Preferential Diversity

Xiaoyu Ge
University of Pittsburgh
Pittsburgh, U.S.A
xiaoyu@cs.pitt.edu

Panos K. Chrysanthis
University of Pittsburgh
Pittsburgh, U.S.A
panos@cs.pitt.edu

Alexandros Labrinidis
University of Pittsburgh
Pittsburgh, U.S.A
labrinid@cs.pitt.edu

ABSTRACT

The ever increasing supply of data is bringing a renewed attention to query personalization. Query personalization is a technique that utilizes user preferences with the goal of providing relevant results to the users. Along with preferences, diversity is another important aspect of query personalization especially useful during data exploration. The goal of result diversification is to reduce the amount of redundant information included in the results. Most previous approaches of result diversification focus solely on generating the most diverse results, which do not take user preferences into account. In this paper, we propose a novel framework called *Preferential Diversity* (PrefDiv) that aims to support both relevancy and diversity of user query results. PrefDiv utilizes user preference models that return ranked results and reduces the redundancy of results in an efficient and flexible way. PrefDiv maintains the balance between relevancy and diversity of the query results by providing users with the ability to control the trade-off between the two. We describe an implementation of PrefDiv on top of the HYPRE preference model, which allows users to specify both qualitative and quantitative preferences and unifies them using the concept of preference intensities. We experimentally evaluate its performance by comparing with state-of-the-art diversification techniques; our results indicate that PrefDiv achieves significantly better balance between diversity and relevance.

1. INTRODUCTION

Motivation As the amount of data being generated every day increases exponentially, the term "Big Data" used to represent the challenge of large-scale data processing, is being mentioned more and more frequently in everyday life [11]. This reflects the fact that people are increasingly relying on using data as an integral part of their daily activities (e.g., decisions and collaborations).

The challenge of scalable data processing can be viewed from two viewpoints. Traditionally, scalability has been viewed from a systems point of view, where challenges can be attributed to an increasing rate of data on the one hand, and network bandwidth, processing power, and storage limitation on the other hand. Scalability can also be viewed from a human point view [13]. Given the volumes of data, the challenge here is how to avoid overwhelming the users with irrelevant results.

ExploreDB'15 May 31-June 04 2015, Melbourne, VIC, Australia

ACM 978-1-4503-3740-3/15/05 ...$15.00.
http://dx.doi.org/10.1145/2795218.2795224

Query personalization is a well-know technique in dealing with the scalability challenges from a human point of view. Query personalization often happen at two different levels:

- *Ranking* (local property) – Ranking techniques utilize user *preferences* with the aim of providing the most relevant results to the users [14]. These techniques can be distinguished as *quantitative-based*, *qualitative-based*, or *hybrid,* based on the type of user preferences that they can support.
- *Diversification* (global property) – Since highly ranked objects could be similar to each other, diversification techniques aim to reduce the amount of redundant information in the results. These techniques typically group data in sets that are most "dissimilar" with each other (e.g., [2, 7]).

Diversity has various definitions in the literature [5]. The most common definitions are based on *similarity*, where diversity means to include in the results objects that are dissimilar to each other (e.g. [17]). Other definitions are based on either *semantic coverage*, where diversity means to include objects that belong to different categories (e.g. [1]), or *novelty*, where diversity means to include data that contains new information (i.e., information that has not been presented previously) (e.g. [4]). During the past, many result diversification models have been proposed, e.g. *MaxMin* and *MaxSum* (e.g. [3, 10, 16]) and *DisC Diversity* [6]. Most existing approaches to data diversification follow a top-k approach for selecting diverse results, by assuming a fixed number k denoting the number of diverse results to be retrieved and assigning some diversity score to each selected result.

Even though the goal of diversity is to ensure potentially important data is not lost due to its low ranking, however the result of diversification does not automatically imply relevancy for the users. That is, diversity cannot ensure relevancy as in the case of ranking and to the best of our knowledge, there is no diversity technique that utilizes user preferences to ensure data relevancy similar to rankings. Such a technique could address the problem of diverse results which are at the same time relevant to the users.

In this paper, we present *Preferential Diversity* (PrefDiv), a new personalization technique that combines ranking and diversification, effectively addressing diversity while maintaining the relevance of the results according to user's preferences. PrefDiv follows a top-k approach, utilizing models that return ranked results. PrefDiv provides certain guarantees on the dissimilarity and cover-

age of the results, similar to DiSC Diversity [6]. At a high level, PrefDiv starts by selecting objects above a given *score* or *intensity value*, i.e., the most relevant results according to a user's preferences, and returns k diverse objects so that these objects exhibit a degree of dissimilarity under different dimensions and user-controlled distribution of intensity values.

Contributions This paper's contributions are as follows:

- We introduce a new framework called *PrefDiv*, which is capable of generating results that are not only *relevant* to users' preferences but are also *diverse*. Our framework provides users with a fine control over the trade-off between relevancy and diversity through intuitive tunable parameters.
- We design and implement a prototype of a real system for PrefDiv and design algorithms to work with the HYPRE hybrid preferences model [9] so that PrefDiv can take into consideration both qualitative and quantitative preferences when generating preferred diversified query results.
- We experimentally show that PrefDiv can successfully increase coverage of the result set compared to other alternatives, and achieves a significantly better *Relevancy-Diversity* trade-off ratio than other models.

Outline The rest of the paper is structured as follows. Section 2 presents the background and related works, Section 3 introduces Preferential Diversity and Section 4 describes the experimental environments and comparisons between our approach with other methods. Finally, Section 5 concludes.

2. BACKGROUND AND RELATED WORK

Many ranking techniques using preferences have been proposed. These are comprehensively surveyed in Stefanidis et al. [14]. As mentioned above, these techniques can be distinguished based on the type of preferences they support for filtering and ordering data. Mostly these techniques can handle only one type of preferences, either *quantitative* preferences or *qualitative* preferences. However, each preference type has its own advantages and disadvantages. Hybrid schemes support both qualitative and quantitative preferences in an attempt to exploit the advantages of both types of preferences while eliminating their disadvantages [12, 9]. In our work, we utilize the HYPRE model [9], which is the most recently developed hybrid scheme.

The HYPRE model and prototype system integrate qualitative and quantitative preferences by means of *preference strength* or *intensity*. In other words, a preference in the HYPRE model, is not seen as a binary option; instead, it allows users to express their preferences along with the intensity of that particular preference, i.e., how "strongly" a user feels about a fact. In the HYPRE model, users submit both qualitative and quantitative preferences along with an intensity value. The HYPRE model stores preferences in a labeled directed and acyclic graph. Each node in the graph represents a query predicate. Quantitative preferences are represented using edges that have the same starting and ending point. Qualitative preferences are represented by edges between two different nodes. Each edge is labeled with a value that represents the preference's intensity. Preference intensity is a decimal value between -1 and 1 and is used to express either a negative preference, a positive preference, or equality/indifference. In order to incorporate the nodes in a qualitative preference into the total order generated by the quantitative preferences, the qualitative preferences are converted into quantitative preferences by deriving an intensity value for these nodes based on the existing qualitative preference intensity value and a quantitative preference intensity value (or a default value if this does not exist). When a query is submitted, the system selects the best combination of preferences from the user's profile to filter and rank the query results.

Given our goal of achieving preferential diversity and the effectiveness of the HYPRE model in retrieving and ranking objects based on intensity values, we investigated ways to enhance the result of the HYPRE model with diversity using existing schemes. There are two widely used diversification models, *MaxMin* and *MaxSum*. The goal of these two diversification models is to select a subset S from the object space R, so that the *minimum* or the *total* pairwise distances of objects in S are maximized. Formally,

DEFINITION 1. *MaxMin* generates a subset of R with the maximum $f = min_{p_i,p_j \in S} dist(p_i, p_j)$ where *dist* is some distance function, $p_i \neq p_j$ for all subsets with the same size.

DEFINITION 2. *MaxSum* generates a subset of R with the maximum $f = \Sigma_{p_i,p_j \in S} dist(p_i, p_j)$ where *dist* is some distance function, $p_i \neq p_j$ for all subsets with the same size.

The most recently proposed diversity framework is *DisC Diversity* [6]. DisC Diversity is a method seeking to solve the diversification problem from a different perspective. In DisC Diversity, the number of retrieved diverse results is not an input parameter. Instead, users define the desired degree of diversification based on a combination of content dissimilarity and coverage between results. For any given user query, let R denote the set of all objects in the query result. DisC Diversity considers two objects o_i and $o_j \in R$ to be similar objects, if the distance between o_i and o_j is less than or equal to a tuning parameter r (radius). It selects the representative subset $S \in R$ according to the following conditions: (1) for any objects in R there should be at least one similar object in S and (2) all objects in S should be dissimilar with each other. These two conditions ensure both coverage and the dissimilarity property of a diverse result set. With the tuning parameter r, DisC Diversity is capable of supporting one important feature called *zooming* that allows users to adjust the value r. When r is increasing the result set becomes smaller and more diverse, and when r decreases, the result becomes larger and less diverse.

Our PrefDiv has several similarities with DisC Diversity. The key differences between PrefDiv and DisC Diversity are (1) PrefDiv follows the top-k paradigm that provides users with the option to specify the size of the final result set by assigning a value to parameter k, whereas DisC Diversity adjusts the size of the result set by changing its radius parameter r and (2) PrefDiv focuses on both the *relevance* of the result set with respect to the users' preference and the *diversity* of the result set; DisC Diversity focuses only on the most diverse representative subset. In addition, our implementation of PrefDiv achieves high coverage by means of the HYPRE model. In the next section, we will present PrefDiv in detail.

3. PREFDIV

In this section, we present the intuition and details of our proposed *Preferential Diversity* (PrefDiv) framework. Without loss of generality, we will present PrefDiv utilizing the HYPRE model which has motivated PrefDiv and used in its experimental evaluation. In this, PrefDiv utilizes the HYPRE model to retrieve relevant data

Table 1: Parameters of PrefDiv

Parameters	Range	Usage
I	$0 \leq$I≤ 1	Selects the objects with intensity value $\geq I$ in the initial set.
k	$0 \leq$k$\leq S$[1]	Specifies the size of result set.
r	$0 \leq$r$\leq M$[2]	Determines weather a pair of objects is similar.
A	$0 \leq$A≤ 1	Determines the number of objects to be promoted to the result set for each iteration.

[1] S = Size of Data Set [2] M = Max distance of dataset

and then outputs a representative set that balances the trade-off between relevance and diversity.

As mentioned in the previous section, the HYPRE model is capable of generating results that are most relevant to each individual user's interests. It achieves this by combining the intensity values of qualitative and quantitative preferences provided by each user. However, the most relevant result set is not necessarily the best quality result set. Experimenting with the HYPRE prototype, we have observed that data with high intensity value tends to be more similar to each other, compared to the other data in the result set of a query. Actually, it is not uncommon for some objects that fit the query requirements and the user's preferences to be hidden from the user due to their relatively low combined intensity values compared to other objects. Although data with a high intensity value have a higher possibility to fulfill a user's interests, this same data might not be able to provide a broad view of the data, which is essential for data exploration. Thus, even though retrieving data with a high combined intensity value is important, increasing the coverage of the result is equally important for improving the quality of query results.

When the size of a result set is fixed, an increase in coverage can be achieved by means of diversity. Increasing the coverage of a result set that contains the highest intensity value data will reduce the total intensity value of the result set, since some high intensity value data will be replaced with lower intensity value ones. This observation motivated our PrefDiv framework whose goal is to increase the coverage of the result set while minimizing its impact on the total intensity value, thus improving the overall quality of the results of user queries. At the same time, PrefDiv aims to allow users maximum flexibility in adjusting the degree of diversification – similar to DisC Diversity [6], and give the user full control over the trade-off between relevancy and accuracy. PrefDiv achieves this through four tunable parameters which are described next.

3.1 PrefDiv Parameters

There are four user-specified parameters that drive the behavior of PrefDiv (summarized in Table 1):

- I, which is the intensity value used to select the *initial* or *input* set of objects. It is passed to HYPRE, and HYPRE returns all the objects with an intensity value that is greater or equal to I.
- k, which represents the number of objects in the final result.
- r, which represents the radius of similarity. By assigning different values for r, users can directly control the definition of similar and dissimilar data items. Let o_i and o_j represent two different objects in our result set, and $dist(o_i, o_j)$ denote the distance between o_i and o_j with respect to some distance function. o_i and o_j are dissimilar to each other *iff* $dist(o_i, o_j) \geq r$.
- A, which ensures that a user's requirement for relevancy is fulfilled. A defines the distribution of the intensity values of objects in the final result. When $A = 1$, the final result set would simply be the top k objects from the initial set, i.e., the objects with the k highest intensity values. When $A = 0$, the final result contains k dissimilar objects from the initial set. When A is between 0 and 1 and given that PrefDiv is an iterative algorithm (Algorithm 1), the final result will have at least $A * k$ objects from every iteration. For example, when A = 0.5 and k = 20, the first iteration will select at least $20 * 0.5$ items into the final result set, the second iteration will select at least $20 * (0.5 * 0.5)$ items and so on; in each iteration A will be divided by half.

Algorithm 1 PrefDiv

Require:
A set of objects P, a size k, a relevancy parameter A, and a radius r.
Ensure:
A subset R of P.
create result set $R \leftarrow \emptyset$
create a new set $S \leftarrow \emptyset$
while there exist unmark objects in P **do**
  $S \leftarrow$ Pick k objects with highest intensity from P
  for all objects $o_i \in R$ **do**
    for all unmarked o_j in S that belongs to NEIGHBOR(o_i,r) **do**
      mark o_j as "Don't Select"
  while there exist unmarked objects in S **do**
    pick and remove unmarked object $o_j \in S$ that has the highest intensity value
    $R = R \cup o_i$
    if size of $R = k$ **then return** R
    else
      increase number of objects added to R by 1
    for all $o_j \in S$ **do**
      if o_j is unmarked and $o_j \in NEIGHBOR(o_i, r)$ **then**
        mark o_j as "Don't Select"
  while number of objects added to $R <$ $(A * k)$ **do**
    pick and remove object $o_i \in S$ that has highest intensity value
    $R = R \cup o_i$
  if this is the first iteration **then**
    create new set $G \leftarrow \forall objects \in S$ that marked as "Don't select"
  remove all objects in S from P
  decrease A by half
if size of $R < k$ and $\forall objects$ in P are marked **then**
  while size of $R < k$ **do**
    select object $o_i \in G$ that has highest intensity value
return R

3.2 PrefDiv Algorithm

PrefDiv is an iterative algorithm. It accepts as user input the above four parameters, I, k, r and A, and utilizes the HYPRE model to select the initial input set P which contains objects with an intensity value >= I. In each iteration, PrefDiv considers successive subsets S of size k from the input set P. Algorithm 1 formally describes the PrefDiv algorithm.

In the first iteration, PrefDiv considers the top-k objects, which form the first subset S. It selects the object with the highest intensity value from S and removes from S all its similar objects using the NEIGHBOR(o, r) function (in Algorithm 1). NEIGHBOR(o, r) takes an object o and a parameter r as inputs and returns all objects that are neighbors of o with respect to r. Then it proceeds to the next object with the highest intensity value in S. It again keeps this object in S and removes all its neighbors. PrefDiv proceeds in a similar fashion keeping in S only "representative" objects until all objects in S have been considered and kept as representative objects. An iteration completes by moving the $A * k$ objects with the highest intensity values from S to the final result set R. When an iteration completes, if the size of the result set R equals to k, PrefDiv returns R and terminates. Otherwise, it proceeds to the next iteration after selecting the next k objects from the input set P to form the next subset S.

PrefDiv returns a final result set R of user-specified size k, with a user-specified degree of diversity r, and a certain amount of relevancy with respect to the user's preferences by ensuring that at least $A * k$ objects with the highest intensity values from each iteration are part of the final result set.

4. EXPERIMENTAL EVALUATION

We implemented PrefDiv on top of the HYPRE prototype in order to evaluate its performance. Using real data, we compared PrefDiv to MaxMin and MaxSum [3, 10, 16] which follow the same top-down paradigm like PrefDiv, with respect to coverage and the provided Relevancy-Diversity trade-off.

4.1 Evaluation Metrics

In our experimental evaluation, we used the following four metrics:

DEFINITION 3. *Coverage* – Corresponds to the total possible number of objects touched by this result set.

DEFINITION 4. *Total intensity value* – Given one lists of objects, the total intensity value represents the sum of intensity values of the objects in that list.

DEFINITION 5. *Total pairwise distance* – Given one list of objects, the total pairwise distance represents the sum of the pairwise distances of the objects in that list.

DEFINITION 6. *Relevance and Diversity trade-off ratio* – Given a diversification model, the relevance and diversity trade-off ratio represents the percentage of total pairwise distance increase with respect to HYPRE, when sacrificing one percent of total intensity value with respect to HYPRE.

The HYPRE model generates results for a user query by selecting k objects with highest intensity value from all objects that have intensity value greater than or equal to I. Thus, the HYPRE model gives the upper bound of total intensity value for any model. On the other hand, by the definition of MaxSum, the total pairwise distance of a result set generated by MaxSum is the upper bound of the total pairwise distance for any model. Therefore, for any result set, in order to find out how much total intensity value and total pairwise distance are lost, one has to compare with the corresponding upper bound to calculate the performance loss.

The ***relevancy and diversity trade-off ratio*** for each model can be calculated as follows:

First, find the percentage of the ***total intensity value decrease*** for each model with respect to the upper bound of the total intensity value:

$$1 - \frac{Total\ intensity\ value\ of\ a\ model}{Total\ intensity\ value\ of\ results\ from\ HYPRE} \quad (1)$$

Second, find the percentage of the ***total pairwise distance for each model with respect to the upper bound*** of the total pairwise distance:

$$\frac{Total\ pairwise\ distance\ of\ a\ model}{Total\ pairwise\ distance\ of\ MaxSum} \quad (2)$$

Third, find the percentage of ***improvement of the total pairwise distance*** for each model with respect to HYPRE:

$$\frac{Result\ from\ (2)\ of\ this\ model}{Result\ from\ (2)\ of\ HYPRE} - 1 \quad (3)$$

Finally, utilize the results from above to find the ***Relevance and Diversity trade-off ratio***:

$$\frac{Improvement\ of\ total\ pairwise\ distance}{Total\ intensity\ value\ decrease} \quad (4)$$

4.2 Experimental Testbed

We implemented the PrefDiv prototype in Java 1.7, and used a MySQL server to store all the intermediate results from HYPRE. The HYPRE prototype also uses MySQL to store the data, and it uses the Neo4j 2.0 engine to store the preference graph. The HYPRE implements the queries for both MySQL and Neo4j in Java 1.7. The MaxMin and MaxSum algorithm used in our experiments were implemented based on Definitions 1 and 2.

In our experiments we used the same data previously used in the evaluation of the HYPRE prototype [9, 8]. This was extracted from an extended version of the DBLP dataset [15], that contains both the DBLP dataset (2011 version) and information about citations. The relations given by the DBLP dataset are stored in four tables: *author(aid, full name), citation(pid, cid), dblp(pid, title, venue, year, abstract)* and *dblp author(pid, aid)*. The generated preferences cover all possible types of preferences:

- *Venue Preference* (quantitative preference): Preference on the venue based on the venues where an author published
- *Author Preference* (quantitative preference): Preference for an author based on the co-author information
- *Preference of one author over another* (qualitative preference): Author A is preferred over author B.
- *Preference of one venue over another* (qualitative preference): Venue X is preferred over venue Y.
- *Negative Venue Preference* (quantitative preference): For each user A, a negative preference towards the venues where she did not publish but other authors that were cited by A did publish.

For diversity purposes, we consider the author's full name, the paper's title, the paper's venue, the paper's publication year, and whether there is abstract information or not, as the different important dimensions. We have used hamming distance to measure the pairwise distance for the data used in our experiments.

4.3 Experimental Results

Our experimental data set is retrieved through the HYPRE prototype (for the same user as in HYPRE experiments, i.e., uid=38437)

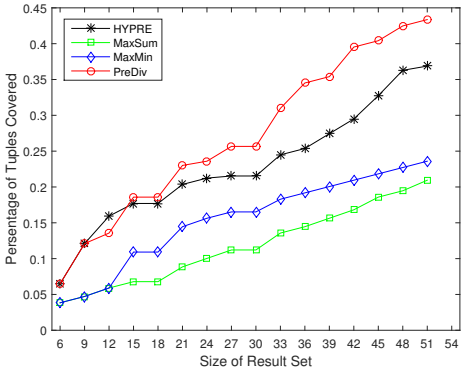

Figure 1: Coverage, A = 0.53, r = 2

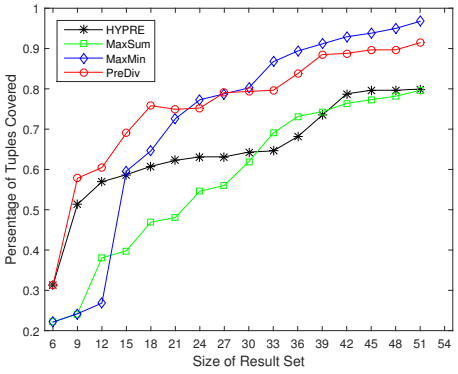

Figure 2: Coverage, A = 0.55, r = 3

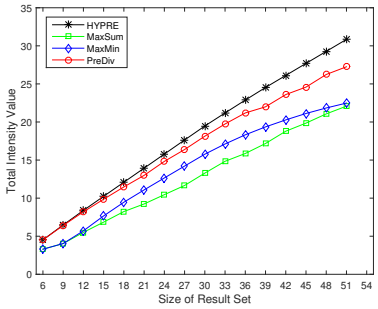

Figure 3: Total Intensity, A = 0.5, r = 3

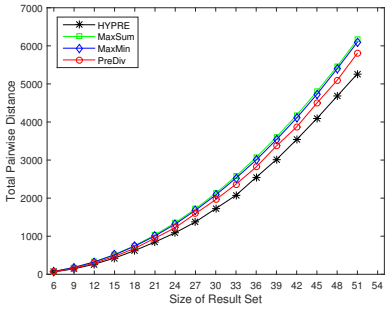

Figure 4: Total Pairwise Distance, A = 0.5, r = 3

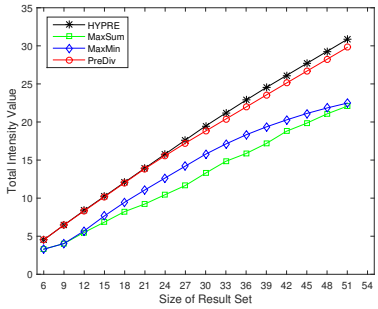

Figure 5: Total Intensity, A = 0.5, r = 2

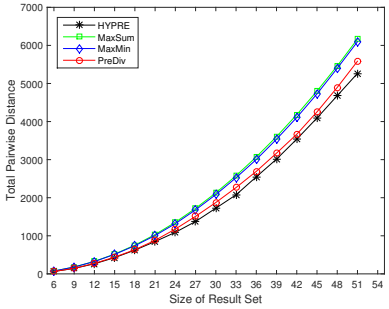

Figure 6: Total Pairwise Distance, A = 0.5, r = 2

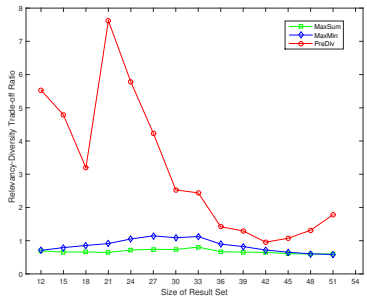

Figure 7: Relevancy-Diversity Trade-off Ratio, A = 0.5, r = 2

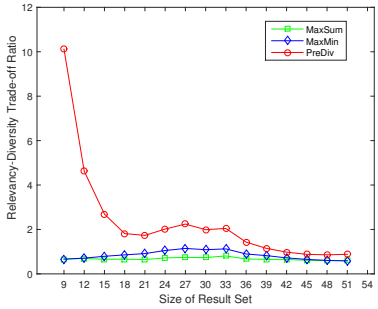

Figure 8: Relevancy-Diversity Trade-off Ratio, A = 0.5, r = 3

with intensity value > 0.1. This consists of 339 tuples with intensity values ranging from 0.7556 to 0.1 and the maximum pairwise distance between two tuples being 5 (i.e., same as the number of dimensions considered for diversity purposes). We evaluate the performance of all models based on these 339 tuples with both $r = 2$ and $r = 3$, because 2 and 3 are the middle points of dimensionality of our experimental data set. Also, in order to properly compare with other alternatives, we had to set the parameter A of PrefDiv to be a constant, and since we want to demonstrate the ability of PrefDiv to balance the trade-off between relevancy and diversity, for this experiment, we set A to be 0.5.

Coverage (Definition 3) We compared the average coverage of PrefDiv, MaxMin, MaxSum and HYPRE (which is simply the top-k objects with the highest intensity values) by initially assigning 6 to k (which is about 2% of the entire experimental data set) then increase k by 3 (about 2%) for each step. We took 16 steps in total, which means that we increased the size of the result set from relatively 2% of the experimental data set to 15% of the experimental data set.

In terms of average coverage, based on results shown in Figure 1, when r is 2, PrefDiv has on average 80.46% improvement over MaxMin, 232.98% improvement compared to MaxSum, and 15.09% improvement when compared to HYPRE. When increasing r to 3 (as shown in Figure 2), results indicate that PrefDiv has on average 19.97% improvement compared to MaxMin, 40.49% improvement compared to MaxSum, and 16.77% improvement compared to HYPRE. These results indicate that when compared to other alternatives, PrefDiv is able to improve the average coverage while retrieving a diverse but still reasonably sized subset of the results.

Table 2: Pairwise Distance k: 30, A: 0.5, r: 3

Distance	One	Two	Three	Four	Five
HYPRE	1	18	98	196	122
PrefDiv	1	3	34	133	264
MaxMin	0	0	0	93	342
MaxSum	0	0	1	45	389

Total intensity value and total pairwise distance (Definitions 4 and 5) Results with a higher total intensity value indicate that the objects within this result set are more relevant to users' preferences. Results with a higher total pairwise distance indicate that the objects within this result set are more dissimilar to each other. We compared these two metrics among different models (as illustrated in Figures 3, 4, 5, and 6). The results indicate that for the total pairwise distance, PrefDiv performs much closer to the upper bound (MaxSum) than the intermediate point between dissimilarity-focused approaches (MaxMin, MaxSum) and relevance-focused approaches (HYPRE). In terms of total intensity, PrefDiv again performs much closer to the upper bound (HYPRE) than the intermediate point.

Also by looking at the distribution of the total pairwise distances for each model, the results from PrefDiv cover a larger range when compared to MaxMin and MaxSum. This helps explain the advantage of PrefDiv in average coverage (Table 2, illustrates the distribution of distance with $k = 30$, $A = 0.5$, $r = 3$ that contains 435 pairwise distances between 30 tuples).

Relevance and Diversity trade-off ratio (Definition 6) As mentioned in Section 3, when measuring the quality of a result set, it is also important to measure the trade-off between relevancy and diversity. Hence for each percentage of total intensity value traded, we seek the most improvement in total pairwise distance.

In our experiments, we calculated the relevance and diversity trade-off ratio for each model (as illustrated in Figures 7 and 8), with k ranging from 12 to 51 (which is about 4% - 15% of the experimental data set), $A = 0.5$, and $r = 2$.

We observed that on average PrefDiv is able to outperform MaxMin by 363%, with a maximum of 834% improvement (when $k = 21$), and outperform MaxSum by 461% on average, with a maximum of 1180% improvement (when $k = 21$). When r is increased to 3, PrefDiv is able to outperform MaxMin by 214% on average, with a maximum of 652% improvement (when $k = 12$) and outperforms MaxSum by 264% on average, with a maximum of 667% improvement (when $k = 12$). These results indicate that PrefDiv is significantly more effective when dealing with the trade-off between relevance and diversity than other alternatives.

Take-away Our experiments show that when compared to other solutions, PrefDiv not only expanded the coverage of the result set (hence, increased the representability of the results), but it also performed significantly better when balancing the trade-off between relevance and diversity.

5. CONCLUSIONS

In this paper we presented a new framework called *Preferential Diversity* (PrefDiv) that aims to find the best balance point between relevance and diversity for query results. PrefDiv's capability to achieve this balance point was shown by implementing PrefDiv on top of the HYPRE preference model that incorporates both qualitative and quantitative preferences and utilizes user-specified parameters to shape the query result. We experimentally evaluated PrefDiv using real data extracted from DBLP. The experimental results showed that PrefDiv supports wider coverage of results than other models. In terms of the relevance-diversity trade-off ratio, PrefDiv can outperform other alternatives by up to 1,180%.

6. ACKNOWLEDGMENTS

We would like to thank Roxana Gheorghiu for sharing with us her HYPRE prototype and Cory Thoma and Marina Drosou, as well as all the anonymous reviewers, for their comments on the paper. This work was funded in part by NSF award OIA-1028162.

7. REFERENCES

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

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

[3] A. Borodin, H. C. Lee, and Y. Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *PODS*, pages 155–166, 2012.

[4] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. BÃijttche, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR*, pages 659–666, 2008.

[5] M. Drosou and E. Pitoura. Search result diversification. *ACM SIGMOD Record*, 39(1):41–47, 2010.

[6] M. Drosou and E. Pitoura. Result diversification based on dissimilarity and coverage. In *VLDB*, pages 13–24, 2012.

[7] P. Fraternali, D. Martinenghi, and M. Tagliasacchi. Top-k bounded diversification. In *SIGMOD*, pages 421–432, 2012.

[8] R. Gheorghiu. *Unifying Qualitative and Quantitative Database Preferences to Enhance Query P ersonalization*. PhD thesis, University of Pittsburgh, Sep 2014.

[9] R. Gheorghiu, A. Labrinidis, and P. K. Chrysanthis. A user-friendly framework for database preferences. In *CollaborativeCom*, pages 205–214, 2014.

[10] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW*, pages 381–390, 2009.

[11] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi. Big data and its technical challenges. *Commun. ACM*, 57(7):86–94, July 2014.

[12] W. Kiessling and G. Köstler. Preference SQL: design, implementation, experiences. In *VLDB*, pages 990–1001, 2002.

[13] A. Labrinidis. The big data - same humans problem. In *Proc. of Conference of Innovative Data Systems Research*, 2015.

[14] K. Stefanidis, G. Koutrika, and E. Pitoura. A survey on representation, composition and application of preferences in database systems. *ACM TODS*, 36(19), 2011.

[15] J. Tang, J. Zhang, R. Jin, Z. Yang, K. Cai, L. Zhang, and Z. Su. Topic level expertise search over heterogeneous networks. *Machine Learning J.*, 82(2):211–237, 2011.

[16] M. R. Vieira, H. L. Razente, M. C. N. Barioni, M. Hadjieleftheriou, D. Srivastava, C. Traina, and V. J. Tsotras. On query result diversification. In *ICDE*, pages 1163–1174, 2011.

[17] C.-N. Ziegler, J. A. K. Sean M. McNee, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW*, pages 22–32, 2005.