducing revenue and maintaining an incentive for advertisers to keep their bids as high as possible. Heterogeneity is aimed at as a notion that spans various occurrences of the same query, and not just a single one.

Notice that none of the above contributions tackles the problem of profile diversity as we do.

7. CONCLUSION

In this paper, we introduced profile diversity to ease inter-community and inter-disciplinary search and recommendation.
We proposed a scoring function (called *DivRSci*) that accounts for query relevance, content diversity to alleviate document similarity in query results, document popularity to account for community endorsements, and finally, discipline and community diversity to expose users to documents owned and shared by different disciplines and communities.

We argued that profile diversity provides good guarantees for inter-community and inter-disciplinary search and recommendation. Profile diversification is done by recommending documents that are shared by trusted and diversified users among all users. Our scoring function is based on a probabilistic model since it provides good guarantees of diversification. We presented in details all involved algorithms and we proposed a new threshold for *DivRSci* suited for profile diversification.

Through experimental evaluation using two benchmarks and comparing *DivRSci* with other scoring functions, we showed that *DivRSci* presents the best compromise between all requirements we have identified. Besides *DivRSci* also shows to be the best generating list of inter-disciplinary and inter-community documents. Finally, we presented the very good gains (factor of 6) of the new proposed threshold, suited for profile diversification.

In future work, we plan to propose a distributed approach for *DivRSci*.

8. ACKNOWLEDGMENTS

The authors would like to thank Romain Chapuis for his effort to build our benchmark and organizing feedback seances.

9. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM '09*, pages 5–14, 2009.

[2] S. Amer-Yahia and M. Benedikt. Efficient network aware search in collaborative tagging sites. *VLDB Endowment '08*, 1(1):710–721, 2008.

[3] S. Amer-Yahia and J. Shanmugasundaram. Efficient Online Computation of Diverse Query Results. *US Patent*, 2011.

[4] A. Anagnostopoulos, A. Z. Broder, and D. Carmel. Sampling Search-Engine Results. In *WWW '05*, pages 245–256, 2005.

[5] A. Angel and N. Koudas. Efficient diversity-aware search. In *SIGMOD '11*, pages 781–792, 2011.

[6] X. Bai, R. Guerraoui, A. Kermarrec, and V. Leroy. Collaborative personalized top-k processing. *TODS*, 36(26), 2011.

[7] H. Chen and D. R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *SIGIR '06*, pages 429 – 436, 2006.

[8] Z. Chen and T. Li. Addressing diverse user preferences in sql-query-result navigation. In *SIGMOD '07*, pages 641–652, 2007.

[9] F. Draidi, E. Pacitti, D. Parigot, and G. Verger. P2Prec: a social-based P2P recommendation system. In *CIKM '11*, pages 2593–2596, 2011.

[10] K. El-Arini, G. Veda, D. Shahaf, and C. Guestrin. Turning down the noise in the blogosphere. In *KDD '09*, pages 289–298. ACM Press, 2009.

[11] E. Feuerstein, P. A. Heiber, J. Martìnez-Viademonte, and R. Baeza-Yates. New Stochastic Algorithms for Scheduling Ads in Sponsored Search. In *LA-WEB '07*, pages 22–31, 2007.

[12] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW '09*, pages 381–390, New York, New York, USA, 2009. ACM Press.

[13] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *TOIS*, 20(4):422–446, 2002.

[14] C. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. 2008.

[15] G. Salton, A. Wong, and C. Yang. A Vector Space Model for Automatic Indexing. *CACM*, 18(11), 1975.

[16] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. Amer-Yahia. Efficient Computation of Diverse Query Results. In *ICDE '08*, pages 228–236. Ieee, Apr. 2008.

[17] C. Yu, L. Lakshmanan, and S. Amer-Yahia. It takes variety to make a world: diversification in recommender systems. In *EDBT '09*, pages 710–721, 2009.

[18] X. Zhu, A. B. Goldberg, J. Van, and G. D. Andrzejewski. Improving Diversity in Ranking using Absorbing Random Walks. In *HLT-NAACL '05*, 2005.

[19] C.-N. Ziegler, S. M. McNee, J. a. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW '05*, pages 22–32, 2005.