ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/249963994

Ranking with Diverse Intents and Correlated Contents

Article · July 2013

DOI: 10.4230/LIPIcs.FSTTCS.2013.351 · Source: arXiv

CITATIONS
0

READS
16

2 authors:



Jian Li
Tsinghua University
87 PUBLICATIONS **776** CITATIONS
SEE PROFILE



Zeyu Zhang
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Instance Optimality in Multi-Armed Bandit View project

All content following this page was uploaded by Jian Li on 19 May 2015.

The user has requested enhancement of the downloaded file.

arXiv:1307.4518v2 [cs.DS] 18 Jul 2013

Ranking with Diverse Intents and Correlated Contents

Jian Li
lijian83@mail.tsinghua.edu.cn
IIIS
Tsinghua University

Zeyu Zhang
zyzhang92@gmail.com
Department of Mathematics
Tsinghua University

July 19, 2013

Abstract

We consider the following document ranking problem: We have a collection of documents, each containing some topics (e.g. sports, politics, economics). We also have a set of users with diverse interests. Assume that user u is interested in a subset I_u of topics. Each user u is also associated with a positive integer K_u, which indicates that u can be satisfied by any K_u topics in I_u. Each document s contains information for a subset C_s of topics. The objective is to pick one document at a time such that the average *satisfying time* is minimized, where a user's satisfying time is the first time that at least K_u topics in I_u are covered in the documents selected so far.

Our main result is an $O(\rho)$-approximation algorithm for the problem, where ρ is the algorithmic integrality gap of the linear programming relaxation of the set cover instance defined by the documents and topics. This result generalizes the constant approximations for generalized min-sum set cover and ranking with unrelated intents and the logarithmic approximation for the problem of ranking with submodular valuations (when the submodular function is the coverage function), and can be seen as an interpolation between these results. We further extend our model to the case when each user may interest in more than one sets of topics and when the user's valuation function is XOS, and obtain similar results for these models.

1 Introduction

1.1 Background

In a typical information retrieval application, we have a set of users and a set of documents. Each user issues a query and we would like to present the user with a rank list of the documents. Hopefully, the top-ranked documents are relevant to the user and our general objective is to maximize the overall user satisfaction. In many IR applications, the *probabilistic ranking principle* (PRP) is considered as a common rule of thumb to rank the documents [35]. PRP states that we should rank the documents in descending order by their probability of relevance and it is the "optimal" way to rank the documents in the sense that PRP minimizes the expected loss (also known as the Bayes risk) under 1/0 loss [30]. However, the 0/1 loss metric does not directly relate to the users' satisfaction and sometimes the ranking given by PRP is clearly suboptimal. Indeed, even the original paper [35] provided such an example (the example was discovered by W.S.Cooper).

Example 1. *[35] The class of users consists of two subclasses $\mathcal{U}_1$ and $\mathcal{U}_2$. $\mathcal{U}_1$ has 100 users and $\mathcal{U}_2$ has 50 users. Any user from $\mathcal{U}_1$ would be satisfied with any document s_1–s_9, but no others. Any*

user from $\mathcal{U}_2$ would be satisfied with only s_{10}. If we consider any document s_1–s_9 on its own, it has a probability of $2/3$ of being relevant to the next user (the ranking algorithm does not know which subclass the user belongs to). Similarly, s_{10} has a probability of $1/3$ of being relevant. Therefore, by PRP, the ranking should be $s_1, s_2, \ldots, s_9, s_{10}$. But this means that $\mathcal{U}_1$ users can be satisfied with s_1 while $\mathcal{U}_2$ users have to see nine irrelevant documents before they retrieve s_{10}. Consider the ranking $s_1, s_{10}, s_2, \ldots, s_9$. $\mathcal{U}_1$ users are still satisfied by the first document, but $\mathcal{U}_2$ users are satisfied with the second document, which is much better than the ranking defined by PRP.

The action of placing several documents aiming at different types of users at the top positions of the rank list (e.g. place s_1 and s_2 as the top-2 in the above example) is called *diversification*. It is a widely accepted fact that *diversification* of the ranking result is helpful in minimizing the risk of user dissatisfaction in a multiuser scenario (See, e.g., [13, 17, 34, 1, 39, 19, 23, 5]). Example 1 is a simple yet instructive illustration why the diverse intents and the correlations of the documents (s_1–s_9 are correlated in a way that any of them could satisfy a $\mathcal{U}_1$ user) are the major reasons for diversification.

1. Diverse intentions: Different users may have different intents towards the same query (e.g., a keyword). However, the ranking algorithm does not know the actual type of an individual user but has to use the same ranking function for the same query. In Example 1, there are two user types $\mathcal{U}_1$ and $\mathcal{U}_2$, and the next user could be either of them. Considering another real life example, the keyword "Michael Jordan" may refer to the famous NBA player in one query, and the U.C. Berkeley Professor in another search.

2. Correlations among documents. Typically, the utility a user can obtain from a set of documents is not the sum of the utilities from individual documents in the set. This is because of the similarity (or dissimilarity) of the documents. For instance, the utility of two very similar documents is not much more than the utility of one of them (e.g., documents s_1 and s_2 in Example 1). Such correlations can be seen as another cause of diversification of the ranking result (see e.g., [1, 39, 19]).

1.2 Problem Formulation

In this section, we propose our model for diversification, which captures both the diversity of users' intents and the correlations of the documents.

Definition 2. *Ranking with Diverse Intents and Correlated Contents (RDC): Here, we have a set $\mathcal{U}$ of users, a set $\mathcal{S}$ of documents, and a set $\mathcal{E}$ of topics. Each user u is interested in a subset I_u of topics. Each user u is also associated with a positive integer K_u which is less or equal to $|I_u|$. Each document s contains a subset C_s of topics and $\mathcal{E} = \bigcup_{s \in \mathcal{S}} C_s$. The objective is to pick an ordering of all documents such that the average satisfying time is minimized, where a user's satisfying time $t_u = \min\{t \mid$ at least K_u topics in I_u are covered by the first t selected documents$\}$.*

It is not hard to see that our RDC model captures both the diversity of the users' intents (i.e., each user is interested in a different subset of topics) and the correlations among documents (i.e., different documents may have some common topics). Now, we discuss some closely related prior work and their relations with our model.

1. Ranking with multiple intents (R-Multi) [5]: Azar et al. proposed the following combinatorial model to capture the diversity of user preferences. We have a set $\mathcal{U}$ of users and a set $\mathcal{S}$ of documents. User u can be satisfied with any K_u document from a subset I_u of documents. The objective is the same as ours, to minimize the cumulative users' satisfying time. We can see that it is a special case of RDC where each document contains a distinct topic.

2. Ranking with unrelated intents (R-Unrel) [3]: This model is a generalization of R-Multi. For each user u and a document s, there is a nonnegative number A_{us} that is the amount of utility u can get from s. u is satisfied if she accumulates at least K_u units of utility. The objective is same as R-Multi. It is also not hard to see that R-Unrel is a special case of RDC where each document contains A_{us} distinct topics.

3. Ranking with submodular intents (R-Submod) [4]: The model is a generalization of both R-Multi and R-Unrel. For each user u, there is a nonnegative submodular function $f_u : \{0,1\}^{\mathcal{S}} \to \mathbb{R}^+ \cup \{0\}$. u is satisfied if the set S of documents she gets is such that $f_u(S) \geq 1$. The objective is same as before. R-Submod generalizes RDC (as well as R-Multi and R-Unrel). If the submodular function f_u is the *coverage function* [1], R-Submod is equivalent to RDC.

1.3 Our results

We find that the approximability of RDC is closely related with the (algorithmic) integrality gap of the underlining set cover instance induced by the documents and topics. In particular, we can show the following result. Let $F \subseteq \mathcal{E}$ be a subset of topics. We denote $\mathsf{SC}(F)$ the set cover instance formed by the subsets $C_s : s \in \mathcal{S}$ and the set of topics in F. Let $\mathsf{LP}(\mathsf{SC}(F))$ be the natural linear programming relaxation for $\mathsf{SC}(F)$:

$$\begin{aligned} \text{minimize} \quad & \sum_{s \in \mathcal{S}} x_s & \\ \text{subject to} \quad & \sum_{s: e \in C_s, s \in \mathcal{S}} x_s \geq 1 & \forall e \in F \\ & x_s \geq 0 & \forall s \in \mathcal{S} \end{aligned}$$

Theorem 3. *Suppose for any $F \subseteq \mathcal{E}$, there is a polynomial time algorithm that can produce a solution for $\mathsf{SC}(F)$ whose cost is at most ρ times the optimal value of $\mathsf{LP}(\mathsf{SC}(F))$. There is a polynomial time factor $O(\rho)$ approximation algorithm for RDC.*

First, we can see that Theorem 3 produces $O(1)$ factor approximation for both R-Multi and R-Unrel. As we mentioned before, if we view R-Multi and R-Unrel as special cases of RDC, the induced set cover instances have very simple structure: each set (document) consists of a disjoint set of elements (topics). In both R-Multi and RDC, the integrality gap of $\mathsf{LP}(\mathsf{SC}(F))$ is 1 for any $F \subseteq \mathcal{E}$ and we can find an integral optimal solution in polynomial time (the algorithm trivially includes all subsets that contains at least one element in F). Hence, $\rho = 1$ and we have a constant factor approximation algorithm. Therefore, our result generalizes the constant approximations for R-Multi in [6, 36, 27] and that for R-Unrel in [3].

[1] The set of documents and the set of topics in I_u form a set cover instance, where I_u is the subset of topics which user u is interested in. For $S \subseteq \mathcal{S}$, the coverage function $f(S)$ is the number of topics in I_u covered by some document in S.

For R-Submod, Azar et al. showed that there is an $O(\log \frac{1}{\epsilon})$ approximation for the problem where ϵ is the minimum non-zero marginal value for f_is [4]. If the submodular functions are coverage function, the result translates to an $O(\log |\mathcal{E}|)$-approximation. It is well known that we can round any fractional solution of $\mathsf{LP}(\mathsf{SC}(F))$ to an integral solution such that the cost of the integer solution is at most $\log |\mathcal{E}|$ of the value of the fractional solution. Hence, Theorem 3 also gives an $O(\log |\mathcal{E}|)$-approximation, reproducing the result in [4] for R-Submod with coverage functions (with a somewhat larger constant hidden in the big-O notation).

Our result can be seen as an interpolation between the constant approximation for R-Multi and R-Unrel (which induce trivial set cover instances) and the logarithmic approximation for R-Submod(which may induce arbitrary set cover instances). Besides the above implications on previous problems, Theorem 3 is also interesting since typically the set cover instances induced by the documents and topics are much easier to approximate than general set cover problem. We provide some useful examples.

1. In R-Multi and R-Unrel, the induced set cover instances are trivial and can be solved optimally.

2. Consider another interesting example where each topic is covered by at most d documents. It is known that we can obtain a d-approximation by a simple deterministic rounding or primal-dual techniques (see e.g. [38]). Hence, in this case, we have an $O(d)$-approximation for RDC.

3. Suppose the VC dimension of the set system $(\mathcal{E}, \mathcal{S})$ is d. It is well known that we can achieve an approximation factor of $O(d \log \tau)$ via the LP approach [20], where τ is the optimum LP value ($O(d \log \mathsf{OPT})$ is known even earlier via non-LP approach [12]). In many cases, $O(d \log \tau)$ can be much smaller than $O(\log |\mathcal{E}|)$.

4. For some geometric set cover problems, we can achieve sub-logarithmic factor approximation algorithms using LP approaches. For example, if each subset corresponds to a unit disk in the plane and each element corresponds to a point, there is a constant approximation [33]. For general disk graphs, a $2^{O(\log^* |\mathcal{E}|)}$-approximation is known [37].

   Several sub-logarithmic factor approximation algorithms are known for certain geometric set cover problems via other techniques, such as local search or dynamic programming [2, 32, 22, 14]. However, it is not clear how to combine those techniques with our LP approach. We leave this as an interesting open question.

Even though the real world topics and documents may not necessarily have low VC-dimension or match any geometric set cover instance, it is still our general belief that the real world instance do not form arbitrary set system and the particularity of those instances may help us to develop sub-logarithm factor approximations, which further implies that RDC can be approximated within the same factor (up to a constant). Exploring the particularity of the real world instances is left as an open question of great importance.

1.4 Related work

Azar et al. [5] introduced R-Multi and first gave an $O(\log n)$ factor approximation algorithm. Bansal et al. [6] improved the approximation ratio to a constant (a few hundreds). Subsequently, the constant was further reduced to about 28 in [36], and then to 12.4 [27]. An important special

case of R-Multi, where $K_u = 1$ for each u, is called the *min-sum set cover* problem. Feige et al. [21] developed a 4-approximation and proved that it is NP-hard to achieve an approximation factor of $4 - \epsilon$ for any constant $\epsilon > 0$. In fact, it is conjectured that R-Multi can also be approximated within a factor of 4 [27]. Another special case of R-Multi where $K_u = |I_u|$ has also been studied under the name of *minimum latency set cover* and it is known that there is a polynomial time approximation algorithm with factor 2 [25, 28], which is also optimal assuming a variant of the Unique Games Conjecture [9]. Im et al. [26] considered a generalization of R-Submod where there is metric switching cost and gave a poly-logarithmic factor approximation algorithm for it.

There is a huge literature on search result diversification in IR and DB literature. We refer interested readers to [13, 17, 1, 19, 23, 10] and the references therein. In practice, the overall satisfying time as defined above is not a direct measure of the overall user satisfaction. Alternative measures have been proposed in the literature, such as discounted cumulative gain (DCG) and mean average precision (MAP). Bansal et al. considered R-Multi with DCG being the objective function and obtained an $O(\log \log n)$-approximation [8].

2 A Constant Factor Approximation Algorithm

In this section, we will prove Theorem 3 by giving a randomized LP rounding algorithm.

2.1 The LP Relaxation

We use the following linear program relaxation. Here we use boolean variable x_{st} to represent whether document s is selected at time t. y_{ut} indicates if user u is satisfied after time t. z_{st} represents if document s has been selected at time t.

(**LP**) :

$$\text{minimize} \quad \sum_{u \in \mathcal{U}} \sum_{t=1}^{n} (1 - y_{ut}) \tag{1}$$

$$\text{subject to} \quad \sum_{t=1}^{n} x_{st} = 1 \qquad \forall s \in \mathcal{S} \tag{2}$$

$$\sum_{s \in \mathcal{S}} x_{st} = 1 \qquad \forall t \in [n] \tag{3}$$

$$z_{st} = \sum_{t'=1}^{t} x_{st'} \qquad \forall t \in [n] \tag{4}$$

$$\sum_{e \in I_u} (y_{ut} - \min\{ \sum_{s: e \in C_s} z_{st}, 1\}) \leq (|I_u| - K_u) y_{ut} \qquad \forall u \in \mathcal{U}, t \in [n] \tag{5}$$

$$x_{st}, y_{ut}, z_{st} \in [0, 1] \qquad \forall s \in \mathcal{S}, u \in \mathcal{U}, t \in [n] \tag{6}$$

Constraints (2) and (3) make sure that a document can be selected only once and each time we pick one document. The meaning of z_{st} is captured in constraints (4). Constraints (5) guarantee that a user u is satisfied if less than $|I_u| - K_u$ topics havn't been covered. However, it is known that the integrality gap of this LP is unbounded (even for R-Multi) [6]. To remedy this, [6] uses the *knapsack cover* constraints to replace the simple covering constraints (5) In our case, we define

$S(e,u,F) = \{s \mid e \in C_s, s \in T_2(u,F)\}$ where $T_1(u,F)$ is the set of all documents that cover at least $K_u - |F|$ topics in $I_u \backslash F$, and $T_2(u,F) = \mathcal{S} \backslash T_1(u,F)$. And we use the following constraints instead of (5):

$$y_{ut}(K_u - |F|) \leq (K_u - |F|) \sum_{s \in T_1(u,F)} z_{st} + \sum_{e \in I_u \backslash F} \sum_{s \in S(e,u,F)} z_{st}$$
$$\forall u \in \mathcal{U}, t \in [n], F \subseteq I_u, |F| \leq K_u \qquad (7)$$

Constraints (7) differ from the knapsack cover constraints in [6] in that we handle sets T_1 and T_2 seperately, for technical reason that will be clear from the analysis.

Now we show that (**LP**) is indeed an LP relaxation of RDC. We just need to prove that any feasible solution to RDC satisfies constraints (7): If $y_{ut} = 0$, the inequality must be true because the left side is 0. If $y_{ut} = 1$, there are two cases. The first case is that at least one document in $T_1(u,F)$ has been selected, which means $\sum_{s \in T_1(u,F)} z_{st} \geq 1$. The other case is that at least $(K_u - |F|)$ topics have related document in $T_2(u,F)$, which means $\sum_{e \in I_u \backslash F} \sum_{s \in S(e,u,F)} z_{st} \geq (K_u - |F|)$. Therefore both cases satisfy the inequality. So we have proved the following lemma:

Lemma 4. *The optimal value* $\mathsf{OPT}_{\mathsf{LP}}$ *of (**LP**) is at most the optimal total satisfying time of RDC.*

2.2 A Randomized Rounding Algorithm

Assume (x^*, y^*, z^*) be the optimal fractional solution to (**LP**). We also assume that for any $F \subseteq \mathcal{E}$, there is a poly-time algorithm **AlgoSC** which can produce an integral solution for $\mathsf{SC}(F)$ whose cost is at most ρ times the value of the *fractional* optimal solution to $\mathsf{LP}(\mathsf{SC}(F))$. Our randomized rounding scheme consists of $\lceil \log n \rceil + 1$ rounds, where in the k-th round, we perform the following procedure.

- Let $t = 2^k$, $G_k = \emptyset$ and $p_e = \min\{1, 50 \sum_{s:e \in C_s} z^*_{st}\}$, $\forall e \in \mathcal{E}$.
- Let $P_k = \{e \in \mathcal{E}, p_e = 1\}$. Let the set $H_k \subseteq \mathcal{S}$ be the solution of **AlgoSC**$(\mathsf{SC}(P_k))$.
- For each $s \in \mathcal{S} \backslash H_k$, add document s to G_k independently with probability $\min\{1, 50 z^*_{st}\}$.
- If there are more than $(70 + \rho) \cdot 2^k$ documents in $H_k \cup G_k$, we say this round is "overflowed" and select nothing, else we select all the documents in $H_k \cup G_k$ in arbitrary order in this round.

Our algorithm builds on the ideas developed in [6] (as well as [3]). A key technical difference between our algorithm and [6] is that we need to deal with those topics that are almost covered (i.e., the set P_k) and the rest separately. It will be clear soon from the analysis, for a particular user u, independent rounding (step 3) can guarantee that, at a cost not much more than the fractional optimal, topics in $I_u \backslash P_k$ are covered with constant probability. For these topics, we can use a Chernoff-like concentration result for submodular functions to show this. Topics in $I_u \cap P_k$ are handled separately by **AlgoSC** to make sure they are covered in the k-th round. This is where the approximiblity of the set cover instance jumps in.

2.3 The Analysis

Constraints (4) and (6) show that the optimal solution z^*_{st} is monotonically non-decreasing with t for all $s \in \mathcal{S}$. Thus it's easy to see that y^*_{ut} is monotonically non-decreasing with t for all $u \in \mathcal{U}$.

For each $u \in \mathcal{U}$, let $t^*_u = \max\{t \in [n] \mid y^*_{ut} \leq \frac{1}{2}\}$, then $\sum_{t=1}^{n}(1 - y^*_{ut}) \leq \sum_{t=1}^{t^*_u}(1 - y^*_{ut}) \leq \frac{1}{2}t^*_u$. Thus we have the fact that $\mathsf{OPT}_{\mathsf{LP}} \geq \frac{1}{2}\sum_u t^*_u$.

Before we start to prove our Theorem 3, we need the following Chernoff-type bounds:

Lemma 5. *If $X_1, X_2, \ldots, X_n$ are independent $\{0,1\}$-valued random variables with $X = \sum_i X_i$ such that $\mathbb{E}[X] = \mu$, then we have that*

1. [31] $\Pr[X < (1-\delta)\mu] \leq e^{-\frac{\delta^2}{2}\mu}$.
2. [11] $\Pr[X > \mu + \beta] \leq \exp(-\frac{\beta^2}{2\mu + \frac{2}{3}\beta})$.

Lemma 6. [15] *Let $f : \{0,1\}^n \to \mathbb{R}^+$ be a monotone submodular function with marginal values in $[0,1]$. Let $\mu = \mathbb{E}[f(X_1, \ldots, X_n)]$. Then for any $\delta > 0$,*

$$\Pr\Big[f(X_1, \ldots, X_n) \leq (1-\delta)\mu\Big] \leq e^{-\frac{\delta^2}{2}\mu}.$$

We now give the following lemma:

Lemma 7. *For any user $u \in \mathcal{U}$ and a non-overflowed round k such that $2^k \geq t^*_u$. The probability that $H_k \cup G_k$ does not satisfy u is at most 0.023.*

Proof. Fix a user u. Consider constraints (7) for $F = P_k \cap I_u$ and $t = 2^k$. If $|F| \geq K_u$, u is clearly satisfied in this phase because all the documents in H_k are selected in this round. Therefore, we consider the case where $|F| < K_u$. From constraints (7) we know that:

$$(K_u - |F|)\sum_{s \in T_1(u,F)} z^*_{st} + \sum_{e \in I_u \setminus F}\sum_{s \in S(e,u,F)} z^*_{st} \geq y^*_{ut}(K_u - |F|) \geq \frac{1}{2}(K_u - |F|).$$

Here, either $\sum_{s \in T_1(u,F)} z^*_{st}$ must be greater or equal to $\frac{1}{5}$, or $\sum_{e \in I_u \setminus F}\sum_{s \in S(e,u,F)} z^*_{st}$ must be greater or equal to $\frac{3}{10}(K_u - |F|)$.

In the case of $\sum_{s \in T_1(u,F)} z^*_{st} \geq \frac{1}{5}$. If there is an $s \in T_1(u,F)$ such that $50 z_{st} \geq 1$, then s is selected and user u is satisfied. Otherwise, since we select documents independently in our algorithm and the expected number of selected documents in $T_1(u,F)$ is

$$\mathbb{E}\Big[|(G_k \cup H_k) \cap T_1(u,F)|\Big] = \sum_{s \in T_1(u,F) \setminus H_k} 50 z^*_{st} + |T_1(u,F) \cap H_k| \geq \sum_{s \in T_1(u,F)} 50 z^*_{st} \geq 50 \times \frac{1}{5} = 10.$$

From Lemma 5 (1), we know that the probability that $H_k \cup G_k$ contains less than one document in $T_1(u,F)$ is

$$\Pr\Big[|(G_k \cup H_k) \cap T_1(u,F)| < (1 - \frac{9}{10}) \cdot 10\Big] \leq \exp(-\frac{(\frac{9}{10})^2}{2} \cdot 10) < 0.018.$$

Therefore the probability that user u is not satisfied by $H_k \cup G_k$ in this case is at most 0.018.

In the case of $\sum_{e\in I_u\setminus F}\sum_{s\in S(e,u,F)} z^*_{st} \geq \frac{3}{10}(K_u - |F|)$, assume boolean vector $\mathbf{w} = \{w_s\} \in \{0,1\}^{|T_2(u,F)|}$ indicates the selected documents in $T_2(u,F)$. Let submodular function $f(\mathbf{w}) = \sum_{e\in I_u\setminus F} \min\{1, \sum_{s\in S(e,u,F)} w_s\}$, i.e. the number of topics in $I_u\setminus F$ that the selection of documents $\mathbf{w}$ covers. Suppose $\overline{\mathbf{z}_t} = \{\overline{z^*_{st}}\}_{s\in T_2(u,F)}$ to be a random 0/1 vector that is obtained as follows: Independently set $\overline{z^*_{st}}$ to be 0 with probability $(1 - 50z^*_{st})$ if $s \in T_2(u,F)\setminus H_k$, and 1 otherwise. Since for any $e \in I_u\setminus F$, $\sum_{s:e\in C_s} z^*_{st} < \frac{1}{50}$ (see the definition of F and P_k), we can find:

$$\begin{aligned}
\mathbb{E}\Big[f(\overline{\mathbf{z}_t})\Big] &= \sum_{e\in I_u\setminus F} \Pr\Big[e \text{ is covered by } \overline{\mathbf{z}_t}\Big] \\
&= \sum_{e\in (I_u\setminus F)\setminus \bigcup_{s\in H_k} C_s} (1 - \prod_{s\in S(e,u,F)} (1 - 50z^*_{st})) + |\{e \mid e \in (I_u\setminus F) \cap \bigcup_{s\in H_k} C_s\}| \\
&\geq \sum_{e\in I_u\setminus F} (1 - \prod_{s\in S(e,u,F)} (1 - 50z^*_{st})) \\
&\geq \sum_{e\in I_u\setminus F} (1 - \prod_{s\in S(e,u,F)} e^{-50z^*_{st}}) \\
&\geq \sum_{e\in I_u\setminus F} (1 - \exp(-\sum_{s\in S(e,u,F)} 50z^*_{st})) \\
&\geq \sum_{e\in I_u\setminus F} (1 - \tfrac{1}{e}) \sum_{s\in S(e,u,F)} 50z^*_{st} \\
&\geq (1 - \tfrac{1}{e})15(K_u - |F|)
\end{aligned}$$

where the penultimate inequality is because $(1 - e^{-x}) \geq (1 - \frac{1}{e})x$, $\forall x \in [0,1]$.

From Lemma 6, we know that

$$\Pr[f(\overline{\mathbf{z}_t}) \leq |K_u| - |F|] = \Pr[f(\overline{\mathbf{z}_t}) \leq (1 - \frac{14e-15}{15e-15})] \leq e^{-(\frac{14e-15}{15e-15})^2 \frac{15e-15}{2e}} < 0.023.$$

This shows that the probability that user u is not satisfied by $H_k \cup G_k$ in this case is at most 0.023, which complete the proof of Lemma 7. □

Lemma 8. *The probability that the algorithm "overflowed" in round k is at most 0.03.*

Proof. First we show $|H_k| \leq \rho \times 2^k$. It is easy to see that $\mathbf{z} = \{z_{st}\}_{s\in\mathcal{S}}$ is a feasible solution of $\mathsf{LP}(\mathsf{SC}(P_k)))$. By our assumption on **AlgoSC**, we have that

$$|H_k| \leq \rho \sum_{s\in\mathcal{S}} z^*_{st} = \rho \sum_{s\in\mathcal{S}} \sum_{t'=1}^{t} x^*_{st'} = \rho \sum_{t'=1}^{t} \sum_{s\in\mathcal{S}} x^*_{st'} = \rho \cdot 2^k.$$

where the last equation is because $t = 2^k$ in round k.

Therefore, it is suffice to show that $\Pr[|G_k| \geq 70 \cdot 2^k] < 0.03$. In our setting,

$$\mathbb{E}\Big[|G_k|\Big] = \sum_{s\in\mathcal{S}\setminus H_k} \min\{1, 50z^*_{st}\} \leq \sum_{s\in\mathcal{S}} 50z^*_{st} = \sum_{s\in\mathcal{S}} \sum_{t'=1}^{t} 50x^*_{st'} = \sum_{t'=1}^{t} \sum_{s\in\mathcal{S}} 50x^*_{st'} = 50 \cdot 2^k.$$

From Lemma 5 (2), we know that

$$\Pr\Big[|G_k| > 50 \cdot 2^k + 20 \cdot 2^k\Big] \leq \exp(-\frac{400 \cdot 2^{2k}}{100 \cdot 2^k + \frac{40}{3} \cdot 2^k}) < 0.03.$$

□

Now we will prove our Theorem 3.

Proof. Let **Satisfy**(u) denote the satisfying time of u in our algorithm. From constraints (2), (3) and (4), we know that $z^*_{sn} = 1$ for all $s \in \mathcal{S}$, thus all the users must be satisfied after $\lceil \log n \rceil + 1$ rounds. If some user u is satisfied before the $\lceil \log t^*_u \rceil$th round, the satisfying time **Satisfy**(u) $\leq 2^{\lceil \log t^*_u \rceil}$.

Otherwise, if some user u is satisfied after the $\lceil \log t^*_u \rceil$th round, since we select at most $(70+\rho)\cdot 2^k$ documents in each round, the satisfying time of user u is at most $2 \cdot (70+\rho) \cdot 2^k$ if he is satisfied in the k-th round. From Lemma 7 and Lemma 8, we know that the probability that user u isn't satisfied after the k-th round where $2^k \geq t^*_u$ is less than $1-(1-0.023)\times(1-0.03) < 0.053$. Notice that the probability is independent in each round, we get the expected total satisfying time:

$$\begin{aligned}
\mathbb{E}\Big[\sum_{u\in\mathcal{U}} \textbf{Satisfy}(\text{u})\Big] &\leq \sum_{u\in\mathcal{U}} (2\cdot(70+\rho)\cdot 2^{\lceil \log t^*_u\rceil} + \sum_{i=\lceil \log t^*_u\rceil+1}^{\lceil \log n\rceil+1} (70+\rho)\cdot 2^i \cdot 0.053^{i-\lceil \log t^*_u\rceil}) \\
&\leq \sum_{u\in\mathcal{U}} ((140+2\rho)t^*_u + (70+\rho)t^*_u \sum_{i=1}^{\infty} 0.106^i) \\
&< (149+2.12\rho)\sum_{u\in\mathcal{U}} t^*_u \\
&< (298+4.3\rho)\mathsf{OPT}_{\mathsf{LP}} = O(\rho\mathsf{OPT}).
\end{aligned}$$

This complete the proof of Theorem 3 from Lemma 4. □

2.4 Solving the LP

In order to use ellipsoid method to find the optimal solution, we need to find a polynomial-time separation oracle that verifies if a candidate solution satisfies all constraints [24]. Unfortunately, constraints (7) contains exponentially many inequalities and it is hard to find such a separation oracle. However, we can use the trick mentioned in [3]. Note that in our analysis, we only consider one knapsack inequality in each round where $F = P_k \cap I_u$ and $t = 2^k$. Thus if there is a solution satisfies all these $\lceil \log n \rceil + 1$ inequalities in (7) as well as other constraints in (2), (3), (4) and (6), it is enough for our algorithm even if it is not a feasible solution for (**LP**). Therefore in each iteration of the ellipsoid method, we just need to check the validity of polynomial constraints, which forms a polynomial algorithm.

3 Extensions

3.1 Ranking with Groups of Intents and Correlated Contents (RGC)

Now we extend RDC to the problem that each user u may interest in more than one sets of topics $I_{u1}, I_{u2}, \ldots, I_{up}$, and a user is satisfied if at least one of these groups is satisfied, where p is at most polynomial of n. Same as RDC, a set I_{ui} is satisfied if K_{ui} topics in I_{ui} are covered. This time we

could change our relaxed LP as follows:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{u\in\mathcal{U}}\sum_{t=1}^{n}(1-y_{ut}) & \\
\text{subject to} \quad & \sum_{t=1}^{n} x_{st} = 1 & \forall s \in \mathcal{S} \\
& \sum_{s\in\mathcal{S}} x_{st} = 1 & \forall t \in [n] \\
& y_{ut} \le \max_i\{g_{uit}\} & \forall u \in \mathcal{U}, t \in [n] \\
& z_{st} = \sum_{t'=1}^{t} x_{st'} & \forall t \in [n] \\
& g_{uit}(K_{ui} - |F|) \le (K_{ui} - |F|) \quad \sum_{s\in T_1(u,i,F)} z_{st} + \sum_{e\in I_u\setminus F}\sum_{s:e\in C_s, s\in T_2(u,i,F)} z_{st} & \\
& & \forall i \in [p], u \in \mathcal{U}, t \in [n], F \subseteq I_{ui}, |F| \le K_{ui} \\
& x_{st}, y_{ut}, z_{st}, g_{uit} \in [0,1] & \forall s \in \mathcal{S}, u \in \mathcal{U}, t \in [n], i \in [p]
\end{aligned}
$$

where $T_1(u,i,F)$ is the set of all documents that cover at least $K_{ui}-|F|$ objects in $I_{ui}\setminus F$, $T_2(u,i,F) = \mathcal{S}\setminus T_1(u,i,F)$, and g_{uit} indicates if for user u, group i is satisfied after time t.

The algorithm and analysis are almost the same as in RDC, so we won't talk about it more. Finally we get Theorem 9.

Theorem 9. *Suppose for any $F \subseteq \mathcal{E}$, there is a polynomial time algorithm that can produce a solution for $\mathsf{SC}(F)$ whose cost is at most ρ times the optimal value of $\mathsf{LP}(\mathsf{SC}(F))$. There is a polynomial time factor $O(\rho)$ approximation algorithm for RGC.*

3.2 Ranking with XOS Valuations (RXOS)

Notice that all the problems we have mentioned in this paper are special cases of R-Submod, where the users' satisfying functions are submodular functions of the set of documents. There is another family of valuations called XOS. An XOS function is a set function which is the maximum of several additive set functions. An additive set function $f : \{0,1\}^{\mathcal{S}} \to \mathbb{R}^+ \cup \{0\}$ has the form $f(F) = \sum_{s\in F} A_s$, $\forall F \subseteq S$, where A_s is a constant associated with each element $s \in S$. Since the family of submodular functions is contained in XOS [29], R-Submod is a special case of RXOS.[2] Suppose for each user, the XOS function contains only polynomial number of additive set functions which are non-negative. We can give an O(1)-approximation algorithm. (If the number of additive set functions is exponential, the best approximate rate we can hope for is $O(\log n)$ since RXOS generates R-Submod.)

For each user u, suppose the additive set functions be $f_{ui}, i = 1,2,\ldots,p$, where $f_{ui}(F) = \sum_{s\in F} A_{uis}$ for $F \subseteq \mathcal{S}$. Without lose of generality, we can let the satisfying time of user u be $t_u = \min\{t \mid \max_i f_{ui}(\text{the first } t \text{ selected documents}) \ge 1\}$. Now we could have the following LP

[2]The number of additive set functions which are needed to represent a submodular function may be exponential

relaxation:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{u\in\mathcal{U}}\sum_{t=1}^{n}(1-y_{ut}) & \\
\text{subject to} \quad & \sum_{t=1}^{n} x_{st} = 1 & \forall s\in\mathcal{S} \\
& \sum_{s\in\mathcal{S}} x_{st} = 1 & \forall t\in[n] \\
& y_{ut} \le \max_i\{g_{uit}\} & \forall u\in\mathcal{U}, t\in[n] \\
& z_{st} = \sum_{t'=1}^{t} x_{st'} & \forall t\in[n] \\
& g_{uit}(1-\sum_{s\in F} A_{uis}) \le (1-\sum_{s\in F} A_{uis}) \quad \sum_{s\in T_1(u,i,F)} z_{st} + \sum_{s\in T_2(u,i,F)} A_{uis}z_{st} & \\
& & \forall i\in[p], u\in\mathcal{U}, t\in[n], F\subseteq\mathcal{S}, \sum_{s\in F} A_{uis}\le 1 \\
& x_{st}, y_{ut}, z_{st}, g_{uit} \in [0,1] & \forall s\in\mathcal{S}, u\in\mathcal{U}, t\in[n], i\in[p]
\end{aligned}
$$

where $T_1(u,i,F) = \{s \mid A_{uis} \ge (1-\sum_{e\in F} A_{uie}), s\in\mathcal{S}\}$, $T_2(u,i,F) = \mathcal{S}\backslash T_1(u,i,F)$, and g_{uit} indicates if $f_{ui}(\{\text{the first } t \text{ selected documents}\}) \ge 1$.

This time we do not need to consider the set cover instances, and the k-th round of our algorithm can be:

- Let $t = 2^k$, $G_k = \emptyset$.
- For each $s\in\mathcal{S}$, add document s to G_k independently with probability $\min\{1, 50z^*_{st}\}$
- If there are more than $70\cdot 2^k$ documents in G_k, we say this round is "overflowed" and select nothing, else we select all the documents in G_k in arbitrary order in this round.

All the discussions are the same as in section 2.3 except there is no H_k, and in the case that $\sum_{s\in T_2(u,i,F)} A_{uis}z_{st} \ge \frac{3}{10}(1-\sum_{s\in F} A_{uis})$, we could have $\mathbb{E}\left[\sum_{s\in T_2(u,i,F)} A_{uis}\overline{z^*_{st}}\right] \ge 15(1-\sum_{s\in F} A_{uis})$ and use Lemma 6 directly.

Finally we can have the following theorem:

Theorem 10. *Suppose for each user, the XOS function contains only polynomial additive set functions which are non-negative. There is an $O(1)$-approximation for RXOS.*

4 Final Remarks

As we mentioned in the introduction, the real world document-topic instance do not form arbitrary set system and may be easier to approximate than the general combinatorial set cover problem. There is a huge literature on algorithms for classifying or clustering the documents and modeling document-topic relations. Many of those works leverage the underlining special structure of the document-topic instance to achieve good classification or clustering. It is an interesting further direction to explore the connections to those works and see whether the assumptions made or the special structures used in those works would translate to interesting set cover instances that are easier to approximate.

We could extend our model in several ways to capture other factors that may affect the search result. For example, we can capture that each user only has limited patience in the following

variant. For each user i, we have a patience level t_i that the maximum number of documents user i will examine. If the user does not get a relevant document after examining t_i documents, she will leave the system. Now the objective is to maximize the expected number of satisfied users. This generalizes the traditional scheduling problem with deadlines. We can also incorporate uncertainty into the user preferences. Namely, a user is interested in a particular document with a certain probability. The resulting stochastic version of the problem has a similar flavor with the sequential trial optimization defined in [18] or the stochastic matching problem in [16, 7].

Finally, we note that our approximation algorithm is mainly of theoretical interests since we need to use the ellipsoid algorithm to solve a linear program with exponential constraints, which is computationally expensive in practice. Hence, developing more efficient algorithms for RDC (even with worse performance guarantee) is of great practical interests.

Acknowledgements Jian Li would like thank Yossi Azar for a stimulating discussion about the RDC model and for providing the manuscript [3].

References

[1] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14. ACM, 2009.

[2] Christoph Ambühl, Thomas Erlebach, Matúš Mihalák, and Marc Nunkesser. Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 3–14. Springer, 2006.

[3] Y. Azar and I. Gamzu. Ranking with Unrelated Intents, 2010.

[4] Yossi Azar and Iftah Gamzu. Ranking with submodular valuations. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1070–1079. SIAM, 2011.

[5] Yossi Azar, Iftah Gamzu, and Xiaoxin Yin. Multiple intents re-ranking. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 669–678. ACM, 2009.

[6] Nikhil Bansal, Anupam Gupta, and Ravishankar Krishnaswamy. A constant factor approximation algorithm for generalized min-sum set cover. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1539–1545. Society for Industrial and Applied Mathematics, 2010.

[7] Nikhil Bansal, Anupam Gupta, Jian Li, Julián Mestre, Viswanath Nagarajan, and Atri Rudra. When lp is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4):733–762, 2012.

[8] Nikhil Bansal, Kamal Jain, Anna Kazeykina, and Joseph Seffi Naor. Approximation algorithms for diversified search ranking. In *Automata, Languages and Programming*, pages 273–284. Springer, 2010.

[9] Nikhil Bansal and Subhash Khot. Optimal long code test with one free bit. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 453–462. IEEE, 2009.

[10] Allan Borodin, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *Proceedings of the 31st symposium on Principles of Database Systems*, pages 155–166. ACM, 2012.

[11] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. A sharp concentration inequality with applications. *Random Structures & Algorithms*, 16(3):277–292, 2000.

[12] Hervé Brönnimann and Michael T Goodrich. Almost optimal set covers in finite vc-dimension. *Discrete & Computational Geometry*, 14(1):463–479, 1995.

[13] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336, 1998.

[14] Timothy M Chan and Sariel Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *Discrete & Computational Geometry*, 48(2):373–392, 2012.

[15] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding for matroid polytopes and applications. *arXiv preprint arXiv:0909.4348*, 2009.

[16] Ning Chen, Nicole Immorlica, Anna R Karlin, Mohammad Mahdian, and Atri Rudra. Approximating matches made in heaven. In *Automata, Languages and Programming*, pages 266–278. Springer, 2009.

[17] Z. Chen and T. Li. Addressing diverse user preferences in SQL-query-result navigation. In *SIGMOD*, pages 641–652, 2007.

[18] Edith Cohen, Amos Fiat, and Haim Kaplan. Efficient sequences of trials. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 737–746. Society for Industrial and Applied Mathematics, 2003.

[19] Marina Drosou and Evaggelia Pitoura. Search result diversification. *ACM SIGMOD Record*, 39(1):41–47, 2010.

[20] Guy Even, Dror Rawitz, and Shimon Moni Shahar. Hitting sets when the vc-dimension is small. *Information Processing Letters*, 95(2):358–362, 2005.

[21] Uriel Feige, László Lovász, and Prasad Tetali. Approximating min sum set cover. *Algorithmica*, 40(4):219–234, 2004.

[22] Matt Gibson and Imran A Pirwani. Algorithms for dominating set in disk graphs: breaking the logn barrier. In *Algorithms–ESA 2010*, pages 243–254. Springer, 2010.

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

[24] Martin Grötschel, László Lovász, and Alexander Schrijver. Geometric algorithms and combinatorial optimization. 1988.

[25] Refael Hassin and Asaf Levin. An approximation algorithm for the minimum latency set cover problem. In *Algorithms–ESA 2005*, pages 726–733. Springer, 2005.

[26] Sungjin Im, Viswanath Nagarajan, and Ruben van der Zwaan. Minimum latency submodular cover. In *Automata, Languages, and Programming*, pages 485–497. Springer, 2012.

[27] Sungjin Im, Maxim Sviridenko, and Ruben van der Zwaan. Preemptive and non-preemptive generalized min sum set cover. *Mathematical Programming*, pages 1–25, 2012.

[28] David Karger, Cliff Stein, and Joel Wein. Scheduling algorithms. *CRC Handbook of Computer Science*, 1997.

[29] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.

[30] C.D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press New York, NY, USA, 2008.

[31] Rajeev Motwani. *Randomized algorithms*. Cambridge university press, 1995.

[32] Nabil Hassan Mustafa and Saurabh Ray. Ptas for geometric hitting set problems via local search. In *Proceedings of the 25th annual symposium on Computational geometry*, pages 17–22. ACM, 2009.

[33] Saurav Pandit, Sriram V Pemmaraju, and Kasturi Varadarajan. Approximation algorithms for domatic partitions of unit disk graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 312–325. Springer, 2009.

[34] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *ICML*, pages 784–791, 2008.

[35] S.E. Robertson. The probability ranking principle in IR. *Journal of documentation*, 33(4):294–304, 1993.

[36] Martin Skutella and David P Williamson. A note on the generalized min-sum set cover problem. *Operations Research Letters*, 39(6):433–436, 2011.

[37] Kasturi Varadarajan. Weighted geometric set cover via quasi-uniform sampling. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 641–648. ACM, 2010.

[38] Vijay V Vazirani. *Approximation algorithms*. springer, 2001.

[39] Jun Wang and Jianhan Zhu. Portfolio theory of information retrieval. In *SIGIR*, pages 115–122, 2009.

View publication stats