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, π, ℓ
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, π, ℓ
Output: a list of recommendations S
$S = \emptyset$
SORT(V) w.r.t π_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.

[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. 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	$\|V\|$	$\|E\|$	$\bar{\delta}$	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.

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

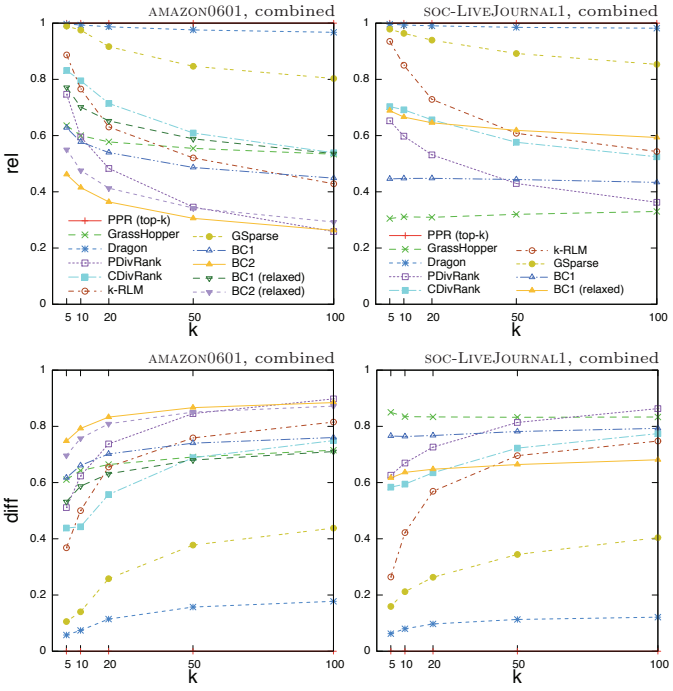


Figure 2: Normalized relevance (*rel*) and different ratio (*diff*) scores with varying k. `Dragon` and `GSparse` return results around 70% similar to the top-k relevant set, this is generally not enough to improve the diversity of the results.

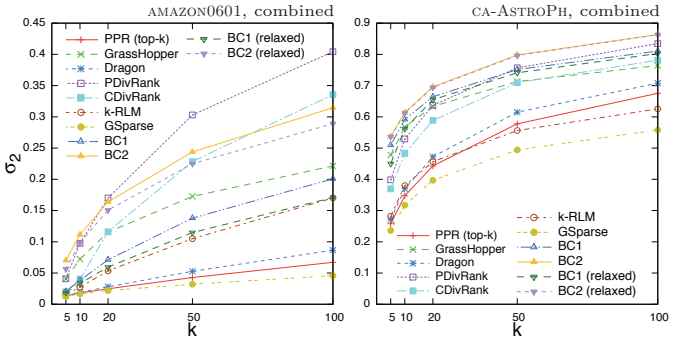


Figure 3: Coverage (σ_2) of the algorithms with varying k. `BestCoverage` and `DivRank` variants have the highest coverage on the graphs while `Dragon`, `GSparse`, and k-`RLM` have similar coverages to top-k results.

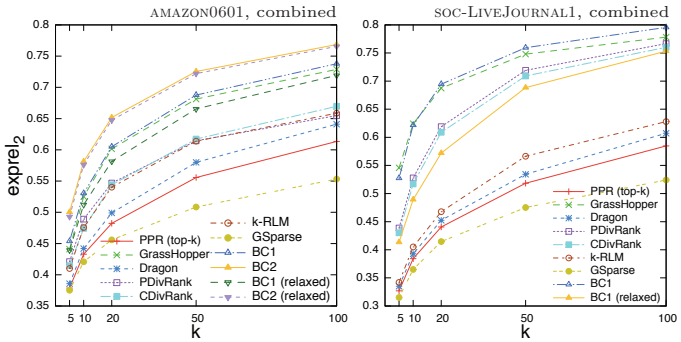


Figure 4: Expanded relevance ($exprel_2$) with varying k. BC_1 and BC_2 variants mostly score the best, `GrassHopper` performs high in soc-LiveJournal1. Although `PDivRank` gives the highest coverage on amazon0601 (Fig. 3), it fails to cover the relevant parts.

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 $rel \to 0$); 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.

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`

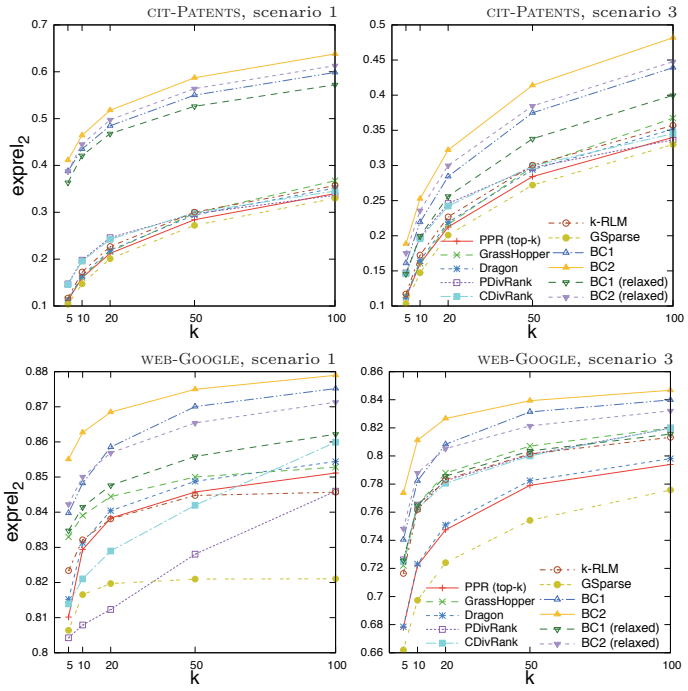


Figure 5: Expanded relevance ($exprel_2$) with varying k. BestCoverage variants perform higher than usual on cit-Patents dataset with scenario-1 queries because of the low average degree ($\bar{\delta} = 8.7$) and low clustering coefficient ($CC = 0.09$) of the graph. The relaxed algorithms perform closer to their originals, meaning that they were both efficient and effective on this type of sparsely connected graphs.

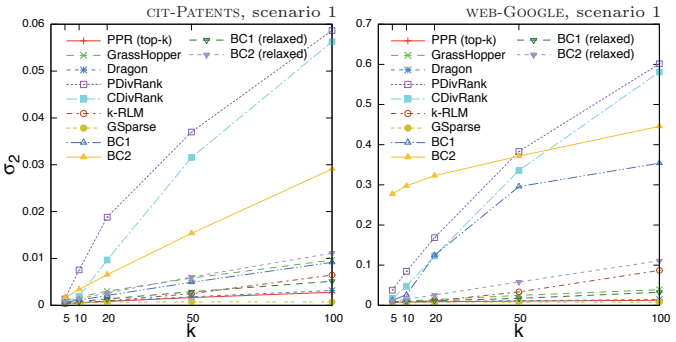


Figure 6: Coverage (σ_2) of the algorithms with varying k. DivRank variants appear to be implicitly optimizing the *size* of the expansion set, without considering whether those results are still relevant to the query (cf. corresponding $exprel_2$ in Figure 5).

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 $\texttt{BC}_1$ and $\texttt{BC}_2$ perform closer to $\texttt{BC}_1$ and $\texttt{BC}_2$, 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

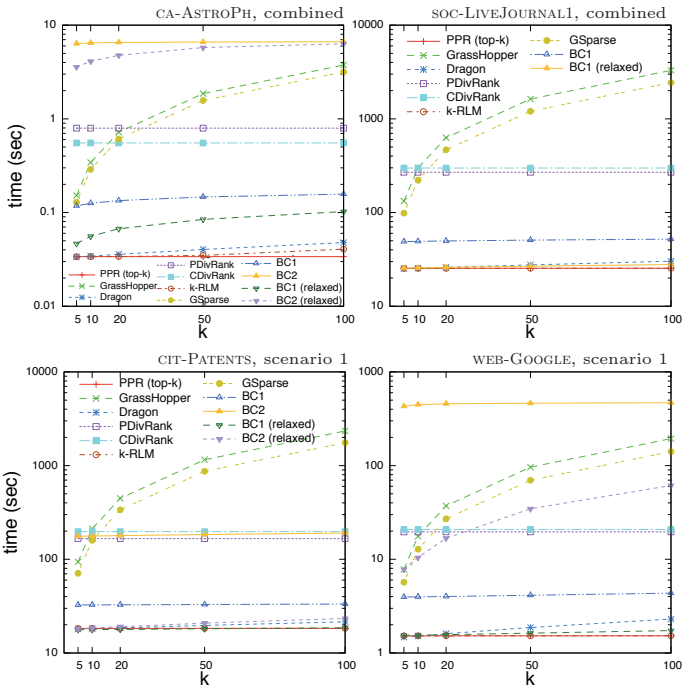


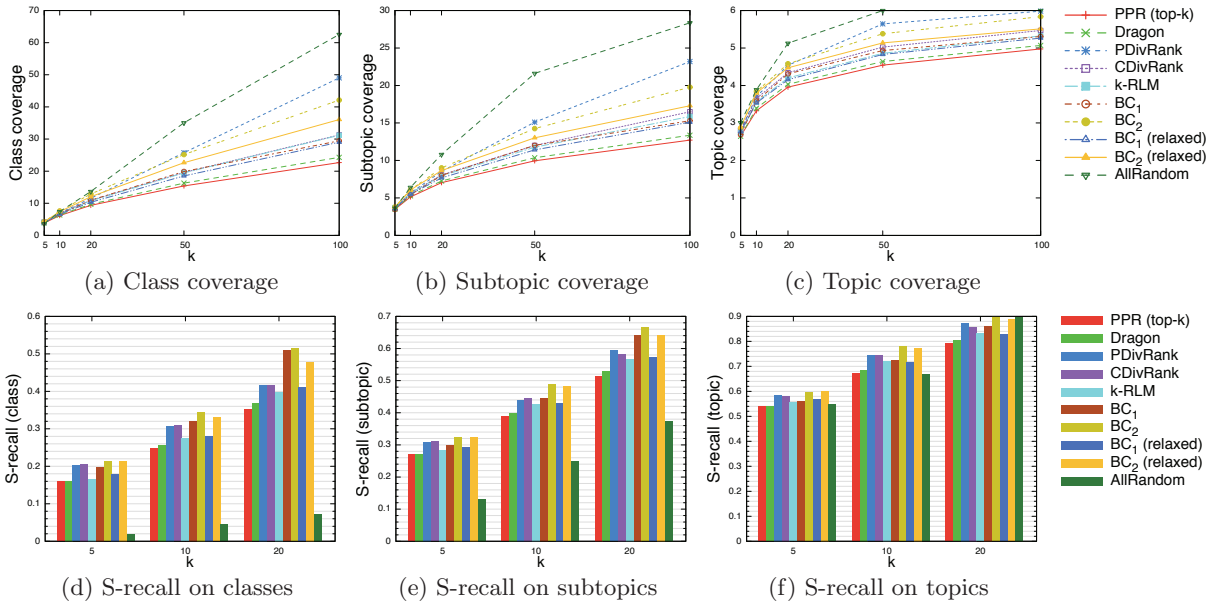
Figure 7: Running times of the algorithms with varying k. $\texttt{BC}_1$ method always perform better with a running time less than GrassHopper and DivRank variants, while the relaxed versions score similarly with a slight overhead on top of the PPR computation.

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).

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 $\texttt{BC}_1$ 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 $\leq \texttt{BC}_1\texttt{(relaxed)} \leq$ Dragon $\leq \texttt{BC}_1$. Confirming the observation in [16], DivRank variants are more efficient than GrassHopper for $k > 10$. Runtime of $\texttt{BC}_2$ depends on the dataset properties while its relaxed variant has comparable running times to DivRank variants. Both $\texttt{BC}_2$ 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

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

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, $\texttt{BC}_2$ variants and $\texttt{BC}_1$ score the best while $\texttt{BC}_1$ `(relaxed)` and `DivRank` variants have similar *S-recall* scores, even though $\texttt{BC}_1$ `(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.