Diversified Recommendation on Graphs: Pitfalls, Measures, and Algorithms

Onur Küçüktunç[1,2], Erik Saule[1], Kamer Kaya[1], and Ümit V. Çatalyürek[1,3]
[1]Dept. Biomedical Informatics, The Ohio State University
[2]Dept. of Computer Science and Engineering, The Ohio State University
[3]Dept. of Electrical and Computer Engineering, The Ohio State University
kucuktunc.1@osu.edu, {esaule,kamer,umit}@bmi.osu.edu

ABSTRACT

Result diversification has gained a lot of attention as a way to answer ambiguous queries and to tackle the redundancy problem in the results. In the last decade, diversification has been applied on or integrated into the process of PageRank- or eigenvector-based methods that run on various graphs, including social networks, collaboration networks in academia, web and product co-purchasing graphs. For these applications, the diversification problem is usually addressed as a bicriteria objective optimization problem of relevance and diversity. However, such an approach is questionable since a *query-oblivious* diversification algorithm that recommends most of its results without even considering the query may perform the best on these commonly used measures. In this paper, we show the deficiencies of popular evaluation techniques of diversification methods, and investigate multiple relevance and diversity measures to understand whether they have any correlations. Next, we propose a novel measure called *expanded relevance* which combines both relevance and diversity into a single function in order to measure the coverage of the relevant part of the graph. We also present a new greedy diversification algorithm called `Best-Coverage`, which optimizes the *expanded relevance* of the result set with $(1-1/e)$-approximation. With a rigorous experimentation on graphs from various applications, we show that the proposed method is efficient and effective for many use cases.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process; H.3.4 [**Systems and Software**]: Performance evaluation (efficiency and effectiveness)

Keywords

diversity; relevance; graph mining; result diversification

1. INTRODUCTION

Algorithms developed for graph-based recommendation are very popular among web services; for instance, Amazon uses co-purchasing information to recommend products to its customers, IMDB recommends movies to its visitors based on the information such as director, cast, and ratings, and Google uses the web-graph and the user histories for personalized web search. The recommendations are usually made based on user preferences, either explicitly expressed or based on what she has been looking at recently. These preferences are used as the objects of known interest to seed the algorithms.

One of the common problems of popular recommendation algorithms is the pollution of top recommendations with many similar items, i.e., *redundancy*. It is typically not interesting to be recommended slight variations of the same product if you have a wide interest. The redundancy problem is solved via result *diversification*, which has gained a lot of attention in many fields recently [1, 8, 12, 14, 16, 19, 20, 21]. Diversification usually refers to the process of returning a set of items which are related to the query, but also dissimilar among each other. The problem of recommending a diversified set is inherently qualitative and is evaluated differently in various contexts [3, 4, 7, 23].

Most diversification studies in the literature rely on various assumptions, e.g., objects and queries are categorized beforehand [22], or there is a known distribution that specifies the probability of a given query belonging to some categories [1]. In the context of information retrieval or web search, since the search queries are often *ambiguous* or *multifaceted*, a query should represent the *intent* of an average user with a probability distribution [22]. Intent-aware methods in the literature aim to cover various relevant categories with one or more objects, or as TREC defines its diversity task, *"documents are judged on the basis of the subtopics, based on whether or not a document satisfies the information need associated with that subtopic"* [6].

In this work, we assume that the graph itself is the only information we have, and no categories or intents are available. We are interested in providing recommendations based on a set of objects of interest. The recommended items should be related to the user's interests while being dissimilar to each other. This particular problem has attracted a lot of attention recently, and many algorithms and evaluations have been proposed [5, 8, 13, 14, 16, 20, 24, 25].

Evaluation of algorithms' quality is one interest of the paper. Usually, algorithms are evaluated by expressing the problem as a bicriteria optimization problem. The first criteria is related to relevancy, e.g., the sum of the personalized PageRank scores, and the second is related to diversity, e.g., the density or the expansion ratio of the subgraph formed by the recommended set. These two criteria are either aggregated (often with a simple linear aggregation) or they are considered simultaneously with Pareto dominance (where the solutions are in the relevancy-diversity objective space).

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

As the first contribution, we show that such an evaluation is inappropriate. Indeed, we design *query-oblivious* algorithms for the two popular combinations of objectives that return most of the recommendations without considering the user's interests, yet, perform the best on these commonly used measures.

We argue that a result diversification algorithm should be evaluated under a measure which tightly integrates the query in its value. The *goodness* measure proposed in [20] has such a property; however, it is shown to be dominated by the relevance. We propose a new measure called *expanded relevance* ($exprel_\ell$) which computes the coverage of the relevant part of the graph. We show that the query-oblivious algorithms cannot optimize $exprel_\ell$.

We also investigate various quality indices by computing their pairwise correlations. This highlights that the *goodness* measure is highly correlated with the sum of ranking scores. That is the algorithms that perform well on *goodness* produce results sets which are not much different from top-k relevant set. The $exprel_\ell$ measure we propose appears to have no high correlation with other measures.

To optimize $exprel_\ell$ of the result set, we propose a greedy algorithm BestCoverage. Because of the submodular properties of $exprel_\ell$, BestCoverage is a $(1-1/e)$-approximation algorithm with complexity $\mathcal{O}(kn\Delta^\ell)$, where k is the number of recommended items, n is the number of vertices in the graph, and Δ is the maximum degree. We propose a relaxation of BestCoverage with complexity $\mathcal{O}(k\bar{\delta}^\ell\Delta^\ell)$, where $\bar{\delta}$ is the average degree of the graph. We experimentally show that the relaxation carries no significant harm to the *expanded relevance* of the results.

2. BACKGROUND

2.1 Problem Definition

We target the problem of diverse recommendation on graphs assuming that the user has a history or specified interests in some of the items. Therefore, the objective is to return a set of items which extend those interests.

Let $G = (V, E)$ be an undirected graph where $V = \{v_1, \ldots, v_n\}$ is the vertex set and E is the edge set. Given a set of m seed nodes $\mathcal{Q} = \{q_1, \ldots, q_m\}$ s.t. $\mathcal{Q} \subseteq V$, and a parameter k, return top-k items which are relevant to the ones in $\mathcal{Q}$. With diversity in mind, we want to recommend items not only relevant to $\mathcal{Q}$, but also covering different aspects of the query set.

2.2 PageRank and personalized PageRank

We define a random walk on G arising from following the edges (links) with equal probability and a random restart at an arbitrary vertex with $(1-d)$ teleportation probability. The probability distribution over the states follows the discrete time evolution equation:

$$\mathbf{p}_{t+1} = \mathbf{P}\ \mathbf{p}_t, \tag{1}$$

where $\mathbf{p}_t$ is the vector of probabilities of being on a certain state at iteration t, and $\mathbf{P}$ is the transition matrix defined as:

$$\mathbf{P}(u,v) = \begin{cases} (1-d)\frac{1}{n} + d\frac{1}{\delta(v)}, & \text{if } (u,v) \in E \\ (1-d)\frac{1}{n}, & \text{otherwise,} \end{cases} \tag{2}$$

where $\delta(v)$ is the degree of the vertex $v \in V$. If the network is ergodic (i.e., irreducible and non-periodic), (1) converges to a stationary distribution $\pi = \mathbf{P}\pi$ after a number of iterations. And the final distribution π gives the PageRank scores [2] of the nodes based on *centrality*.

In our problem, a set of nodes $\mathcal{Q}$ was given as a query, and we want the random walks to teleport to only those given nodes. Let us define a prior distribution p^* such that

$$p^*(v) = \begin{cases} 1/m, & \text{if } v \in \mathcal{Q} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

If we substitute the $(1/n)$s in (2) with p^*, we get a variant of PageRank, which is known as *personalized PageRank* (PPR) or *topic-sensitive PageRank* [10]. PPR scores can be used as the relevance scores of the items in the graph. Note that the rank of each seed node is reset after the system reaches to a steady state, i.e., $\forall q \in \mathcal{Q}$, $\pi_q \leftarrow 0$, since the objective is to extend $\mathcal{Q}$ with the results.

PPR is preferred as the scoring function in our discussions because (i) some of the methods in the experiments are variants of PPR which compute relevant but diverse set of results, (ii) some measures and objective functions are defined on the stationary distribution of PPR, and (iii) alternative scoring functions and probability distributions on graph produce similar results to PPR. On the other hand, the discussions on evaluations and some diversification techniques are independent of the preferred scoring function, hence we believe that the discussions will still interest the majority of the readers.

2.3 Result Diversification on Graphs

We classify the diversification methods for the recommendation problem based on whether the algorithm needs to rank the items only once or multiple times.

Diversification by query refinement. This set of algorithms rank the items k times to select the results one by one, and refine the search at each step.

GrassHopper [24] is a well-known diversification algorithm which ranks the graph k times by turning the highest-ranked vertex into a sink node at each iteration. Since the probabilities will be collected by the sink nodes when the random walk converges, the algorithm estimates the ranks with the number of visits to each node before convergence. GrassHopper uses matrix inversion to find the expected number of visits; however, inverting a sparse matrix makes it dense, which is not practical for the large and sparse graphs we are interested in. Therefore, we estimate the number of visits by iteratively computing the cumulative ranks of the nodes with PPR.

GSparse [13] employs an incremental ranking approach similar to GrassHopper, but the algorithm disconnects the selected node from the graph instead of converting it into a sink node. After executing the ranking function, the graph is sparsified for the next iteration by removing all the edges of the highest ranked node. This way, the graph becomes less dense around the selected nodes, hence, the remaining nodes at these regions will attract less visits during the random walk. The process is repeated until k nodes are selected.

Recently, manifold ranking has become an alternative to personalized PageRank and several diversification methods were proposed based on the idea of turning highly ranked nodes into sinks [5, 8, 25]. Aside from the ranking strategy, manifold ranking with sink points is quite similar to GrassHopper when the probabilities are estimated with cumulative scores. Since the manifold ranking is a different ranking of the graph, we carry out our experiments based only on PPR by leaving the discussion of using manifold ranking instead of PPR open for the time being.

Diversification by vertex selection. The following algorithms run the ranking function once, then carefully select a number of vertices to find a diverse result set.

DivRank [16] adjusts the transition matrix based on the number of visits to the vertices so far using a variant of random walks, called *vertex-reinforced random walks* (VRRW) [18]. It assumes that there is always an organic link for all the nodes returning back to the node itself which is followed with probability $(1-\alpha)$:

$$p_0(u,v) = \begin{cases} \alpha \frac{w(u,v)}{\delta(u)}, & \text{if } u \neq v \\ 1-\alpha, & \text{otherwise,} \end{cases} \tag{4}$$

where $w(u,v)$ is equal to 1 for $(u,v) \in E'$, and 0 otherwise. The transition matrix $\mathbf{P}_t$ at iteration t is computed with

$$\mathbf{P}_t(u,v) = (1-d)\ p^*(v) + d\ \frac{p_0(u,v)\ \eta_t(v)}{\sum_{z\in V} p_0(u,z)\ \eta_t(z)}, \tag{5}$$

where $p^*(v)$ is given in (3), and $\eta_t(v)$ is the number of visits of vertex v up to iteration t. It ensures that the highly ranked nodes collect more value over the iterations, resulting in the so called *rich-gets-richer* mechanism. In each iteration of VRRW, the transition probabilities from a vertex u to its neighbors are adjusted by the number of times they are visited up to that iteration t. Therefore, u gives a high portion of its rank to the frequently visited neighbors. Since the tracking of $\eta_t(.)$ is nontrivial, the authors propose to estimate it with *cumulative ranks* (CDivRank), i.e., the sum of the scores upto iteration t, or, since the ranks will converge after sufficient number of iterations, with *pointwise ranks* (PDivRank), i.e., the last score at iteration $t-1$.

A recently proposed algorithm, Dragon [20], employs a greedy heuristic to find a near-optimal result set that optimizes the *goodness* measure, which punishes the score when two neighbors are included in the results (see (15)). We will investigate this measure more in the upcoming section.

Frequently visited nodes tend to increase the ranks of their neighbors because of the smoothing process of random walks [16]. Based on this observation, algorithms using local maxima have been proposed. The Relaxed Local Maxima algorithm (k-RLM) [13] incrementally includes each local maxima within top-k^2 results to S until $|S| = k$ by removing it from the subgraph for the next iteration.

3. MEASURES AND EVALUATION

3.1 Classical relevance and diversity measures

Let us first review some classical measures for computing the relevance and diversity of the results with respect to the query. The measures are important since either they are typically used as –or a part of– the objective function of the diversification method, or the results are evaluated based on those measures.

Normalized relevance: The relevancy score of a set can be computed by comparing the original ranking scores of the resulting set with the top-k ranking list [20], defined as

$$rel(S) = \frac{\sum_{v\in S} \pi_v}{\sum_{i=1}^{k} \hat{\pi}_i}, \tag{6}$$

where $\hat{\pi}$ is the sorted ranks in non-increasing order.[1] Normalization with $\sum_{i=1}^{k} \hat{\pi}_i$ is preferred over $\sum_{v\in S} \pi_v$ since the distribution of scores in a random walk depends on the graph size, query, connectivity, etc., and normalized scores are comparable among different settings.

Difference ratio: A diversified result set is expected to be somewhat different than the top-k relevant set. Because the highly ranked nodes increase the ranks of their neighbors [16], the top-k results, recommended by the original PPR, is not diverse enough as shown in [19] and in our experiments. Nevertheless, the original result set has the utmost relevancy. This fact can mislead the evaluation of the experimental results. Therefore, we decided to measure the difference of each result set from the set of original top-k nodes. Given $\hat{S}$ to be the top-k relevant set, the difference ratio is computed with

$$diff(S,\hat{S}) = 1 - \frac{|S \cap \hat{S}|}{|S|}. \tag{7}$$

nDCG: We use normalized discounted cumulative gain (nDCG), for measuring the relevancy as well as the ordering of the results. It is defined as

$$\text{nDCG}_k = \frac{\pi_{s_1} + \sum_{i=2}^{k} \frac{\pi_{s_i}}{\log_2 i}}{\hat{\pi}_1 + \sum_{i=2}^{k} \frac{\hat{\pi}_i}{\log_2 i}}, \tag{8}$$

where π is the relevancy vector (e.g., stationary distribution of a random walk), $\hat{\pi}$ is the sorted π in non-increasing order, and $s_i \in S$ is the i^{th} point in result set S.

ℓ-step graph density: A variant of graph density measure is the ℓ-step graph density [20], which takes the effect of in-direct neighbors into account. It is computed with

$$dens_\ell(S) = \frac{\sum_{u,v\in S, u\neq v} d_\ell(u,v)}{|S| \times (|S|-1)}, \tag{9}$$

where $d_\ell(u,v) = 1$ when v is reachable from u within ℓ steps, i.e., $d(u,v) \leq \ell$, and 0 otherwise. The inverse of $dens_\ell(S)$ is used for the evaluation of diversity in [16].

ℓ-expansion ratio: As an alternative to density, *expansion ratio* and its variant *ℓ-expansion ratio* [14] measure the coverage of the graph by the solution set, computed with:

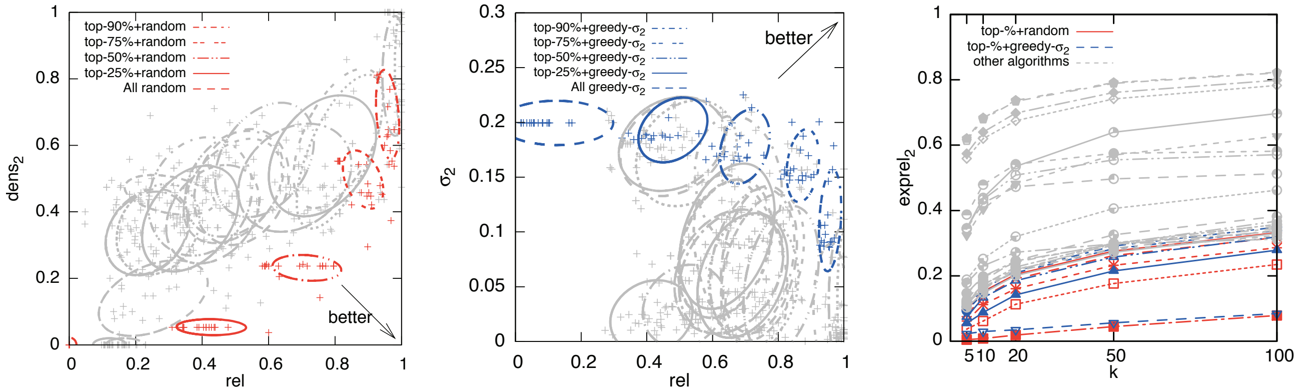
$$\sigma_\ell(S) = \frac{|N_\ell(S)|}{n}, \tag{10}$$

where the *expansion set* with 1-distance neighbors is defined as $N(S) = S \cup \{v \in (V-S) : \exists u \in S, (u,v) \in E\}$, and the *$\ell$-step expansion set* is defined in [14] as:

$$N_\ell(S) = S \cup \{v \in (V-S) : \exists u \in S, d(u,v) \leq \ell\}. \tag{11}$$

Note that the intent-aware measures, such as intent-aware expected reciprocal rank (ERR-IA) [4], α-normalized discounted cumulative gain (α-nDCG@k) [7], intent-aware mean average precision (MAP-IA) [1], are not included to the discussions, but they are important measures for evaluating the diversity of the results when data and queries have some already known categorical labels. Our problem has no assumptions of a known distribution that specifies the probability of an item belonging to a category.

[1] $\hat{\pi}$ does not denote estimated or predicted relevance scores.



(a) *rel* vs. *dens*$_2$ (b) *rel* vs. σ_2 (c) *exprel*$_2$

Figure 1: Evaluation of top-%+random (red) and top-%+greedy-σ_2 (blue) methods versus other algorithms (gray) based on selected relevance/diversity measure pairs and combined *exprel*$_2$ measure. The other algorithms include `GrassHopper`, `DivRank`, `Dragon`, k-`RLM`, `GSparse`, and `BestCoverage`, but they are not highlighted here since we do not want to prematurely compare those against each other.

3.2 Bicriteria optimization measures

Maximum Marginal Relevance (MMR) [3] is the most popular diversification method that optimizes a bicriteria objective, *marginal relevance*, which is a linear combination of independently measured relevance and novelty. The method greedily and implicitly optimizes the following objective assuming that the similarity of all items to the query items are already computed in π:

$$f_{MMR}(S) = (1-\lambda)\sum_{v\in S}\pi_v - \lambda \sum_{u\in S} \max_{\substack{v\in S\\ u\neq v}} sim(u,v), \quad (12)$$

where λ is the importance of relevance over novelty and *sim* is a similarity metric. The problem with (12) is that two different measures are aggregated without taking their compatibility into account.

The same premise is also valid for any type of linear aggregation of a relevance and a diversity measure. For example, [14] tries to optimize the following diversified ranking measure:

$$f_L(S) = \sum_{v\in S}\pi_v + \lambda\frac{|N(S)|}{n}, \quad (13)$$

where λ is the tradeoff between relevance and diversity, and the diversity of the result set is measured with the expansion ratio. Similarly in [15], relevance part is scaled with $(1-\lambda)$.

Other bicriteria objectives include max-sum diversification, which reduces to MaxSumDispersion problem, max-min diversification, which reduces to MaxMinDispersion problem, etc. For example, *k-similar diversification set problem* [21] is defined based on MaxSumDispersion as:

$$f_{MSD}(S) = (k-1)(1-\lambda)\sum_{v\in S}\pi_v + 2\lambda\sum_{u\in S}\sum_{\substack{v\in S\\ u\neq v}} div(u,v), \quad (14)$$

where $div(u,v)$ can be selected as a weighted similarity, *tf/idf* cosine similarity, or the Euclidean distance depending on the problem. We refer the reader to [9] for more information on objectives and distance functions.

3.3 Bicriteria optimization is not the answer

We argue that bicriteria optimization is inappropriate, and hence, the diversification methods that seem to optimize both criteria are problematic. Let us return back to our original problem: the items in a graph structure are ranked based on a given query and a ranking method (e.g., `PPR`), and our aim is to rerank those items so that we can include more results from different aspects of the query and reduce redundancy of top-k relevant set.

Suppose that we work on the web graph and we want to *diversify* the results of a search engine which displays $k = 10$ results to the user. Do you think the quality of the top-k list would improve if we replace some results from the end of the list with random web pages?

We design two *query-oblivious* algorithms for the two popular combinations of objectives, which are monotonous (e.g., linear or quadratic) aggregations of max-sum relevance and max-sum diversity (graph density *dens* or expansion ratio σ) objectives. The algorithms will return some of the results without considering the user's interests, yet, will perform the best on the following commonly used measures:

- **top-%+random:** returns a given percentage of the results (e.g., 50%, 75%, etc.) from top-k, and the rest randomly from the search space.
- **top-%+greedy-σ_2:** returns a given percentage of the results (e.g., 50%, 75%, etc.) from top-k, and try to maximize σ_2 with the rest of the results without taking the query into account.

To prove our point, we compute the normalized relevance (*rel*) and selected diversity measure (*dens*$_2$ and σ_2) of the results for the diversification methods in the literature and for the *query-oblivious* algorithms.[2] We fit a multi-variate Gaussian on top of the results to show the mean and moments of the distribution when two objectives are considered simultaneously. A result which further minimizes *dens*$_2$ and maximizes *rel* and σ_2 is favorable and better. This is shown with an arrow in the Figs. 1(a) and 1(b).

[2] Figure 1 gives only the results for AMAZON0601 dataset using scenario 3 queries and $k = 20$. Comparisons of *query-oblivious* methods on given bicriteria measures and *exprel*$_2$ for other datasets and query types are provided in the supplementary material: `http://bmi.osu.edu/hpc/data/Kucuktunc13WWW/randoms.pdf`

Figure 1(a) shows the results of **top-%+random** as well as other algorithms with respect to *rel* vs. *dens*$_2$ evaluation. Figure 1(b) similarly shows the results of **top-%+greedy-**σ_2 as well as other algorithms with respect to *rel* vs. σ_2 evaluation. Here, *query-oblivious* methods seem to recommend the best result sets when a bicriteria evaluation is used. Yet, we know that those algorithms are designed to trick the evaluation, as well as produce useless results in user's point of view.

Using only the first half of top-k results gives a normalized relevance score greater than or equal to 0.5 since the ranks are sorted in non-increasing order. Furthermore, the ranks has a power-law distribution that makes *rel* much higher than 0.5. Therefore, the relevance objective is mostly satisfied.

We further argue that no matter which relevance and diversity measures are selected, there always exists a *query-oblivious* algorithm which optimizes both measures in a meaningless way, useless in practice but looks great on the paper. This is the problem of evaluating result diversification as a bicriteria optimization problem with **a relevance measure that ignores diversity**, and **a diversity measure that ignores relevancy**.

3.4 Combined measures

As a result of our experiments on bicriteria optimization, we argue that we need a combined measure that tightly integrates both relevance and diversity aspects of the result set. It is reasonable to design the combined measure based on the query, the rankings, and the graph structure we already have.

The *goodness* measure [20] is a step towards a meaningful combined measure. It penalizes the score when two results share an edge, meaning that they are neighbors and they possibly increase their ranks by feeding each other during the random walk. The measure is computed with

$$f_G(S) = 2\sum_{i\in S}\pi_i - d\sum_{i,j\in S}\mathbf{A}(j,i)\pi_j - (1-d)\sum_{j\in S}\pi_j\sum_{i\in S}p^*(i), \qquad (15)$$

where $\mathbf{A}$ is the row-normalized adjacency matrix of the graph. However, we will show in Section 3.5 that *goodness* is highly dominated by relevance, which reflects negatively on the results of `Dragon` in the experiments.

We present a combined measure of the ℓ-step expansion ratio (σ_2) and relevancy scores (*rel*), which are two popular diversity and relevance measures in the literature, in order to quantify the *relevant-part coverage* of the graph:

ℓ-step expanded relevance:

$$exprel_\ell(S) = \sum_{v\in N_\ell(S)}\pi_v \qquad (16)$$

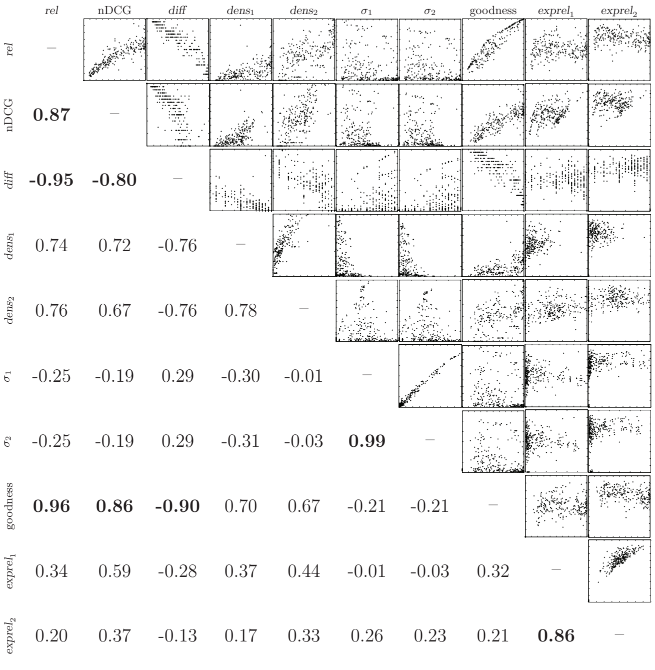
where $N_\ell(S)$ is the ℓ-step expansion set of the result set S, and π is the `PPR` scores of the items in the graph.

This new measure explicitly evaluates the diversity of the results in terms of *coverage* with the given set. In other words, when two results are close to each other in the graph, their expansion sets intersect narrowing the covered part of the search space. Therefore, the items having separate expansion sets will increase the coverage. However, coverage is not the only aspect of *exprel*$_\ell$. The proposed measure also takes the ranking scores into account, and hence the quality of the covered part.

The effect of each result is limited with the given ℓ parameter, i.e., a result covers only its neighbors in the graph if $\ell = 1$, or neighbors of neighbors if $\ell = 2$. Higher values of ℓ are generally not preferred since the expansion set tends to cover most of the graph in those cases.

An important property of *exprel*$_\ell$ measure is that *query-oblivious* algorithms cannot optimize it. Because, the highest ranked items are mostly not diverse enough, and the rest of the results (randomly selected independent of the query) will not contribute much to the measure. Figure 1(c) shows that neither top-%+random (red) nor top-%+greedy-σ_2 (blue) can optimize the measure while the diversification algorithms (gray) can score higher. This proves the validity of the measure for diversification.

Table 1: Correlations of the different relevance, diversity, and combined measures. Pearson correlation scores are given on the lower triangle of the matrix. High correlations are highlighted.



	rel	nDCG	*diff*	*dens*$_1$	*dens*$_2$	σ_1	σ_2	goodness	*exprel*$_1$	*exprel*$_2$
rel	–									
nDCG	**0.87**	–								
diff	**-0.95**	**-0.80**	–							
dens$_1$	0.74	0.72	-0.76	–						
dens$_2$	0.76	0.67	-0.76	0.78	–					
σ_1	-0.25	-0.19	0.29	-0.30	-0.01	–				
σ_2	-0.25	-0.19	0.29	-0.31	-0.03	**0.99**	–			
goodness	**0.96**	**0.86**	**-0.90**	0.70	0.67	-0.21	-0.21	–		
exprel$_1$	0.34	0.59	-0.28	0.37	0.44	-0.01	-0.03	0.32	–	
exprel$_2$	0.20	0.37	-0.13	0.17	0.33	0.26	0.23	0.21	**0.86**	–

3.5 Correlations of the measures

We investigate the mentioned relevance, diversity, and combined measures by computing their pairwise correlations based on the results of the algorithms given in Section 2.3 as well as the query-oblivious top-%+random methods given in the previous section. Table 1 shows the correlations of 10 measures as scatter plots as well as their correlation scores.[3]

[3] Table 1 shows only the measure correlations on AMAZON0601 dataset and with $k = 20$. The results are consistent across various datasets, scenarios, and k values. A complete comparison set is provided in the supplementary material: `http://bmi.osu.edu/hpc/data/Kucuktunc13WWW/corr.pdf`

For the relevance measures, *rel* is highly correlated with nDCG although the latter considers the order of the results. *rel* is also anti-correlated with *diff*, meaning that as the ratio of results other than top-k start to increase, the normalized relevance decreases accordingly.

For the graph diversity measures, ℓ-step expansion ratios (σ_1 and σ_2) are highly correlated among each other. On the other hand, graph density-based measures ($dens_1$ and $dens_2$) do not seem to have any high correlation with other measures.

Among the combined measures, *goodness* is highly correlated with *rel*. This highlights that the *goodness* measure is dominated by the sum of ranking scores, meaning that algorithms that perform better on *goodness* do not return results that are much different from the top-k results of PPR.

The proposed $exprel_\ell$ measure, on the other hand, appears to have no high correlation with any of the other relevance or diversity measures, proving that it is something different than the already known measures. Although the expanded relevance is based on both *rel* and expansion ratio (σ), very low correlation is observed in the results.

4. BEST COVERAGE METHOD

Our strategy so far was to review the attempts to find a good objective function for the result diversification problem on graphs. We have shown that a bicriteria optimization of relevance and diversity can be tricked, and a combined measure should be constructed carefully. The proposed $exprel_\ell$ measure seems to cover both aspects of the intended objective, yet cannot be optimized by the query-oblivious algorithms. We argue that this novel measure can be naturally used as an objective function of a diversification algorithm.

4.1 Problem formulation and complexity

Given a graph $G = (V, E)$, a vector of ranking scores π (stationary distribution of PPR scores in our case) computed based on the query set $\mathcal{Q}$, and the number of required results k, our objective is to maximize the expanded relevance ($exprel_\ell$) of the result set S:

$$S = \operatorname*{argmax}_{\substack{S' \subseteq V \\ |S'|=k}} exprel_\ell(S') = \operatorname*{argmax}_{\substack{S' \subseteq V \\ |S'|=k}} \sum_{v \in N_\ell(S')} \pi_v, \qquad (17)$$

where $N_\ell(S')$ is the ℓ-step expansion set. We refer to this problem as *$exprel_\ell$-diversified top-k ranking* (DTRℓ).

However, it is not hard to see that the objective of finding a subset of k elements that maximizes the expanded relevance is NP-hard. Assuming the graph G and the ranking scores π are arbitrary, DTRℓ is a generalization of the *weighted maximum coverage problem* (WMCP) which is NP-Complete [11]. WMCP is expressed as a set O of objects o_i with a value ω_i and z sets of objects $r_j \subseteq O$, $R = \{r_1, r_2, \ldots, r_z\}$. The problem is to select a subset of R, $P \subseteq R$ such that $|P| = x$ which maximizes $\sum_{o_i \in \{r_j : r_j \in P\}} \omega_i$. The key of the reduction for $\ell = 1$ is to construct an instance of DTRℓ with a bipartite graph $G = (V = R \cup O, E)$ where $(r_j, o_i) \in E$ iff $o_i \in r_j$. We set $\pi_{r_j} = 0$, $\pi_{o_i} = \omega_i$ and $k = x$. The solutions of DTRℓ are dominated by sets S where all the vertices are in R. Indeed, since $\pi_{r_j} = 0, \forall r_j$ there is no advantage in selecting a vertex in O. The rest of the reduction is obvious for $\ell = 1$. For other values of ℓ, the reduction is similar, except each edge of the bipartite graph is replaced in a path of ℓ edges.

Note that the proposed objective in (17) is independent of ordering since the function is defined over an unordered set. This is usually reasonable because there is an assumption that users will consider all k results [1, 14, 20]. In practice, different users may stop at different number of results, hence, several DCG-based metrics are commonly used to compute the importance of returning results in an *ideal ordering*. The near-optimal solutions that we will present in the following section can still output an ordered set of results based on the marginal utility of each selected item at the moment of its inclusion.

4.2 Greedy solution: BestCoverage

Although the optimal solution of the proposed objective function (see (17)) is NP-hard, we will show that a greedy solution that selects the item with the *highest marginal utility* at each step is the best possible polynomial time approximation for the problem.

Let us define the *marginal utility* for a given vertex v and result set S as $g(v, S)$, such that $g(v, \emptyset) = exprel_\ell(\{v\})$ before any results are selected, and $g(v, S) = \sum_{v' \in V'} \pi_{v'}$ where $V' = N_\ell(\{v\}) - N_\ell(S)$ represents the ℓ-step expansion set of vertex v without the items that have already been covered by another result. In other words, $g(v, S)$ is the increase on the $exprel_\ell$ measure if v is included to the result set, i.e., $exprel_\ell(S \cup \{v\}) = exprel_\ell(S) + g(v, S)$.

ALGORITHM 1: BestCoverage

```
Input: $k, G, \pi, \ell$
Output: a list of recommendations $S$
$S = \emptyset$
while $|S| < k$ do
    $v^* \leftarrow \operatorname{argmax}_v g(v, S)$
    $S \leftarrow S \cup \{v^*\}$
return $S$
```

Algorithm 1 incrementally selects the item with the highest marginal utility in each step, then includes it to the result set S. This way, the items that contribute the most to the expanded relevance of the final results are greedily selected as a solution to the given optimization problem. In order to show that the greedy algorithm solves the problem quite well, we first prove that the $exprel_\ell$ is a submodular function:

Definition 4.1. (Submodularity) *Given a finite set V, a set function $f : 2^V \to \mathbb{R}$ is submodular if and only if for all subsets S and T such that $S \subseteq T \subseteq V$, and $j \in V \setminus T$, $f(S \cup \{j\}) - f(S) \geq f(T \cup \{j\}) - f(T)$.*

Lemma 4.2. *$exprel_\ell$ is a submodular function.*

The proof of the lemma follows directly from the definitions of submodularity and $exprel_\ell$. Greedy algorithms are known to generate good solutions when maximizing submodular functions with a cardinality constraint and were used in [1, 14].

Theorem 4.3. *[17] For a submodular set function f, let S^* be the optimal set of k elements that maximizes $f(S)$, and S' be the k-element set constructed greedily by selecting an element one at a time that gives the largest marginal increase to f. Then $f(S') \geq (1 - 1/e)f(S^*)$.*

Corollary 4.4. *BestCoverage is an $(1 - 1/e)$-approximation algorithm for the $exprel_\ell$-diversified top-k ranking problem.*

4.3 Analysis and relaxation of the algorithm

BestCoverage (BC) is a $(1-1/e)$-approximation for maximizing $exprel_\ell$ with complexity $\mathcal{O}(kn\Delta^\ell)$ where n is the number of vertices in the graph, k is the number of recommended objects, and Δ is the maximum degree of the graph.

Obviously, the implementation in Algorithm 1 can be improved by storing the *marginal utility* for every vertex at the expense of $\mathcal{O}(n)$ space, and updating only the vertices that the inclusion of v^* to S would affect. However, for $\ell = 2$, the number of vertices to be updated is $|N_4(\{v^*\})|$, which is $\mathcal{O}(\Delta^4)$ in the worst case. Initializing the marginal utility incurs a cost of $\mathcal{O}(n\Delta^\ell)$. Once a vertex is added to set S, the impact of its distance ℓ neighbors must be adjusted. For a given vertex, adjusting its impact costs $\mathcal{O}(\Delta^\ell)$. For each iteration of the algorithm the impact of at most Δ^ℓ neighbors need to be adjusted. Though, each vertex adjusts its impact only once, so there are $\mathcal{O}(\min\{n, k\Delta^\ell\})$ adjustments. Finally, selecting the vertex with maximal marginal utility requires $\mathcal{O}(n)$ operations[4] per iteration. The overall complexity of the algorithm is $\mathcal{O}(n\Delta^\ell + \min\{n, k\Delta^\ell\}\Delta^\ell + kn)$.

ALGORITHM 2: BestCoverage (relaxed)

```
Input: $k, G, \pi, \ell$
Output: a list of recommendations $S$
$S = \emptyset$
Sort($V$) w.r.t $\pi_i$ non-increasing
$S1 \leftarrow V[1..k']$, i.e., top-$k'$ vertices where $k' = k\bar{\delta}^\ell$
$\forall v \in S1, g(v) \leftarrow g(v, \emptyset)$
$\forall v \in S1, c(v) \leftarrow$ Uncovered
while $|S| < k$ do
    $v^* \leftarrow \text{argmax}_{v \in S1}\, g(v)$
    $S \leftarrow S \cup \{v^*\}$
    $S2 \leftarrow N_\ell(\{v^*\})$
    for each $v' \in S2$ do
        if $c(v') =$ Uncovered then
            $S3 \leftarrow N_\ell(\{v'\})$
            $\forall u \in S3, g(u) \leftarrow g(u) - \pi_{v'}$
            $c(v') \leftarrow$ Covered
return $S$
```

With this optimization, most of the time is spent on initializing the marginal utility. We experimentally found that the returned results are chosen from top-k' results of PPR ranks, where k' is proportional to k and the average degree of the graph. We propose a relaxation of BestCoverage which only considers including in the result set the top-$k\bar{\delta}^\ell$ highest ranked vertices solely based on the relevance scores where $\bar{\delta}$ is the average degree of the graph. All the vertices of the graph still contributes to marginal utility. The complexity of the relaxed version drops to $\mathcal{O}(\min\{n, k\Delta^\ell\}\Delta^\ell + k\min\{n, k\bar{\delta}^\ell\})$ since the cost of the computation of the initial marginal utility is now asymptotically dominated by the cost of adjusting them. Algorithm 2 gives the relaxed BestCoverage algorithm with all mentioned improvements. The impact of the relaxation on the quality of the solution will be discussed in Section 5.3.

5. EXPERIMENTS

5.1 Datasets

In the experiments we use one graph instance for each targeted application area, i.e., product recommendation on shopping websites, collaborator and patent recommendation in academia, friend recommendation on social networks, and personalized web search. The graphs are publicly available at Stanford Large Network Dataset Collection[5]. In summary, amazon0601 is the Amazon product co-purchasing network collected on June 2003. ca-AstroPh is the collaboration network between authors of the papers submitted to arXiv astrophysics category. cit-Patents is the citation network between U.S. patents granted between 1975 and 1999. soc-LiveJournal1 is the graph of LiveJournal social network, and web-Google is the web graph released in 2002 by Google.

The mentioned graphs are re-labeled, converted into undirected graphs. The properties of the graphs are given in Table 2. Note that $\bar{\delta}$ is the average degree of the graph, D is the diameter of the graph, i.e., maximum undirected shortest path length, $D_{90\%}$ is the 90-percentile effective diameter, and CC is the average clustering coefficient.

Table 2: Properties of graphs used in experiments.

Dataset	$\lvert V\rvert$	$\lvert E\rvert$	$\bar{\delta}$	D	$D_{90\%}$	CC
amazon0601	403.3K	3.3M	16.8	21	7.6	0.42
ca-AstroPh	18.7K	396.1K	42.2	14	5.1	0.63
cit-Patents	3.7M	16.5M	8.7	22	9.4	0.09
soc-LiveJournal1	4.8M	68.9M	28.4	18	6.5	0.31
web-Google	875.7K	5.1M	11.6	22	8.1	0.60

5.2 Scenarios and query generation

We generate the queries for the experiments based on three different real-world scenarios:

Scenario 1: A random vertex in the graph is selected as the query. This scenario represents the case where the system does not have any information on the user. For product recommendation, the user can be visiting a product page without signing in to the system. For academic recommendation tasks, a professor can be looking for collaborators.

Scenario 2: A random vertex v along with 10–100 vertices within two distance to v are selected as a query. In this scenario, v and the selected vertices represent an area of interest. For example, the user can be searching for a product within a category, or interested in an academic field. In a social network, the friend list of a person can be used as the query for friend suggestion.

Scenario 3: 2 to 10 random vertices are selected as different interests of the user, and a total of 10 to 100 vertices around those interests are added to the query set. Multiple areas of interest is the most common use case for these applications where users are registered to the system and already have a search or purchase history.

For each dataset, 750 queries were generated, where the average number of the seed nodes varies between 1 and 50 for the scenarios 1 and 3, respectively. In total 3,750 query sets representing different real-world cases were used in the experiments.

[4] It might appear that using a fibonnacci heap should allow to reach a better complexity, but we require the extract-max and decrease-key operations which are incompatible.

[5] Available at: http://snap.stanford.edu/data/index.html

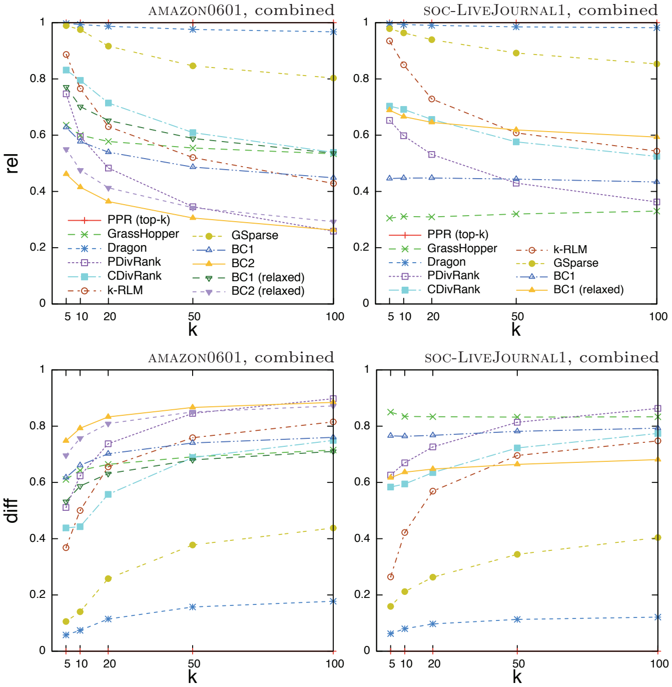


Figure 2: Normalized relevance (*rel*) and different ratio (*diff*) scores with varying k. Dragon and GSparse return results around 70% similar to the top-k relevant set, this is generally not enough to improve the diversity of the results.

5.3 Results

We experiment with the algorithms given in Section 2.3, the datasets described in Section 5.1, and the queries defined in Section 5.2. For the methods that use the ranking scores of PPR, we fix $d = 0.9$ and the number of PPR iterations to 20 in order to be consistent between different queries. For the VRRW computation of DivRank methods, we set $\alpha = 0.25$ and the number of iterations to 50 since VRRW usually takes more iterations to converge. All ranking functions are implemented efficiently with sparse matrix-dense vector multiplication (SpMxV) operations.

On amazon0601, ca-AstroPh, and soc-LiveJournal1 datasets, we observed that the results of different scenarios are similar. Hence, we combine the scenarios and display the results on all queries[6]. Also note that the results of BC$_2$ and its relaxation are omitted from the plots of soc-LiveJournal1 dataset because of the impractical runtimes.

Normalized relevance (*rel*) and difference ratio (*diff*) plots in Figure 2 show that Dragon and GSparse methods almost always return the results having 70% similar items to top-k relevant set, and more than 80% *rel* score. A low *rel* score is not an indication of being dissimilar to the query (unless *rel* $\rightarrow 0$); on the other hand, since the scores have a power-law distribution, a high *rel* score usually implies that the algorithm ignored the diversity of the results and did not change many results in order to keep the relevancy high. The actual *diff* measures are also given in Figure 2.

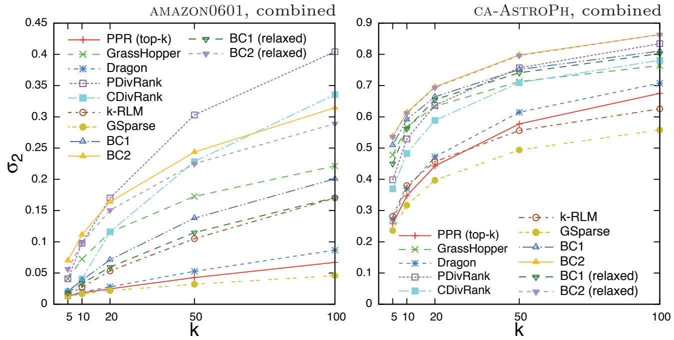


Figure 3: Coverage (σ_2) of the algorithms with varying k. BestCoverage and DivRank variants have the highest coverage on the graphs while Dragon, GSparse, and k-RLM have similar coverages to top-k results.

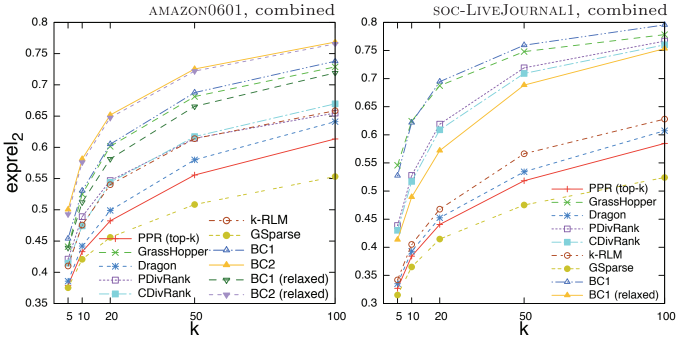


Figure 4: Expanded relevance (*exprel*$_2$) with varying k. BC$_1$ and BC$_2$ variants mostly score the best, GrassHopper performs high in soc-LiveJournal1. Although PDivRank gives the highest coverage on amazon0601 (Fig. 3), it fails to cover the relevant parts.

Based on the expansion ratios (σ_2) in Figure 3, BestCoverage and DivRank variants, especially PDivRank and BC$_2$, have the highest scores, hence the highest coverage on the graphs with their diversified result set. Dragon, GSparse, as well as k-RLM have expansion ratios similar to the top-k results, meaning that these algorithms do not improve the coverage of the given graphs enough. GSparse reduces the expansion ratio even more than the top-k set, proving that it is inappropriate for the diversification task. It is important to note that σ_2 scores are meaningless by itself since query-oblivious greedy-σ_2 algorithm would maximize the coverage.

Figure 4 shows the proposed expanded relevance scores (*exprel*$_2$) of the result sets. BC$_1$ and BC$_2$ variants are significantly better than the other algorithms, where GrassHopper is able to score closer to BestCoverage only in soc-LiveJournal1 dataset. Although DivRank variants perform the highest based on expansion ratio (see Figure 3), their results are shown to be unable to cover the relevant parts of the graph as they score lower than BestCoverage variants.

For cit-Patents and web-Google datasets, we report the results on queries of scenarios 1 and 3 separately. Here we omit the results of scenario-2 queries since they are in between scenarios 1 and 3. These plots share the conclusions we have made so far based on the results on previous three datasets; however, they present different behavior based on the chosen scenario, so we provide a deeper analysis on those.

[6] Due to space limitation we only display one plot per observation highlighted in the text. The complete set of plots for each dataset, scenario, and measure is provided in the supplementary material: `http://bmi.osu.edu/hpc/data/Kucuktunc13WWW/results.pdf`

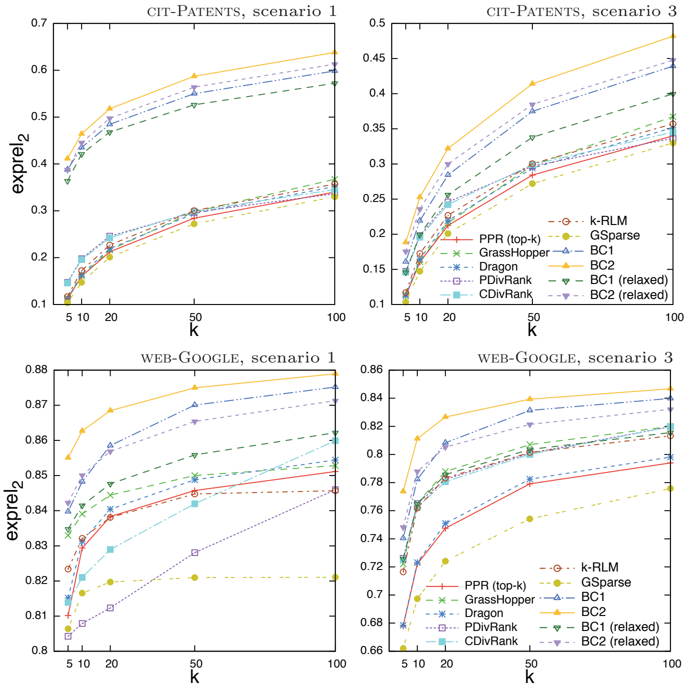


Figure 5: Expanded relevance (*exprel*$_2$) with varying k. BestCoverage variants perform higher than usual on cit-Patents dataset with scenario-1 queries because of the low average degree ($\bar{\delta} = 8.7$) and low clustering coefficient ($CC = 0.09$) of the graph. The relaxed algorithms perform closer to their originals, meaning that they were both efficient and effective on this type of sparsely connected graphs.

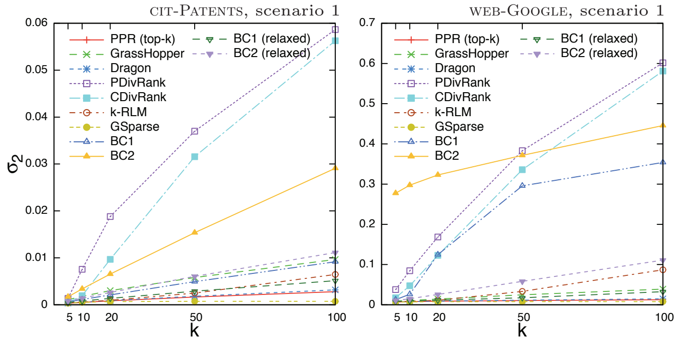


Figure 6: Coverage (σ_2) of the algorithms with varying k. DivRank variants appear to be implicitly optimizing the *size* of the expansion set, without considering whether those results are still relevant to the query (cf. corresponding *exprel*$_2$ in Figure 5).

Figure 5 shows that the *exprel*$_2$ results on cit-Patents dataset vary based on the scenario chosen to generate the queries. In fact, the results are higher than normal for scenario-1 queries. This is because of the low average degree ($\bar{\delta} = 8.7$) and low clustering coefficient ($CC = 0.09$) of the graph. Also note that the relaxations of BC$_1$ and BC$_2$ perform closer to BC$_1$ and BC$_2$, meaning that the relaxed algorithms are both efficient and also effective on this type of sparsely connected graphs.

It is also more clear on plots in Figure 6 that DivRank variants implicitly optimize the expansion ratio (σ_2) of the results, but without considering whether those results are still relevant to the query. As a striking example of scenario-1 queries on web-Google dataset, it is quite interesting to see an algorithm to perform the best with respect to the *size* of the expansion set, but almost the worst with respect to the *relevancy* of the same set (see Figure 5).

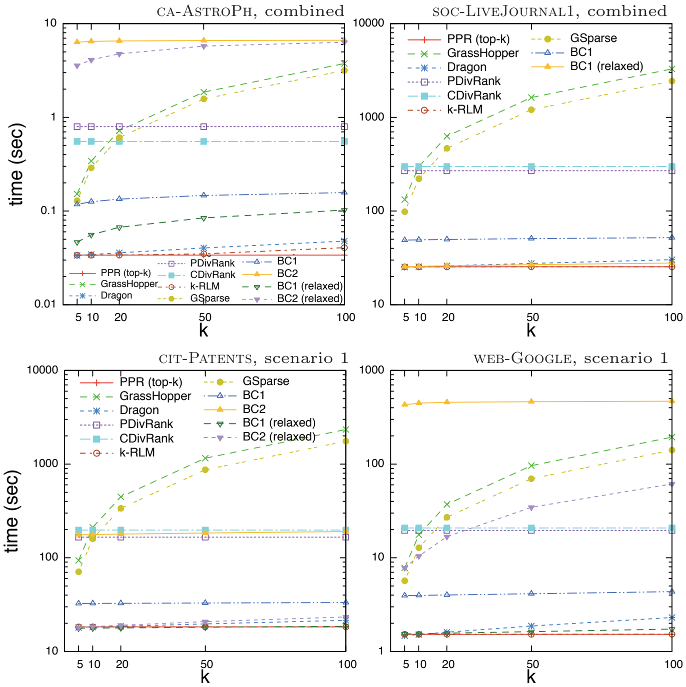
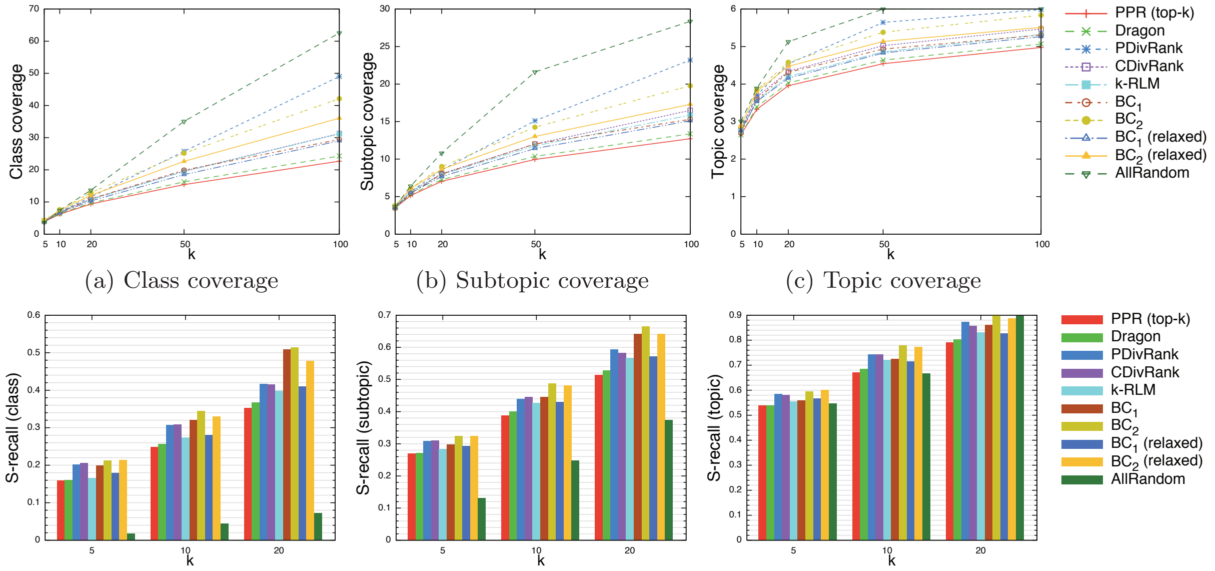


Figure 7: Running times of the algorithms with varying k. BC$_1$ method always perform better with a running time less than GrassHopper and DivRank variants, while the relaxed versions score similarly with a slight overhead on top of the PPR computation.

With the runtime experiments shown in Figure 7, we also confirm that the relaxed variants of BestCoverage perform closer to their originals (see Figure 4) with an order of magnitude or more gain in efficiency. In all cases, even in soc-LiveJournal1, which is the largest dataset in our experiments, the BC$_1$ method always performs better with a running time less than GrassHopper and DivRank variants, while the relaxed version scores closer enough with a running time slightly higher than the original PPR computation. Therefore, in terms of the running times, the efficient algorithms are generally ordered according to PPR $\leq$$k$-RLM $\leq$ BC$_1$(relaxed) $\leq$ Dragon $\leq$ BC$_1$. Confirming the observation in [16], DivRank variants are more efficient than GrassHopper for $k > 10$. Runtime of BC$_2$ depends on the dataset properties while its relaxed variant has comparable running times to DivRank variants. Both BC$_2$ and its variant has a very high runtime on ca-AstroPh since this dataset has the highest average degree ($\bar{\delta} = 42.2$) and the clustering coefficient ($CC = 0.63$), hence, each *exprel*$_2$ computation is more costly than the ones on other datasets.

5.4 Intent-aware results

Among the five datasets we selected for the experiments, cit-Patents has the categorical information. One of the 426 class labels was assigned to each patent, where those classes hierarchically belong to 36 subtopics and 6 high-



(a) Class coverage (b) Subtopic coverage (c) Topic coverage

(d) S-recall on classes (e) S-recall on subtopics (f) S-recall on topics

Figure 8: Intent-aware results on cit-Patents dataset with scenario-3 queries.

level topics[7]. Here we present an evaluation of the intent-oblivious algorithms against intent-aware measures. This evaluation provides a validation of the diversification techniques with an external measure such as *group coverage* [14] and *S-recall* [23].

Intents of a query set $\mathcal{Q}$ is extracted by collecting the classes, subtopics, and topics of each seed node. Since our aim is to evaluate the results based on the coverage of different groups, we only use scenario-3 queries that represent multiple interests.

One measure we are interested in is the *group coverage* as a diversity measure [14]. It computes the number of groups covered by the result set and defined on classes, subtopics, and topics based on the intended level of granularity. However, this measure omits the actual intent of a query, assuming that the intent is given with the classes of the seed nodes.

Subtopic recall (*S-recall*) has been defined as the percentage of relevant subtopics covered by the result set [23]. It has also been redefined as *Intent-Coverage* [25], and used in the experiments of [22]. *S-recall* of a result set S based on the set of intents of the query I is computed with

$$S\text{-}recall(S, I) = \frac{1}{|I|}\sum_{i\in I} B_i(S), \tag{18}$$

where $B_i(S)$ is a binary variable indicating whether intent i is found in the results.

We give the results of group coverage and *S-recall* on classes, subtopics, and topics in Figure 8. The algorithms `GrassHopper` and `GSparse` are not included to the results since they perform worse than `PPR`. The results of `AllRandom` are included to give a comparison between the results of top-k relevant set (`PPR`) and ones chosen randomly.

As the group coverage plots show, top-k ranked items of `PPR` do not have the necessary diversity in the result set, hence, the number of groups that are covered by these items are the lowest of all. On the other hand, a randomized method brings irrelevant items from the search space without considering their relevance to the user query. The results of all of the diversification algorithms reside between those two extremes, where the `PDivRank` covers the most, and `Dragon` covers the least number of groups.

However, *S-recall* index measures whether a covered group was actually useful or not. Obviously, `AllRandom` scores the lowest as it dismisses the actual query (you may omit the *S-recall* on topics since there are only 6 groups in this granularity level). Among the algorithms, `BC`$_2$ variants and `BC`$_1$ score the best while `BC`$_1$ `(relaxed)` and `DivRank` variants have similar *S-recall* scores, even though `BC`$_1$ `(relaxed)` is a much faster algorithm than any `DivRank` variant (see Figure 7).

6. CONCLUSIONS AND FUTURE WORK

In this paper, we address the problem of evaluating result diversification as a bicriteria optimization problem with a relevance measure that ignores diversity, and a diversity measure that ignores relevance to the query. We prove it by running *query-oblivious* algorithms on two commonly used combination of objectives. Next, we argue that a result diversification algorithm should be evaluated under a measure which tightly integrates the query in its value, and presented a new measure called *expanded relevance*. Investigating various quality indices by computing their pairwise correlation, we also show that this new measure has no direct correlation with any other measure. In the second part of the paper, we analyze the complexity of the solution that maximizes the *expanded relevance* of the results, and based on the submodularity property of the objective, we present a greedy algorithm called `BestCoverage`, and its efficient relaxation. We experimentally show that the relaxation carries no significant harm to the *expanded relevance* of the solution.

As a future work, we plan to investigate the behavior of the $exprel_\ell$ measure on social networks with ground-truth communities.

Acknowledgments

This work was supported in parts by the DOE grant DE-FC02-06ER2775 and by the NSF grants CNS-0643969, OCI-0904809, and OCI-0904802.

[7] Available at: `http://data.nber.org/patents/`

7. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proc. ACM Int'l Conf. Web Search and Data Mining*, pages 5–14, 2009.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proc. Int'l Conf. World Wide Web*, pages 107–117, 1998.

[3] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pages 335–336, 1998.

[4] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proc. ACM Conf. Information and Knowledge Management*, pages 621–630, 2009.

[5] X. Cheng, P. Du, J. Guo, X. Zhu, and Y. Chen. Ranking on data manifold with sink points. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):177–191, Jan 2013.

[6] C. L. Clarke, N. Craswell, I. Soboroff, and E. M. Voorhees. Overview of the TREC 2011 Web Track. In *Proc. Text Retrieval Conference (TREC)*, 2011.

[7] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pages 659–666, 2008.

[8] P. Du, J. Guo, J. Zhang, and X. Cheng. Manifold ranking with sink points for update summarization. In *Proc. ACM Conf. Information and Knowledge Management*, pages 1757–1760, 2010.

[9] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *Proc. Int'l Conf. World Wide Web*, pages 381–390, 2009.

[10] T. H. Haveliwala. Topic-sensitive PageRank. In *Proc. Int'l Conf. World Wide Web*, pages 517–526, 2002.

[11] D. S. Hochbaum, editor. *Approximation Algorithms for NP-hard problems*. PWS publishing company, 1997.

[12] O. Kucuktunc and H. Ferhatosmanoglu. λ-diverse nearest neighbors browsing for multidimensional data. *IEEE Transactions on Knowledge and Data Engineering*, 25(3):481 –493, Mar 2013.

[13] O. Kucuktunc, E. Saule, K. Kaya, and U. V. Catalyurek. Diversifying citation recommendation. Technical Report arXiv:1209.5809, ArXiv, Sep 2012.

[14] R.-H. Li and J. X. Yu. Scalable diversified ranking on large graphs. In *Proc. IEEE Int'l Conf. Data Mining*, pages 1152–1157, 2011.

[15] R.-H. Li and J. X. Yu. Scalable diversified ranking on large graphs. *IEEE Transactions on Knowledge and Data Engineering*, PP(99):1, 2012. preprint.

[16] Q. Mei, J. Guo, and D. Radev. DivRank: the interplay of prestige and diversity in information networks. In *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pages 1009–1018, 2010.

[17] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions-I. *Mathematical Programming*, 14(1):265–294, 1978.

[18] R. Pemantle. Vertex-reinforced random walk. *Probab. Theory Related Fields*, 92:117–136, 1992.

[19] F. Radlinski and S. Dumais. Improving personalized web search using result diversification. In *Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pages 691–692, 2006.

[20] H. Tong, J. He, Z. Wen, R. Konuru, and C.-Y. Lin. Diversified ranking on large graphs: an optimization viewpoint. In *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pages 1028–1036, 2011.

[21] M. R. Vieira, H. L. Razente, M. C. Barioni, M. Hadjieleftheriou, D. Srivastava, C. Traina, and V. J. Tsotras. On query result diversification. In *Proc. IEEE Int'l Conf. Data Engineering*, pages 1163 –1174, 2011.

[22] M. J. Welch, J. Cho, and C. Olston. Search result diversity for informational queries. In *Proc. Int'l Conf. World Wide Web*, pages 237–246, 2011.

[23] C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pages 10–17, 2003.

[24] X. Zhu, A. B. Goldberg, J. V. Gael, and D. Andrzejewski. Improving diversity in ranking using absorbing random walks. In *Proc. HLT-NAACL*, pages 97–104, 2007.

[25] X. Zhu, J. Guo, X. Cheng, P. Du, and H.-W. Shen. A unified framework for recommending diverse and relevant queries. In *Proc. Int'l Conf. World Wide Web*, pages 37–46, 2011.