Diversity Maximization Under Matroid Constraints

Zeinab Abbassi
Department of Computer Science
Columbia University
zeinab@cs.olumbia.edu

Vahab S. Mirrokni
Google Research, New York
mirrokni@google.com

Mayur Thakur
Google New York
mayurthakur@google.com

ABSTRACT

Aggregator websites typically present documents in the form of representative clusters. In order for users to get a broader perspective, it is important to deliver a *diversified* set of representative documents in those clusters. One approach to diversification is to maximize the average dissimilarity among documents. Another way to capture diversity is to avoid showing several documents from the same *category* (e.g. from the same news channel). We model the latter approach as a (partition) matroid constraint, and study diversity maximization problems under matroid constraints. We present the *first* constant-factor approximation algorithm for this problem, using a new technique. Our local search 0.5-approximation algorithm is also the *first* constant-factor approximation for the max-dispersion problem under matroid constraints. Our combinatorial proof technique for maximizing diversity under matroid constraints uses the existence of a family of Latin squares which may also be of independent interest.

In order to apply these diversity maximization algorithms in the context of aggregator websites and as a preprocessing step for our diversity maximization tool, we develop greedy clustering algorithms that maximize weighted coverage of a predefined set of topics. Our algorithms are based on computing a set of cluster centers, where clusters are formed around them. We show the better performance of our algorithms for diversity and coverage maximization by running experiments on real (Twitter) and synthetic data in the context of real-time search over micro-posts. Finally we perform a user study validating our algorithms and diversity metrics.

1. INTRODUCTION

Aggregator websites such as news aggregators have become increasingly common during the past decade. These websites (e.g., Google News) typically present snippets of documents (e.g., news articles) in the form of clusters. In these clusters, related documents are aggregated and a few of the representative items in that cluster are highlighted to give a glimpse of the documents in that cluster. In order for users to get a broader perspective on different topics, it is important to deliver a *diversified* set of representative documents in those clusters. Presenting a diversified set of documents is also important in the context of commerce search [2]. There are two ways to capture diversity among documents:

- In the first approach, after defining a *distance function* ℓ among documents based on some implicit metrics (e.g. their content), we aim to to maximize the average pairwise distance among these documents [9]. This measure has been motivated and studied in the context of diversity maximization in recommender systems and commerce search [9, 2]. In particular, Bhattacharya, Gollapudi, and Munagala [2] argue that maximizing this diversity measure is desirable in the context of commerce (or product) search. They validate this claim through a user study. From an algorithmic point of view, this problem is known as the maximum dispersion problem and approximation algorithms are known for it [10, 9, 2].

- Another approach to diversification is based on considering explicitly defined *categories* of documents such as news channels or product brands. This approach enforces a constraint that no more than one (or two) of the documents from the same category may appear in the output. This more explicit diversity constraint can be modeled using a (partition) matroid constraint, i.e., we can define a (partition) matroid over the set of documents and aim to find a subset of representative documents that form an independent set of that matroid.

In order to model the above ways to capture diversity, we define the diversity maximization problem under matroid constraints: assuming that we can show a limited number of representative documents in the output, the goal of the diversity maximization problem under matroid constraints is to choose a subset of a small number of representative documents with the maximum diversity such that they form an independent set of a matroid constraint (e.g., not more than 1 (or 2) of the documents belong to the same category). [1] We study this problem from an algorithmic perspective as well as experimentally using simulations and a user study.

1.1 Our Contributions

Diversity maximization under matroid constraints. As our main technical contribution, we present the *first* constant-factor approximation algorithm for the diversity maximization problem under matroid constraints using a new local search technique. Previous approximation algorithms for diversity maximization (or maximum dispersion) are based on a greedy approach that does not

[1] More generally, we consider the problem of choosing a set of diversified representative documents inside a given set of (possibly overlapping) clusters, and study the *clustered diversity maximization* problem under matroid constraints (See Section 2 for more details).

handle the matroid constraints [10, 2]. On the other hand, the local search technique is appropriate for handling such matroid constraints. Moreover, our algorithm can handle an extra constraint to find representative documents inside a given set of (possibly overlapping) clusters. Our local search 0.5-approximation algorithm is also the first constant-factor approximation for the max-dispersion problem under matroid constraints [10, 9, 2]. Our combinatorial proof technique for maximizing diversity under matroid constraints uses the existence of a family of Latin squares which may find other theoretical applications.

Preprocessing for Clustering. As part of the input to the diversity maximization problem discussed above, we need to pre-compute a set of clusters $(C_1, \ldots, C_k)$. Such a set of clusters can be found using many different techniques. While this is not the focus of this paper, we discuss two simple algorithms to produce these clusters. These algorithms will be used as the preprocessing step of the diversity maximization problem and they will be evaluated in the experimental evaluation section. Both of these algorithms follow the following framework: first compute a subset $S = \{d_1, \ldots, d_k\}$ of top k center documents, and then build a cluster C_i around each center d_i. In other words, we think of each document in this set as a center document for a cluster around it. In order to find these centers, we associate a set of topics to each document, and in computing the set of centers, we aim to cover a large range of topics. To do so, we define a weighted coverage function, and aim to maximize a weighted coverage of the documents in S. The weighted coverage of a subset S is defined as the sum of the weighted coverage of this set for each topic, where the *weighted-coverage* of S for each topic t is the maximum extent to which any document in S covers topic t. Although this weighted coverage problem is different from the previously studied max k-coverage problems [8], we show that it is a submodular set function and thus a simple greedy algorithm gives a $1 - \frac{1}{e}$-approximation to maximize this function. This greedy algorithm generates k documents $d_1, d_2, \ldots, d_k$ by iteratively picking a new document that covers the maximum marginal weighted-coverage. After identifying these k documents, we can generate k clusters $C_1, C_2, \ldots, C_k$ by putting documents that are similar to center d_i in cluster C_i. In the next part, we aim to identify representative documents in each cluster C_i by solving a diversity maximization problem.

Experimental Evaluation. We show the effectiveness of our algorithms in practice through experiments and a user study. We perform experiments on both real and randomly generated data to confirm the performance of our algorithms for maximizing diversity and weighted-coverage.

Data. The algorithms developed in this paper can be applied to different applications of aggregator websites such as news aggregators (like Google News), commerce search [2], and real-time search for online social networks. As for the *real dataset*, we focus on the application of developing an aggregator website in the context of real-time search for the stream of Twitter micro-posts. This application is motivated by fast data generation on online social media sites such as Twitter, Facebook, and Google+ which calls for a quick way of summarizing hot real-time trends in a collection of micro-posts. We extract candidate documents and important topics through careful mining of recent popular queries and their topics. We compute a hotness score for each topic, and also a relevance measure for each micro-post covering a topic (See Section 5 for details). This information can be used to produce a clustering of important micro-posts covering hot topics in a micro-blog environment, and then compute a small number of diversified micro-posts to represent those clusters. We compare the performance of the local search algorithm, a greedy algorithm, and a naive sorting algorithm on these datasets.

Observations. For several of the randomly generated datasets, the diversity of the local search algorithm is more than 300% higher than that of the greedy or the naive sorting algorithm. For the real data set, we observe that the greedy algorithm and the local search algorithm consistently outperform the sorting algorithm for maximizing weighted-coverage and diversity respectively: For example, we observe that on average, weighted-coverage of the greedy algorithm is 78% higher than that of the sorting algorithm, and the local search algorithm achieves 28% higher diversity than the sorting algorithm.

User Study. Finally, we conduct a user study through Amazon Mechanical Turk validating our algorithms and our metrics. This user study shows that most users prefer to see a diversified set of documents across clusters in an aggregator website, however, most users prefer a less diversified set of representatives inside the clusters.

Organization. The rest of this paper is organized as follows. After discussing related work, in Section 2, we define our model, and present formal definitions of the (clustered) diversity maximization problems under matroid constraints. In Section 3, we present our main technical contribution, i.e., a local search 0.5-approximation algorithm to maximize diversity subject to matroid constraints. In Section 3.1, we define a variant of the problem generalizing the maximum dispersion problem and observe the first constant-factor approximation algorithm for the max-dispersion problem under matroid constraints. Then in Section 4, we present the preprocessing step of computing a set of clusters by choosing a number of cluster centers maximizing their weighted-coverage. In particular, in subsection 4.1, we present a greedy algorithm to compute the cluster centers maximizing the weighted-coverage, and and show that this objective function is submodular, implying a constant approximation factor of the greedy algorithm for the coverage maximization problem. In Section 5, we present our experimental results, showing the performance of our algorithms on real and randomly generated data. Finally, we present our user study validating our metrics and algorithms in Section 6.

1.2 More Related Work.

Diversity Maximization Problem. The diversity maximization problem studied in this paper generalizes the maximum dispersion problem [10, 9, 2]. This problem has been explored in the context of diversity maximization for recommender systems [9], and commerce search [2]. A $1/2$-approximation greedy algorithm has been developed for the unconstrained variant of this problem [10], and the variant with knapsack constraints [2]. None of these greedy algorithms gives a constant-factor approximation for this problem under matroid constraints. By developing a local search algorithm and a different proof technique, we get the first $1/2$-approximation algorithms for the maximum dispersion problem under matroid constraints. Such matroid constraints are also natural in the context of product search and can be directly applied to similar problems studied in [2].

Diversity in Recommender Systems and Web Search Ranking and relevance maximization along with diversification have been extensively studied in recommender systems, web search, and database systems. While these results differ from our work in various aspects, we point out some related work in this extensive literature which implies the importance of diversification in these applications.

In the context of web search, maximizing diversity has been explored as a post-processing step [3, 24]. Other papers explore ranking while taking into account diversity by a query reformulation for re-ranking the top searches [19] or by sampling the search results by reducing homogeneity [1]. Other methods are based on clustering results into groups of related topics [15], or expectation maximization for estimating the model parameters and reaching an equilibrium [20]. Moreover, in the context of recommender systems, diversification has been explored in various recent papers [26, 17, 13, 25]. For example, topical diversity maximization is disscused in [26], and explanation-based diversity maximization is explored in [25]. From the information retrieval perspective, more realtime diversification ranking methods have been developed exploiting user browsing behaviors [7] or exploiting query reformulations for web search results [21]. This topic has been also explored in database systems [6, 14]. For example, notions of diversity has been suggested based on presenting decision trees to users [6]. Also computing diverse query results for online shopping has been studied in [23]. Most of the above work consider other types of diversity metrics, and to the best of our knowledge, our diversity maximization problem subject to matroid constraints has not been explored formally prior to our results.

2. PRELIMINARIES

2.1 Diversity Maximization Problems

Documents and Distance Function. Our model consists of a set of documents D and pairwise distance function $\ell : D \times D \to R$ capturing the dissimilarity among documents. Throughout this paper, we assume that the distance function ℓ is a metric satisfying the triangle inequality, i.e., for any three documents, $\ell(d_1, d_3) \leq \ell(d_1, d_2) + \ell(d_2, d_3)$. In our main application, we will use the weighted Jaccard Distance as the pairwise dissimilarity between documents which is a metric. For more details on this, see Section 2.2. The documents $d_i \in D$ may correspond to various objects in different applications: in the context of a news aggregator, these documents correspond to news articles; for product search, the documents represent descriptions of a product, and for real-time search, they correspond to online (micro-)posts by users.

Diversity Function. Our main focus in this paper is to find a small set of representative documents maximizing *diversity*. Given a set of documents $S \subseteq D$, the diversity of documents in S is defined as the sum of pairwise distances between documents in S, i.e.,

$$\text{diversity}(S) = \sum_{i \in S} \sum_{j \in S} \ell(i, j).$$

This diversity measure has been motivated and studied in the context of recommender systems and commerce search [9, 2]. For example, Bhattacharya, Gollapudi, and Munagala [2] argue that maximizing this diversity measure is desirable in the context to commerce search, and validate this claim through a user study. The simplest version of the diversity maximization problem is to choose a set S of k documents with maximum diversity. Below, we discuss more general variants of this problem in the presence of a (possibly overlapping) clustering of document and also under matroid constraints.

Clustered Diversity Maximization. Given a set of clusters $\mathcal{C} = (C_1, C_2, \ldots, C_k)$, find a set S of p representative documents $r^i_1, r^i_2, \ldots, r^i_p$ in each cluster C_i maximizing diversity(S).

Note that in the definition of the problem above, clusters $C_1, \ldots, C_k$ need not to be disjoint, and can be arbitrary overlapping subsets of nodes. In particular, each cluster may include the whole set of documents, i.e, $C_i = D$. For this special case, the diversity maximization algorithm is to choose a set of pk nodes in the graph to maximize diversity (independent of their coverage). Since the distance function ℓ satisfies triangle inequality [4], this special case of the clustered diversity maximization problem is equivalent to the maximum dispersion problem [10].

Motivation for Matroids. In order to identify diversified representative documents, a natural requirement is to retrieve documents from different categories. In settings that such categories are explicitly given, this constraint might be enforced in addition to maximizing the diversity function defined above. One example of such constraints is as follows: Consider a partitioning of documents D into q subcategories $D_1, D_2, \ldots, D_q$, where $D = \cup_{1 \leq i \leq q} D_i$ and D_is are disjoint sub-categories of documents. For example, in the context of news aggregators, each category D_i may correspond to a news domain; for product search, the subcategory D_i may correspond to a specific brand, and in the context of real-time search, a subcategory may correspond to online posts from the same user. We would like to find a subset of documents maximizing total diversity such that at most one document from each category D_i is present in the whole set (or each given cluster C_i). The constraint of not having more than one document from each subcategory can be captured as a special matroid constraint, called *the partition matroid constraint*. Here, we first define a matroid constraint, then define the diversity problem under a general set of matroid constraints.

Matroid Constraints. A matroid $\mathcal{M}$ is defined as a family of subsets of the ground set of documents $\mathcal{E}(\mathcal{M}) = D$, called independent sets. The set of independent sets S of a matroid $\mathcal{M}$ is denoted by $\mathcal{I}(\mathcal{M})$. For a given matroid $\mathcal{M}$, the associated *matroid constraint* is $S \in \mathcal{I}(\mathcal{M})$. As is standard, $\mathcal{M}$ is a *uniform matroid* of rank r if $\mathcal{I}(\mathcal{M}) := \{X \subseteq \mathcal{E}(\mathcal{M}) : |X| \leq r\}$. A *partition matroid* is the direct sum of uniform matroids. Note that uniform matroid constraints are equivalent to cardinality constraints, i.e, $|S| \leq k$. For more details about matroids, see [22].

Now, we formally define the problem:

Clustered Diversity Maximization Under Matroid Constraints. Let $\mathcal{M}$ be a matroid over the set of documents D with a family of independent sets $\mathcal{I}(\mathcal{M})$. Given a set of clusters $\mathcal{C} = (C_1, C_2, \ldots, C_k)$, our goal is to find an independent set $R_i \in \mathcal{I}(\mathcal{M})$ of p representative documents $R_i = \{r^i_1, r^i_2, \ldots, r^i_p\} \subseteq C_i$ $(1 \leq i \leq k)$ while maximizing diversity$(\cup_{1 \leq i \leq k} R_i)$.

A special class of matroids is a partition matroid $\mathcal{M}$: Given a partitioning of documents D to q subsets $(D_1, D_2, \ldots, D_q)$, a subset $S \subset D$ is an independent set of partition matroid $\mathcal{M}$ iff S has at most one document from each subset D_j for each $1 \leq j \leq q$. As discussed earlier, our main motivation for studying matroid constraints is partition matroids.

2.2 Topics and Distance Function

As discussed earlier, the distance function ℓ between documents for our main application is defined based on the generalized Jaccard distance function. In this section, we define the Jaccard distance function and other preliminaries related to the distance function.

Topics. In order to define the distance function between documents, we consider a set of topics T. The topics $t \in T$ correspond to important themes or subjects of those documents, e.g. they may correspond to hot news or micro-post topics for news aggregators or micropost real-time search; or to their brand and their features for the product search application.

Pairwise relevance and edge weights $w(d, t)$. For each document $d \in D$ and any topic $t \in T$, the weight $w(d, t)$ models the relevance of a topic t to a document d, i.e., it represents the extent to which a document is related to a topic $t \in T$. The larger the weight $w(d, t)$ is, the more relevant document d is to topic t. Throughout the this paper, we assume that the weights are given and computed in advance [2] For the theoretical part we assume that the weights are given.

We construct an edge-weighted bipartite topic-document bipartite graph $G(D, T, E)$ between documents and topics with edge weights $w(d, t)$. Our goal is find a set of clusters with a set of representative documents in each cluster covering a large portion of topics while maintaining diversity. A central property that we need to satisfy in this paper is matroid constraints.

Generalized Jaccard Distance. To further investigate the diversity of a set of documents, we define the following generalized Jaccard distance between documents: Given an edge-weighted bipartite graph $G(D, T, E)$, for two documents d_1 and d_2, the *generalized Jaccard distance* between d_1 and d_2 is defined as:

$$\ell(d_1, d_2) = 1 - \frac{\sum_{t\in T} \min(w(d_1, t), w(d_2, t))}{\sum_{t\in T} \max(w(d_1, t), w(d_2, t))}.$$

This distance function is a natural generalization of the Jaccard distance functions over sets with weighted elements. This generalized Jaccard distance has been studied already and it has been shown that it satisfies the triangle inequality [4], i.e, for any three documents, $\ell(d_1, d_3) \leq \ell(d_1, d_2) + \ell(d_2, d3)$.

3. DIVERSITY MAXIMIZATION UNDER MATROID CONSTRAINTS

Here, we present a local search algorithm to choose a set of diversified representatives in each cluster under a (partition) matroid constraint $\mathcal{M}$. We prove that it achieves an approximation ratio of $\frac{1}{2}$ in the worst case. Roughly speaking, the algorithm starts from an arbitrary set R_i of representative documents in each cluster, and at each step, it considers all pairs $(d, d') \in R_i \times C_i \backslash R_i$ of documents inside and outside of the representatives such that $R_i \backslash \{d\} \cup \{d'\} \in \mathcal{I}(\mathcal{M})$, and examines swapping these two documents, i.e, removing document d and adding document d' to the set of representatives. If this swap increases the total diversity of the documents by a factor of $1 + \frac{\epsilon}{n}$, i.e, if it increases the sum of pairwise distances of representatives by that factor, we make this swap, i.e, we let $R_i = R_i \backslash \{d\} \cup \{d'\}$.

Local search algorithm to choose diversified representatives

```
Input: A set of clusters $\mathcal{C} = (C_1, C_2, \ldots, C_k)$ over
a set of documents $D$, a matroid $\mathcal{M}$ over documents, and
and a distance function $\ell$ among documents.
Output: For each cluster $C_i$ ($1 \leq i \leq k$), a set of $p$ documents
$R_i = \{r^i_1, r^i_2, \cdots, r^i_p\} \in \mathcal{I}(\mathcal{M})$ where $R_i \subseteq C_i$.
Goal: Find a set of representatives maximizing total
diversity, i.e., $\sum_{1\leq i\leq k, 1\leq j\leq p} \sum_{1\leq i'\leq k, 1\leq j'\leq p} \ell(r^i_j, r^{i'}_{j'})$.
Initialize: Let $R_i$ be a set of top documents in $C_i$.
While there exists a local improvement do
    For any pair of documents $(d, d') \in R_i \times C_i \backslash R_i$
        such that $R_i \backslash \{d\} \cup \{d'\} \in \mathcal{I}(\mathcal{M})$ do
        If removing $d$ from $R_i$ and adding $d'$ to $R_i$ increases
        the total diversity by a factor of $1 + \frac{\epsilon}{n}$, i.e.,
        if $g(S \backslash \{d\} \cup \{d'\}) > (1 + \frac{\epsilon}{n}) g(S)$ then
            Let $R_i := R_i \backslash \{d\} \cup \{d'\}$.
```

One desirable property of this local search algorithm is its flexibility to optimize other objective functions in the case where there are more than one choice for local improvements. Examples of these objective functions are quality or coverage of the documents or the popularity of the news channels. To optimize a different objective function, one can try the sequence of local operations in the order of that objective, for example, we initialize the representatives to the set of documents with the highest quality score, and consider swapping other documents with these documents in the order of their quality score. This quality score may take into account the reliability and coverage of each document, or the popularity of owner of the micro-post. We first prove that the algorithm achieves a good guaranteed approximation ratio, and then we study its computational time complexity.

Approximation Factor. Here, we show that the exact local search algorithm achieves a guaranteed approximation ratio of $1/2$. Such a proof for the exact local search algorithm simply implies a guaranteed approximation of $0.5 - \frac{\epsilon}{n}$ for the approximate local search algorithm above. This proof is based on the existence a class of diagonal Latin squares, and thus this technique may find other theoretical applications.

THEOREM 3.1. *Given a set of clusters $\mathcal{C} = (C_1, C_2, \ldots, C_k)$, the above local search algorithm is a $1/2$-approximation algorithm for the problem of choosing a set of p representatives $R_i \in \mathcal{I}(\mathcal{M})$ for each cluster C_i, maximizing the total diversity of the representatives.*

PROOF. Consider an optimum solution $\mathcal{O} = \{O_1, \ldots, O_k\}$ where $O_i = \{O^i_1, \ldots, O^i_p\} \in \mathcal{I}(\mathcal{M})$. Let the output of the local search algorithm be $\mathcal{L} = \{L_1, L_2, \ldots, L_k\}$ where $L_i = \{L^i_1, \ldots, L^i_p\} \in \mathcal{I}(\mathcal{M})$. Our goal is to show that

$$\text{L} = \sum_{1\leq i\leq k, 1\leq j\leq p} \sum_{1\leq i'\leq k, 1\leq j'\leq p} \ell(L^i_j, L^{i'}_{j'}) \quad \geq$$

$$\frac{1}{2} \sum_{1\leq i\leq k, 1\leq j\leq p} \sum_{1\leq i'\leq k, 1\leq j'\leq p} \ell(O^i_j, O^{i'}_{j'}) \quad = \tfrac{1}{2}\text{OPT}$$

To prove this, we need to employ a useful exchange property of matroids (see [22]). Intuitively, this property states that for any two independent sets I and J, we can add any element of J to the set I, and kick out at most one element from I while keeping the set independent. Moreover, each element of I is allowed to be kicked out by at most one element of J.

PROPOSITION 3.2. *[[22]] Let $\mathcal{M}$ be a matroid and $I, J \in \mathcal{I}(\mathcal{M})$ be two independent sets. Then there is a mapping $\pi : J \setminus I \to (I \setminus J) \cup \{\phi\}$ such that:*

1. $(I \setminus \pi(b)) \cup \{b\} \in \mathcal{I}(\mathcal{M})$ *for all* $b \in J \setminus I$.

2. $|\pi^{-1}(e)| \leq 1$ *for all* $e \in I \setminus J$.

Applying Proposition 3.2, we may consider a mapping π_i between each subset L_i and O_i such that $L_i \backslash L^i_j \cup O^i_{\pi_i(j)} \in \mathcal{I}(\mathcal{M})$. Without loss of generality, we may assume $\pi_i(L^i_j) = O^i_j$, and thus from Proposition 3.2, $L_i \backslash L^i_j \cup O^i_j \in \mathcal{I}(\mathcal{M})$. Now consider removing L^i_j from L_j and adding O^i_j to L_j. Using the local optimality of $\mathcal{L}$ and since $L_i \backslash L^i_j \cup O^i_j \in \mathcal{I}(\mathcal{M})$, we know that for any $1 \leq i \leq k$ and $1 \leq j \leq p$:

$$\sum_{1\leq i'\leq k, 1\leq j'\leq p} \ell(L^i_j, L^{i'}_{j'}) \geq \sum_{1\leq i'\leq k, 1\leq j'\leq p} \ell(O^i_j, L^{i'}_{j'})$$

Now, adding the above inequality for all $1 \leq i \leq k$ and $1 \leq j \leq$

[2] These weights can be computed in different ways for different applications. In Section 5, we describe one specific way to compute those weights for the real-time application.

p, we get:

$$\begin{aligned} \mathrm{L} = \sum_{1\leq i\leq k, 1\leq j\leq p} \sum_{1\leq i'\leq k, 1\leq j'\leq p} \ell(L_j^i, L_{j'}^{i'}) \quad &\geq \\ \sum_{1\leq i\leq k, 1\leq j\leq p} \sum_{1\leq i'\leq k, 1\leq j'\leq p} \ell(O_j^i, L_{j'}^{i'}) & \end{aligned}$$

For ease of notation, let $o_i = O^{\lfloor \frac{i}{k} \rfloor + 1}_{(i - k\lfloor \frac{i}{k} \rfloor)+1}$, i.e, for $n = kp$, and let $o_0, o_2, \ldots, o_{n-1}$ correspond to the set of all representatives in the optimal solution O_j^i's for $1 \leq i \leq k$ and $1 \leq j \leq p$. Similarly, let $l_0, l_2, \ldots, l_{n-1}$ correspond to the set of representatives in the local optimal solution L_j^i. In this new notation, we can re-write the above inequalities as:

$$\begin{aligned} \mathrm{L} = \sum_{0\leq i\leq n-1} \sum_{0\leq j\leq n-1} \ell(l_i, l_j) \quad &\geq \\ \sum_{0\leq i\leq n-1} \sum_{0\leq j\leq n-1} \ell(o_i, l_j) & \end{aligned}$$

In order rewrite the above inequalities, we use a pattern similar to a family of Latin squares. Consider an $n \times n$ diagonal Latin square, with elements a_{ij} for $0 \leq i \leq n-1$ and $0 \leq j \leq n-1$. This Latin square has the following properties (that are useful later to finish the proof): In each row and column of Latin square, each number $0, 1, 2, \ldots, n-1$ appears exactly once, and $a_{ii} = i$. The existence of such Latin squares have been already proved [12]. Therefore, the summation $\sum_{0\leq i\leq n-1} \sum_{0\leq j\leq n-1} \ell(o_i, l_j)$ appeared in the above formula may be rewritten as

$$\frac{1}{2} \sum_{0\leq i\leq n-1} \sum_{0\leq j\leq n-1} \ell(o_i, l_{a_{ij}}) + \ell(o_j, l_{a_{ij}}).$$

Now consider the following set of triangle inequalities among the documents: $\ell(o_i, l_{a_{ij}}) + \ell(o_j, l_{a_{ij}}) \geq \ell(o_i, o_j)$. Using the structure of the Latin square, and by putting it all together, we get:

$$\begin{aligned} \mathrm{L} = \sum_{1\leq i\leq k, 1\leq j\leq p} \sum_{1\leq i'\leq k, 1\leq j'\leq p} \ell(L_j^i, L_{j'}^{i'}) \quad &\geq \\ \sum_{1\leq i\leq k, 1\leq j\leq p} \sum_{1\leq i'\leq k, 1\leq j'\leq p} \ell(O_j^i, L_{j'}^{i'}) \quad &= \\ \frac{1}{2} \sum_{0\leq i\leq n-1} \sum_{0\leq j\leq n-1} \ell(o_i, l_{a_{ij}}) + \ell(o_j, l_{a_{ij}}) \quad &\geq \\ \frac{1}{2} \sum_{0\leq i\leq n-1} \sum_{0\leq j\leq n-1} \ell(o_i, o_j) \quad &= \\ \frac{1}{2} \sum_{1\leq i\leq k, 1\leq j\leq p} \sum_{1\leq i'\leq k, 1\leq j'\leq p} \ell(O_j^i, O_{j'}^{i'}) = \quad & \tfrac{1}{2}\mathrm{OPT} \end{aligned}$$

This completes the proof. □

Running time of the local search algorithm. It can be seen that this algorithm converges as the value of the metric increases at each step. In fact, this algorithm has a polynomial running time, since a swap only is done if the metric increases by a factor of $1 + \frac{\epsilon}{n}$ for an appropriately small constant $\epsilon > 0$. To show this, first we observe that by starting from a set S including two furthest documents, we have $g(S) \geq \frac{\mathrm{OPT}}{k^2}$ for the initial set S. Starting from such a set, and using the *approximate* local improvements, it follows that if the algorithm performs t local improvements, then $(1 + \frac{\epsilon}{n})^t \leq k^2$ since at each step, the metric increases by $1 + \frac{\epsilon}{n}$ factor and the value of the metric is upper bounded by k^2 times the initial value of the metric. It follows that $t \leq O(\log_{1+\frac{\epsilon}{n}}(k^2)) = O(\frac{n}{\epsilon}\log(k))$. Therefore the algorithm finishes at most after $O(\frac{n}{\epsilon}\log(k))$ local improvements, and thus it runs in polynomial time.

3.1 Diversity Maximization Under Global Matroid Constraints

As another variant of the diversity maximization problem, one can consider the matroid constraint over all representatives (instead of having the matroid constraint for the representatives in each cluster). In this variant, we find a set of p representatives $R_i \subseteq C_i$ such that the whole set $\cup_{1\leq i\leq k} R_i$ is an independent set in the matroid. For example, we may enforce the constraint that at most one representative in $\cup_{1\leq i\leq k} R_i$ can be chosen from each D_j for $1 \leq j \leq q$. The significance of this variant is that it generalizes the max-dispersion problem under matroid constraints. To see this, consider this problem with all equal clusters $C_i = D$ and $p = 1$. Therefore, a $\frac{1}{2}$-approximation algorithm for this problem implies a $\frac{1}{2}$-approximation algorithm for the max-dispersion problem with matroid constraints. In the following, we formally define this variant.

Clustered Diversity Maximization Under Global Matroid Constraints. Let $\mathcal{M}$ be a matroid over the set of documents D and family of independent sets $\mathcal{I}(\mathcal{M})$. Our goal is to find an independent set $R_i \subset C_i$ of p representative documents $R_i = \{r_1^i, r_2^i, \ldots, r_p^i\}$ for each cluster C_i $(1 \leq i \leq k)$ such that $\cup_{1\leq i\leq k} R_i \in \mathcal{I}(\mathcal{M})$, and maximize diversity$(\cup_{1\leq i\leq k} R_i)$.

The local search algorithm can be easily modified to solve this new variant. To see this, consider a local search algorithm in which at each step, we only allow swap operations that keep the whole set of representatives an independent set of the matroid. One can easily check that the above proof can be adapted to show that such a local search algorithm gives a $\frac{1}{2} - \epsilon$-approximation algorithm. This, in turn, implies the first constant-factor approximation for the maximum dispersion problem under matroid constraints.

4. PREPROCESSING FOR CLUSTERING

As part of the input to the diversity maximization problem discussed above, we need to pre-compute a set of clusters $(C_1, \ldots, C_k)$. Such a set of clusters can be found using many different techniques. While this is not the focus of this paper, for the sake of completeness, we discuss two simple algorithms to produce these clusters. We will also evaluate these algorithms in the experimental evaluation section. The ideas behind both of these algorithms are based on finding a subset $S = \{d_1, \ldots, d_k\}$ of top k center documents, and then building a cluster around each center d_i. In other words, we think of each document in this set as a center document for a cluster around it, i.e., after computing these center documents, we construct a clustering $\mathcal{C} = (C_1, C_2, \ldots, C_k)$ of documents where $d_i \in C_i$ by associating each document $d \in D$ to one or more *close* center documents in S. In computing the set of centers S, we aim to cover a large range of topics, and we take two approaches: maximize a weighted-coverage function of the documents in S, and maximize the diversity(S) while taking into account their weighted-coverage. We will first define the weighted-coverage function, and then algorithms aiming to optimize it.

4.1 Greedy Algorithm for Coverage Maximization

In this section, we define the weighted-coverage function and present an algorithm for the maximum weighted-coverage problem.

Weighted-coverage. Given a set of documents D and topics T, and an edge-weighted bipartite graph

$G(D, T, E)$ with edge weights $w(d, t)$, the weighted-coverage of subset $S \subseteq D$ is the sum of the weighted coverage of topics covered by documents in S, where the weighted coverage of each topic t is $\max_{d \in S} w(d, t)$ i.e., the extent to which this item t is covered. More formally, the weighted-coverage of documents in S is equal to:

$$\text{coverage}(S) = \sum_{t \in T} \max_{d \in S} w(d, t).$$

Maximum weighted-coverage problem. Given a parameter k (i.e., the number of documents) and an edge-weighted bipartite graph $G(D, T, E)$, the goal of the *maximum weighted-coverage problem* is to choose a subset S of k center documents maximizing the weighted-coverage coverage(S).

Note that this problem is different from the well-studied set cover or the maximum k-coverage problem [8], but it is still NP-hard, since the same problem with 0/1 as edge weights is equivalent with an unweighted version of maximum k-coverage problem [8] which is NP-Hard.

Greedy Algorithm for Coverage Maximization. Given the definition of the coverage function, the *greedy algorithm* to choose the top k documents is as follows: start from an empty set S of documents, at each step choose a document d with the maximum marginal increase in function coverage (i.e, $\max \text{coverage}(S \cup \{d\}) -$ coverage(S), and add this document to set S. Repeat this greedy selection procedure until k elements are picked.

By proving the submodularity of the weighted-coverage function defined above, we can prove that *greedy algorithm* above achieves at least $1 - \frac{1}{e}$ of the optimum of the maximum weighted-coverage problem [18]. A proof of this fact can be found in appendix.

4.2 Combined Heuristic for Center Selection

One way of choosing the set of centers in the clusters is to find k center documents with maximum diversity. However, other than the diversity metric, one may care about other objective functions such as relevance, weighted-coverage, and popularity of the micro-posts in the set of centers to form the clusters. A desirable property of the local search algorithm for maximizing diversity is its flexibility in choosing the order of local improvements, and the initial set of documents to start with. For example, if we want to simultaneously maximize the weighted coverage and diversity metrics, we can start with a set of documents with high weighted coverage, and then run the local search algorithm as a post-processing step. Also in running the local search algorithm, we should try to swap documents with higher weighted-coverage at each step. We use this guideline in the implementation of the local search algorithm. As an alternative approach to find the clusters, we also study the following heuristic for choosing the initial k centers by combining ideas for maximizing coverage and diversity. The algorithm is described in Figure 4.2.

Note that the algorithm in Figure 4.2 is a simple heuristic algorithm combing the ideas behind the greedy and the local search algorithms for maximizing weighted-coverage and diversity. While we do not prove a worst-case approximation factor for this combined heuristic, we will show its reasonable performance on real and random data in our experimental evaluations.

5. EXPERIMENTAL EVALUATION

In order to evaluate our algorithms in practice, in this section, we perform experiments on real world data (from Twitter) and on randomly generated data. More specifically, we examine the practical performance of our algorithms and their variants by comparing them with each other and a with baseline algorithm. We begin by describing the data and our metrics.

```
Combined Heuristic (for Coverage and Diversity of Centers)
Input: The edge-weighted bipartite graph G(D,T,E)
with edge weights $w(d,t)$ for each $d \in D$ and $t \in T$.
Output: A set $S$ of $k$ documents $S = \{d_1, d_2, \cdots, d_k\}$.
Goal: Find a set $S$ of cardinality $k$.
Initialize: $S = \emptyset$
Sort documents $D$ in the non-increasing order of
   weighted-coverage, i.e., in order of $\sum_{t \in T} w(d,t)$
Let the sorted documents be $(d_1, d_2, \ldots, d_{|D|})$
For $i$ from 1 to $|D|$ do
   if $|S| < k$ then
      Let $S := S \cup \{d_i\}$.
   elseif swapping $d_i$ with any $d_j$ for $j < i$ increases diversity, then
      Let $S := S \backslash \{d_j\} \cup \{d_i\}$.
```

Figure 1: Combined Heuristic for Cluster Center Selection

Real Data from Twitter. We run our experiments on 61 families of twitter posts, each of which is constructed as follows: We consider 10,000 to 25,000 top queries for Google real-time search, and for each query, we consider 30 top micro-posts returned for that query. This ranking is done based on various criteria such as the relevance of those twitter posts as well as their popularity and importance for the topic of the query. For each dataset, this process results in tens of thousands of documents, and a mapping of those top queries to thousands of topics. We construct the edge-weighted bipartite graph between these documents and topics using the algorithm described in Section 5. As a result, we get 61 edge-weighted bipartite graphs with an average of 1,277 topics and average of 12,500 documents. For this data set, the categories D_i forming the (partition) matroid constraints are sets of micro-posts from the same Twitter user. We consider identifying 10 or 20 centers and clusters with three representatives in each cluster.

Processing Twitter data. An important source of identifying hot topics in the online micro-blogging environment is the set of popular queries by users. As a first step of identifying important micro-posts, we identify popular queries that have been searched more often recently by users, and use these queries to retrieve a set of micro-posts matching those queries. In this setting, since tweets do not contain many terms, there is less need to discount term frequency by inverse document frequency (as in TF-IDF). We assume that a real-time search engine is available and one can identify the most relevant documents for those queries.

Let Q be the set of top queries on a real-time search engine. Each query has a query score query_score(q), representing the popularity of this query, e.g., the number of users who searched for this query. For each q in Q, let D_q be a ranked list of top l micro-posts or documents for q, and for each document $d \in D_q$, we also have a relevance_score(q, d).

For each q in Q, let $T_q \subset T$ be the set of topics corresponding to q. We transform each query q to a set of topics T_q, and construct an edge-weighted bipartite graph $G(D, T, E)$ between documents $D = \cup_{q \in Q} D_q$ and all topics $T = \cup_{q \in Q} T_q$. The weight of the edge between a topic $t \in T$ and and a document $d \in D$ is denoted by $w(d, t)$ and is computed as follows: for each query $q \in Q$ and each topic $t \in T_q$, we associate a weight weight(q, t) which is proportional to the score of the query normalized by the number of topics associated with this query q, e.g,

$$\text{weight}(q, t) = \frac{\text{query\_score}(q)}{|T_q|}.$$

Now the weight $w(d,t)$ in the topic-document bipartite graph is

$$w(d,t) = \sum_{q:t \in T_q} \text{weight}(q,t)\text{relevance\_score}(q,d).$$

This weight captures the extent to which online entry d covers topic t modeling the relevance of this topic to the micro-post. Intuitively, these topics represent the main topics of hot trends over the online environment. After forming the above edge-weighted topic-document bipartite graph, our goal is find a set of top documents covering the maximum amount of topics.

Randomly generated data. We generate the underlying bipartite graph between documents and topics at random. In particular, we generate families of random bipartite graphs each with 20000 documents and 2000 topics. We examine random bipartite graphs in which each edge is present with probability p independent of all other edges. We consider six values for the probability p, i.e., $p = 0.1\%, 0.5\%, 1\%, 2\%, 5\%, 10\%$ for having each edge in the bipartite graphs. The categories D_i for the partition matroid constraint are generated at random among 100 categories. We consider identifying 10 or 100 centers and clusters with three representatives in each cluster.

Metrics. We divide the practical evaluation of our algorithms into two main parts: 1) evaluation of the center selection algorithm for clustering, 2) evaluation of diversity maximization for representative selection given a set of clusters. For each part of the evaluation, we use an appropriate set of metrics. The main metrics are related to diversity and weighted-coverage as defined in this paper, and other metrics are inspired by the k-median and k-means problems with and without outliers [5]. Some of the metrics for evaluating both first and second parts are as follows:

- Weighted-coverage (W-COVERAGE): For a subset S of documents, the weighted-coverage of S is $\sum_{t \in T} \max_{d \in S} w(d,t)$.
- Diversity (DIVERSITY): For a subset S of documents, the diversity of S is $\sum_{d \in S} \sum_{d' \in S} \ell(d,d')$.
- Average distance to the centers (DIST-ALL): For a set $S = \{c_1, c_2, \ldots, c_k)$ of centers, the DIST-ALL objective function is the sum of distances of each document to the closest center, i.e.,

  $$\sum_{d \in D} \min_{1 \leq i \leq k} \ell(d, c_i).$$

  A smaller DIST-ALL implies better coverage of the set of centers.
- Percentage of documents covered in clustering (PERC): For a family of clusters $\mathcal{C} = (C_1, C_2, \ldots, C_k)$, this metric is simply the percentage of documents covered in this clustering, i.e., $\frac{|\cup_{1 \leq i \leq k} C_i|}{|D|.}$ A larger PERC indicates a larger coverage for the set of clusters.
- Average distance of covered documents to the centers (DIST-COVERED): For a set of clusters $\mathcal{C} = (C_1, C_2, \ldots, C_k)$ and a center $c_i \in C_i$ for each $1 \leq i \leq k$, we define the mean-distance DIST-COVERED to centers as

  $$\sum_{1 \leq i \leq k} \sum_{d \in C_i} \ell(d, c_i).$$

  A smaller DIST-COVERED implies better coherence for these clusters.

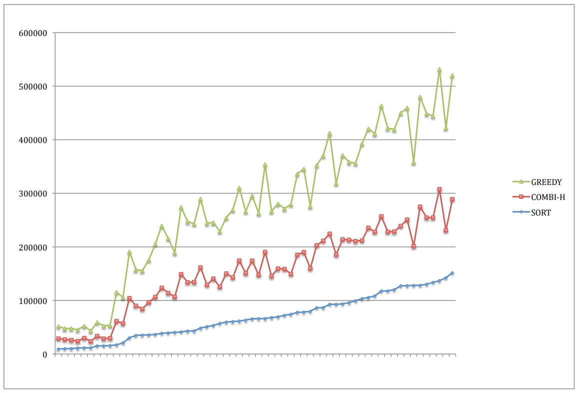

Figure 2: Vectors of the W-COVERAGE metric for all the instances. X-axis corresponds to instances, and Y axis corresponds to the coverage metric. From top to bottom, curves correspond to GREEDY, COMB-H, and SORT algorithms. Instances are ordered based on the coverage value of the output for SORT algorithm.

Table 1: Percent-coverage (PERC) of documents vs. the average distance of covered documents to the centers (DIST-COVERED) for three algorithms. The X axis represents PERC and Y axis corresponds to DIST-COVERED.

Algorithm	Diversity	N-Diversity	Coverage	N-Coverage
Comb-H	189.99	99%	86130	125%
GREEDY	189.90	99%	122981	178%
SORT	145.82	77%	68755	100%

W-COVERAGE and DIVERSITY can be defined for any set of documents, e.g. a set of centers for clusters, or union of sets of representatives from each cluster. We say a document d is covered by a subset of documents S, if at least for one document $d' \in S$, the distance $\ell(d,d')$ is less than 1. The PERC function is an unweighted variant of the W-COVERAGE, and the DIST-ALL and DIST-COVERED metrics correspond to the coherence of clusters, i.e, the closer the documents are to the centers (in terms of their ℓ distance), the smaller the DIST-ALL and DIST-COVERED functions are, and thus we have more coherent clusters. While DIST-ALL captures the average distance of all documents, the DIST-COVERED captures the average distance of documents covered by the centers.

One other metric for evaluating the second part (i.e, diversity maximization for choosing representative documents inside each given cluster) is as follows:

- Intra-cluster diversity (INTRA-DIVERSITY): For a family of clusters $\mathcal{C} = (C_1, C_2, \ldots, C_k)$ and a subset $R_i \subseteq C_i$ of representatives inside each cluster, we define the intra-cluster diversity as follows:

  $$\sum_{1 \leq i \leq k} \sum_{d,d' \in R_i} \ell(d,d').$$

Experiments. In the first part, we explore the following three algorithms and compare them in terms of diversity, weighted-coverage, and other metrics:

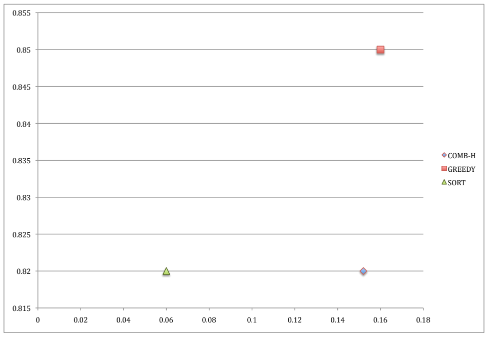

Figure 3: Average of DIVERSITY, normalized DIVERSITY, W-COVERAGE, and normalized W-COVERAGE of the initial set of centers various algorithms. DIVERSITY is normalize compared to the maximum possible diversity, i.e., $\binom{m}{2}$, where m is the number of documents. W-COVERAGE is normalized based on the coverage of the SORT algorithm.

Table 2: Average percentage of documents covered by each algorithm (PERC), average of average GJD distance of covered documents to the centers(DIST-COVERED), and average of average GJD distance of all documents to the centers(DIST-ALL).

Algorithm	PERC	DIST-COVERED	DIST-ALL
Comb-H	0.152	0.82	0.97
GREEDY	0.160	0.85	0.98
SORT	0.06	0.82	0.99

- Sort-based Algorithm [SORT]: Sort documents based on the total weight of topics connected to them in the edge-weighted bipartite graph.
- Greedy algorithm [GREEDY]: The greedy algorithm described in Section 4.1.
- The combined heuristic [COMB-H]: The combined heuristic algorithm, combining the local search and greedy algorithms, described in Section 4.2.

In the second part, we compare different algorithms for diversity maximization for representative selection in each cluster. We assume that given a set of center documents, clusters have been formed by putting each document in the cluster associated with the closest center. Documents that are not overlapping their set of topics with the any of the centers are thrown away, i.e., if the minimum generalized Jaccard distance (ℓ) of a document to centers is 1, then this document is not in any cluster. In particular, we compare the performance of the following three algorithms in terms of diversity, weighted-coverage, intra-cluster diversity, clustering coverage.

- A baseline sort heuristic [BASE]: In this algorithm, we choose the representatives as follows: In each cluster, the first representative is the center of the cluster from which this cluster was made. The rest of the $p-1$ representatives are $p-1$ documents with the maximum total weight of covering topics.
- The local search algorithm maximizing diversity of representatives [LOCAL] described in Section 3.

Table 3: Average of W-COVERAGE, DIVERSITY, and INTRA-DIVERSITY of the set of representatives for six algorithms on real data sets.

Algorithm	Diversity	Intra-Divers.	Coverage
Comb-H BASE	1732	24.62	131870
Comb-H INTRA-L	1755	47.17	124803
Comb-H LOCAL	1756	46.59	124195
Max-Cov. BASE	1730	22.58	136462
Max-Cov. INTRA-L	1755	47.03	124699
Max-Cov. LOCAL	1756	46.61	126107

- A modified local search algorithm maximizing the intra-cluster diversity [INTRA-L]: In this algorithm, we slightly modify the local search algorithm to optimize the intra-cluster diversity as opposed to total diversity of all representatives. In particular, we swap a document outside of the set with a document inside only if it increases the distance of this document to the set of documents in the same cluster.

Observations. Our experiments on both real and randomly generated data confirm the better performance of our algorithms for maximizing diversity and weighted-coverage. The better performance of the local search algorithm for diversity maximization compared to the greedy algorithm is more clear for randomly generated data sets.

For the real data, we report statistics for the set of 20 centers, or the 20 clusters based on these 20 centers, and the set of 60 representative documents in these clusters. First we observe that GREEDY and Comb-H algorithms consistently outperform the SORT algorithm for maximizing the weighted-coverage and diversity metrics respectively. For example, Figure 2 shows that the weighted-coverage of these two algorithms outperform that of SORT for each of the instances. Moreover, Table 1 shows that on average, coverage of GREEDY is 78% higher than that of SORT, and coverage of Comb-H is 25% higher than the coverage of SORT. Also the coverage of GREEDY is 42% higher than that of Comb-H. In addition, Figure 1 shows that on average, diversity of both GREEDY and Comb-H is 28% higher than diversity of SORT. Although we expect the diversity of Comb-H to be better than that of GREEDY, our real datasets do not show a significant difference between the diversity of GREEDY and Comb-H. However, we will observe such a difference on the random datasets (See Figure 4).

Other than the coverage and diversity metrics for the 20 centers, we also construct 20 clusters out of these centers by assigning each document to the closest center and report clustering metrics like PERC and DIST-COVERED, as well as the DIST-All metrics for these centers and clusters (See Figures 5 and 2). Intuitively, we expect algorithms with good coverage and diversity to have high percentage of covered documents, and small average distance of other documents to clusters, i.e., smaller DIST-ALL. We observe that GREEDY and Comb-H achieve comparable PERC and DIST-ALL. They both have larger PERC and smaller DIST-All compared to SORT. On the other hand, GREEDY has a slightly larger DIST-COVERED, implying that since GREEDY covers more topics, the resulting clusters may end up being less coherent, and thus there seem to be tradeoff between the coverage of the centers of the clusters and the coherency (DIST-COVERED) of the clusters. Figure 5 depicts this tradeoff. Finally, we report the average of Diversity, Intra-Diversity, and Coverage for the set of 60 representatives for the six algorithms combining Comb-H. and GREEDY (or Max-

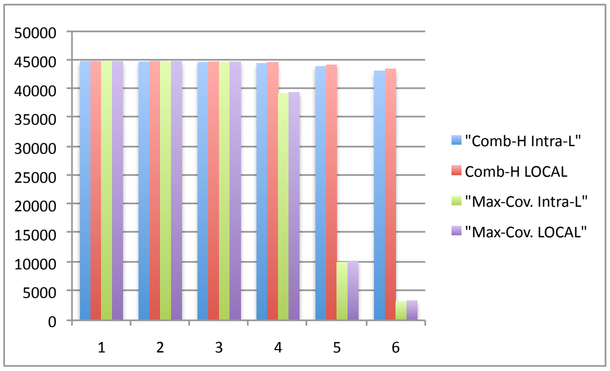

Figure 4: Average diversity metric for four algorithms on six random family of networks. On the X axis, from left to right, the bars correspond to random networks with probabilities $p = 0.1\%, 0.5\%, 1\%, 2\%, 5\%, 10\%$ on the edges of the bipartite graph.

Table 4: Average of W-COVERAGE, and DIVERSITY of two algorithms for six families of random graphs.

Algorithm	Diversity	Intra-Diversity	Coverage
Comb-H. LOCAL$p = 1\%$	44769	290	1470
Max-Cov. LOCAL $p = 1\%$	44747	298	1558
Comb-H. LOCAL$p = 2\%$	44641	291	1712
Max-Cov. LOCAL $p = 2\%$	39410	279	1749
Comb-H. LOCAL$p = 5\%$	44222	290	1879
Max-Cov. LOCAL $p = 5\%$	10150	141	1929
Comb-H. LOCAL$p = 10\%$	43498	285	1938
Max-Cov. LOCAL $p = 10\%$	3378	81	1999

Cov.) to find the centers and BASE, LOCAL, and INTRA-L to find the representative inside clusters (See Figure 3). The results from this part are similar to previous part, i.e., the coverage of Max-Cov. algorithm is larger, but the diversity of all the algorithms are comparable. The Intra-diversity of representatives for the algorithms using Intra-L is larger than the other algorithms.

The results for randomly data sets are similar in most aspects. Here, we discuss the main difference: while, for real data sets, the diversity metric for the output of GREEDY (or Max-Cov.) algorithm was comparable to that of Comb-H, we have observed families of instances of random data sets in which the average diversity of the output of Comb-H which is aiming to maximize the diversity is much higher. This is particularly significant for dense instances in which the probability of an edge in the document-topic bipartite graph is large, e.g. $p = 5\%$ or $p = 10\%$. For instance, for $p = 5\%$, the Diversity of the output of Comb-H. is more than 300% higher than that of GREEDY (See Figure 4). For brevity, we report the details of these metrics for a subset of algorithms (See Figure 5), but the results for other algorithms follow the same pattern.

6. USER STUDY

We conducted a user study with the goal to validate our metrics and algorithms. In this section, we describe the user study along with the results and observations.

The user study was designed in the form of a survey on Amazon's Mechanical Turk simulating a trends' aggregator website. Amazon's Mechanical Turk (MTurk) is a crowdsourcing online marketplace where *requesters* use human intelligence of *workers* to perform certain tasks, also known as HITs (Human Intelligence Tasks). Workers browse among existing tasks and complete them for a monetary payment set by the requester [16]. Once a worker completes the task, the requester can decide whether to approve it. In particular, if the requester believes that the worker completed the task randomly, she can reject her work. In that case, the worker does not get paid for the particular task and her approval rate is decreased. For our studies, we only hired workers that had approval rates of over 95%, that is, workers who had performed well in the past.

Our survey contains two categories of questions: The first question is how much people prefer seeing a diversified set of clusters, and the second question is to validate inter-cluster versus intra-cluster diversity, i.e, while showing a diversified set of clusters, do people want to see a diversified set of representatives inside each cluster or not.

For each category, we present four questions each with two options where each option is a set of clusters. The user is then asked which option she liked to see if she visited a realtime trends aggregator website. For the first category , the two options differed in the degree of diversity in them, e.g., one is generated by the diversity maximization algorithm and the other one is generated by the baseline sorting algorithm. In the second category, both options consist of clusters with a high diversity, but in one of them each cluster consists of a diversified set of representatives and in the other one, each cluster consists of a less diversified set of representatives. In each survey, we repeated two of the questions with the answers in reverse order for validity check. If the answers to those questions were inconsistent, we discarded the survey answers from that user. Overall, 130 out of 300 respondents had answered these two questions inconsistently. Therefore, we analyzed the remaining 170 valid responses.

Each question consists of two main parts: 1) *"Which one do you prefer?"* and 2) *"Which one do you think others prefer?"* The answer to the first question provides information on how a particular user likes diversity, and the answer to the second question shows how users expect an aggregator to look like. Moreover, we included the second question in the survey for two reasons. First, by asking the second question the subjects felt that the survey's goal was to study how people think their decisions are similar to the others, weakening the reactivity effect [11]. Second, we offered three bonus payments ($5 each, which is 50 times the amount we paid for each HIT) to the three workers whose answers to the second question was closest to the others' answers to the first question in order to deter workers from answering randomly.

Observations. We report the answers to questions 1 and 2 in each of the two categories as follows:

- For the first category, on average (over 4 questions) 65% of the respondents preferred to see a more diversified see of clusters. Moreover, 69% of the respondents thought that most people would like to see a diversified set of clusters.
- For the second category of questions, on average (over 4 questions) 72% of the respondents preferred to see less diversified representatives inside each cluster (intra diversity), while seeing a set of diversified clusters. Moreover, 76% of the respondents thought that most people would like to see lower intra diversity representatives.

Overall, we conclude that while most users would prefer to see a more diversified set of clusters in the aggregator website, many users prefer to see a less diversified set of items inside each cluster. This implies that the most appropriate metric and algorithm from the users' point of view is to maximize diversity among representatives of different clusters, but present more coherent representatives inside clusters.

7. CONCLUSION

In this paper, we study the diversity maximization problem under matroid constraints which is useful in a range of applications – in news aggregators, for aggregating trending topics in a microblogging environment, and in commerce search [2]. On the theoretical side, we present the first constant-factor approximation algorithm for this problem applying a new local search technique which implies the first constant-factor approximation algorithm for the maximum dispersion problem subject to matroid constraints. From the practical perspective, we show reasonable performance of these algorithms by running a user study validating our metrics, and by running experiments on synthetic and real data. From an algorithmic perspective, it would be interesting to design approximation algorithms that simultaneously maximize the diversity and weighted-coverage functions discussed in this paper. From a practical standpoint, it would be nice to apply these ideas in the context of commerce search [2] where matroid constraints are also relevant.

8. REFERENCES

[1] A. Anagnostopoulos, A. Z. Broder, and D. Carmel. Sampling Search-Engine Results. In *WWW*, 2006.

[2] S. Bhattacharya, S. Gollapudi, and K. Munagala. Consideration set generation in commerce search. In *WWW*, pages 317–326, 2011.

[3] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, 1998.

[4] M. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, pages 380–388, 2002.

[5] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *SODA*, pages 642–651, 2001.

[6] Z. Chen and T. Li. Addressing Diverse User Preferences in SQL-Query-Result Navigation. In *SIGMOD*, 2007.

[7] Z. Cheng, B. Gao, and T.-Y. Liu. Actively predicting diverse search intent from user browsing behaviors. In *WWW*, pages 221–230, 2010.

[8] U. Feige. A threshold of ln for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

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

[10] R. Hassin, S. Rubinstein, and A. Tamir. Approximation algorithms for maximum dispersion. *Oper. Res. Lett.*, 21(3):133–137, 1997.

[11] P. Heppner, B. Wampold, and D. Kivlighan. *Research design in counseling*. Brooks/Cole Pub Co, 2008.

[12] A. J. W. Hilton. On double diagonal and cross latin squares. *Journal of London Mathematical Sociecy*, 2-6:679–689, 1971.

[13] J. A. Konstan. Introduction to recommender systems. In *SIGIR*, 2007.

[14] G. Koutrika, A. Simitsis, and Y. Ioannidis. Précis: The Essence of a Query Answer. In *ICDE*, 2006.

[15] K. Kummamuru, R. Lotlikar, S. Roy, K. Singal, and R. Krishnapuram. A Hierarchical Monothetic Document Clustering Algorithm for Summarization and Browsing Search Results. In *WWW*, 2004.

[16] W. Mason and S. Suri. Conducting behavioral research on amazonâĂŹs mechanical turk. *Behavior Research Methods*, pages 1–23, 2010.

[17] S. McNee, J. Riedl, and J. A. Konstan. Being Accurate Is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In *CHI*, 2006.

[18] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions, 1978.

[19] F. Radlinski and S. T. Dumais. Improving Personalized Web Search using Result Diversification. In *SIGIR*, 2006.

[20] D. Rafiei, K. Bharat, and A. Shukla. Diversifying web search results. In *WWW*, pages 781–790, 2010.

[21] R. L. T. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *WWW*, pages 881–890, 2010.

[22] A. Schrijver. *Combinatorial Optimization*. Springer-Verlag, Berlin, 2003.

[23] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. Amer-Yahia. Efficient Online Computation of Diverse Query Results. In *ICDE*, 2008.

[24] D. Xin, H. Cheng, X. Yan, and J. Han. Extracting Redundancy-Aware Top-k Patterns. In *SIGKDD 2006*, 2006.

[25] C. Yu, L. Lakshmanan, and S. Amer-Yahia. Recommendation diversification using explanations. In *ICDE*, 2009.

[26] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW*, 2005.

APPENDIX

A. SUBMODULARITY OF THE WEIGHTED-COVERAGE FUNCTION

THEOREM A.1. *The weighted-coverage function* $\mathrm{coverage}$ *is a submodular set function, and thus the greedy algorithm is a* $1-\frac{1}{e}$*-approximation for the maximum weighted-coverage problem.*

PROOF. A set function g is submodular if and only if for any two subset $A \subseteq B$ and any element $i \notin B$, $g(B \cup \{i\}) - g(B) \leq g(A \cup \{i\}) - g(A)$. Consider two subsets $A \subseteq B$ of documents and a document $h \notin B$. For the weighted coverage function coverage, the difference in the value before and after adding h to sets A and B is equal to $\sum_{t \in T} \max(0, w(h,t) - (\max_{d \in A} w(d,t)))$, and $\sum_{t \in T} \max(0, w(h,t) - (\max_{d \in B} w(d,t)))$, respectively. Therefore, in order to show

$$\text{coverage}(B \cup \{h\} - \text{coverage}(B) \leq \text{coverage}(A \cup \{h\} - \text{coverage}(A),$$

we may prove the inequality for each corresponding term t, i.e, it is sufficient to prove that for each term $t \in T$,

$$\max(0, w(h,t) - (\max_{d \in A} w(d,t))) \geq \max(0, w(h,t) - (\max_{d \in B} w(d,t)))$$

To see the above inequality, first note that since $A \subseteq B$, we have $(\max_{d \in A} w(d,t)) \leq (\max_{d \in B} w(d,t))$. Also, by a simple case analysis we can see that if $x \leq y$, then $\max(0, M - x) \geq \max(0, M - y)$. Thus, the above inequality follows from that fact. □