Essential Web Pages Are Easy to Find

Ricardo Baeza-Yates*
Yahoo Labs
Barcelona
Spain
rbaeza@acm.org

Paolo Boldi†
Dipartimento di Informatica
Università degli Studi di Milano
Italy
paolo.boldi@unimi.it

Flavio Chierichetti‡
Dipartimento di Informatica
Sapienza Università di Roma
Italy
flavio@di.uniroma1.it

ABSTRACT

In this paper we address the problem of estimating the index size needed by web search engines to answer as many queries as possible by exploiting the marked difference between query and click frequencies. We provide a possible formal definition for the notion of *essential web pages* as those that cover a large fraction of distinct queries — *i.e.*, we look at the problem as a version of MAXCOVER. Although in general MAXCOVER is approximable to within a factor of $1 - 1/e \approx 0.632$ from the optimum, we provide a condition under which the greedy algorithm does find the actual best cover (or remains at a known bounded factor from it). The extra check for optimality (or for bounding the ratio from the optimum) comes at a negligible algorithmic cost. Moreover, in most practical instances of this problem, the algorithm is able to provide solutions that are provably optimal, or close to optimal. We relate this observed phenomenon to some properties of the queries' click graph. Our experimental results confirm that a small number of web pages can respond to a large fraction of the queries (*e.g.*, 0.4% of the pages answers 20% of the queries). Our approach can be used in several related search applications, and has in fact an even more general appeal — as a first example, our preliminary experimental study confirms that our algorithm has extremely good performances on other (social network based) MAXCOVER instances.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Information Storage and Retrieval; G.2.1 [**Combinatorics**]: Discrete Mathematics; G.1.6 [**Optimization**]: Numerical Analysis.

Keywords

Web search; query log analysis; layered indices; tiering; max cover; click graph; approximation algorithms; greedy algorithms.

1. INTRODUCTION

Scalability is one of the main issues for web search engines, particularly with respect to query load and index size. In this paper we address the latter problem in a very general sense: it is obvious that having to deal with a smaller index implies the need of less resources and hence is less costly. Fortunately, query frequencies and the frequency of clicks on web pages that answer those queries follow very different power laws, and hence an index that contains a small fraction of all available web pages may answer well, as shown later, a large fraction of the query volume.

In fact, we can always view a search engine as logically composed by two indices. A small (main) index built from the essential web pages that answers the bulk of the queries, and another (secondary) larger index that is tailored to answer long tail queries (see [4, Section 11.4.4]). Here we consider the problem of selecting the web pages for the main index leaving the design of the secondary index for future work.

Formally, we address the following *essential web pages*[1] problem. Given a set Q of n queries and a set of m web pages P, that form a (bipartite) "relevance" graph (for some notion of relevance of a web page to a query, typically derived from user clicks), find a subset of pages $S \subseteq P$ of size K that maximizes the coverage of queries. We say that a query is *covered* if at least one of the pages that are relevant to it is in S, though, in general, one might consider different definitions of coverage as we will see later. In this work we aim to maximize the number of distinct covered queries, though it would also make sense to maximize their total volume (we discuss this case in the conclusions).

We use the following definition of *relevance*: we restrict P to the set of pages that users clicked on sufficiently enough (*i.e.*, at least a certain number of times) as a result of a search. That is, a page is relevant for a query if the ranking algorithm of the web search engine believes it is a relevant page and enough users agree. Our choice gives a very strong notion of relevance.

Notice that to solve the problem outlined above, in some sense all available pages need to be indexed. However, as this problem can be solved off-line, we do not need to construct the index to obtain a good estimation of the top ranked pages for every query. Moreover,

[*] This research started when the second author was visiting Yahoo Labs Barcelona, and was performed using Yahoo's data and computation infrastructure.

WWW 2015, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3469-3/15/05.
http://dx.doi.org/10.1145/2736277.2741100.

† Partially supported by the Yahoo Faculty Research and Engagement Program and by the MIUR-PRIN National Project "ARS TechnoMedia" (Algorithmics for Social Technological Networks).
‡ Partially supported by a Google Faculty Research Award and by the MIUR-PRIN National Project "ARS TechnoMedia" (Algorithmics for Social Technological Networks).

[1] We remark that our notion of "essential" has nothing to do with that described in [20], as better explained in the "Related work" section.

we discuss later that having the index allows to find solutions for specific instances that are interesting in their own right.

The essential web pages problem can be seen as a MAXCOVER [14] problem. Hence, it is in general NP-hard, although it is approximable to within a factor of $1 - 1/e$ from the optimum using a greedy approach [21]. In this paper, we show that, however, in many practical instances of our problem, the greedy algorithm in fact finds the *optimal* cover, or a cover which is close to optimal.

Our approach is based on the observation that the value of the optimal solution of MAXCOVER can be sandwiched from below by the greedy solution and from above by its relaxed LP formulation. It is known that the ratio of the two sides of the sandwich is always lower bounded by $1 - 1/e$. The gap is tight in some cases, in the sense that one can build instances that are as close as one wants to the above bound. By taking the dual of the relaxed MAXCOVER LP, one obtains a linear minimization program whose feasible solutions' values are upper bounds to the integral MAXCOVER problem. Therefore, the dual problem can be used to provide bounds on the approximation that the greedy algorithm produces: this is explained in Section 5. The computational cost of producing this bound is negligible, and the bound obtained is in many cases enough to prove that the greedy output is actually *optimal*, or very close to optimality. In the experiments on our largest click-graph dataset, for example, the bound produced was never worse than 98%; a simpler baseline depending on cardinality sum would yield much worse bounds (ranging from 87% to 93%). In fact, even our bound is too pessimistic: the real approximation ratio of the greedy algorithm for the click-graph datasets on which the exact ratio could be produced was never worse than 99.79%! From a practical standpoint, we observe that adding answer and inverted list caching would make these results even better.

We have also observed very good performances on a non web-index based dataset. We ran our same algorithm on a Twitter based dataset (having users as elements, and a set for each user u that contains u and all the users followed by u), obtaining slightly different, but still extremely good, guaranteed performances. Indeed, the bounds we obtained are not worse than 95% for each $K \geq 50$. It then appears that our algorithm can be used in a variety of settings, without decreasing its extremely good approximation guarantees.

More generally, it is striking to observe that our modified greedy algorithm on the instances of our problem (Section 7) guarantees an approximation that is (provably) extremely good, less than 2.5% away from the optimum when $K \leq 500$ (and less than 5% for $K \leq 140,000$). One might suspect that this extremely good approximation is due to the power-law distributions involved in our instances. However, we prove in Sections 3 and 4 that the $1 - 1/e$ bound cannot be improved on some of these instances.

While a satisfying explanation of this phenomenon is still missing, in Section 6 we provide a proof that the greedy algorithm produces with high probability an almost optimal solution provided that the following three conditions are satisfied:

1. the greedy algorithm is able to include at least $\Omega(\log n)$ queries at every step;
2. there are many queries only included in constantly many web pages; and
3. the instance is chosen uniformly at random, between those satisfying the given distributions.

Note that condition (2) is guaranteed if the queries' clicks follow a power-law distribution, as it happens in practice.

Our final coverage results are quite good. In fact, in our largest dataset, almost 25% of the queries are covered by just 0.7% of the web pages (Section 7). In the Conclusions we discuss our results and give other search related problems that can profit from this powerful imbalance.

2. RELATED WORK

Many recent papers focus their attention on various possible search scenarios that can lead to (variants of the) MAXCOVER problem, and analyze the issues they imply. Examples of applications include influence maximization in social networks [8, 16], whereas in the context of web search it was used to formalize the problem of discoverability of new content [13]. In the latter paper, they overcome the inherent difficulty of the problem using past observations. In fact, various query/documents covering problems have been studied in the literature (*i.e.*, index tiering [17, 5], stochastic query covering [1]).

All the cited scenarios deal with very large datasets, which raises the question of whether the greedy algorithm is amenable to being implemented in a MapReduce framework [9]. One interesting aspect of the algorithm we are presenting in this paper is that we can modify also the MapReduce-greedy algorithm given in [9] to obtain a behavior similar to the one that we describe here for the sequential case, at the expense of a reasonable increase in the number of the MapReduce iterations required.

Another stream of related research is the so-called pruned indices. Pruned indices are smaller indices where term and/or document related information is deleted to use less space. The most recent results show that caching is more effective than index pruning to improve answering time [19]. Nevertheless, the initial motivation of index pruning was to use less space [7], but the techniques used did not force that all references to a given document would need to disappear from the index.

A notion of "essential" page was introduced in a quite different context in [20], but with another meaning: they are not interested in covering queries but rather knowledge (terms), and they express this need as a form of coverage problem. Their problem is different from the vanilla (*i.e.*, unweighted) MAXCOVER problem. They do use a greedy approach to solve it, but do not provide bounds on the quality of the solution obtained. In fact, [11] considers a general problem that includes both MAXCOVER, and the problem in [20], as special cases. The authors of [11] provide an algorithm that returns a $1-1/e+\varepsilon$ approximation in time $O(poly(n) \cdot m^{1/\,poly(\varepsilon)})$; they also provide a variant of greedy for the same problem, but with larger approximation ratios.

3. MAX COVER

MAXCOVER is a well-known NP-hard problem [14] that consists in finding K sets in a given collection so as to maximize their union.

In our intended application items are queries, sets correspond to URLs,[2] and a query belongs to the set of a given URL if that URL is deemed "relevant" for the query. The actual definition of what relevant means is not really important here: in the experimental part of this paper (Section 7) we used a number of query logs of a search engine and extracted the click graph [12], to determine on which URLs people clicked after submitting a query. Other possible variants on the exact definition of the problem are discussed in the conclusions.

Formally, the MAXCOVER problem can be stated as follows:
Problem 1: MAXCOVER.
Input: An instance $\Pi = (\Omega, \mathcal{S}, K)$ is defined by a set Ω of n

[2] We use URL (or document) as a synonym of web page.

items, a family $\mathcal{S} = \{S_1, \dots, S_m\}$ of m subsets of Ω and an integer K.
Output: Find a subset $\mathcal{S}' \subseteq \mathcal{S}$ with $|\mathcal{S}'| \leq K$ maximizing

$$\textsc{Opt}(\Pi) = \left| \bigcup_{S \in \mathcal{S}'} S \right|.$$

From time to time, we shall look at the pair $(\Omega, \mathcal{S})$ as a bipartite graph, with n vertices on the left-hand side (hereafter called *q-vertices*, where "q" stands for "queries", because items are queries) representing the items Ω (queries), and m vertices on the right-hand side (hereafter called *d-vertices*, where "d" stands for "documents") representing the sets $\mathcal{S}$ (web pages); edges represent membership of items to sets, as expressed by the notion of "relevance" adopted.

The degree sequence of q-vertices (*i.e.*, the list of their degrees sorted in non-decreasing order) will be called *q-sequence* ("q" stands for "query degree"), and similarly the degree sequence of d-vertices will be called *d-sequence* (for "document degree"). Since we observed (Section 7) that both sequences are power-law of exponents larger than 2, we are especially interested in dealing with this case.

The MaxCover problem can be formulated [21] as an integral linear programming problem with $n + m$ variables and $2n + 1$ constraints as follows: take one variable x_i (with $1 \leq i \leq n$) for every item and one variable y_j (with $1 \leq j \leq m$) for every set, where all variables are constrained to be non-negative (ideally, in $\{0, 1\}$) and $x_i = 1$ means that the i-th item is covered, whereas $y_j = 1$ means that the j-th set is output.

With this notation, we can formulate MaxCover as:

$$\begin{aligned}
&\max \left(\sum_{i=1}^{n} x_i \right) \qquad \textbf{subject to} \\
&\qquad \sum_{j=1}^{m} y_j \leq K \\
&x_i - \sum_{S_j \ni i} y_j \leq 0 \quad \text{for } i = 1, \dots, n \\
&\qquad x_i, y_j \in [0, 1] \quad \text{for } i = 1, \dots, n \text{ and } j = 1, \dots, m.
\end{aligned} \tag{1}$$

This LP model will be referred to as *MCLP* (for "MaxCover LP") if x_i and y_j are constrained to be integral. The first constraint simply means that we want to take at most K sets, the second i constraints impose that we can say that x_i is covered only if at least one of the sets containing it is taken, whereas the final set of constraints impose that x_i is zero or one.

If we relax the constraint that x_i and y_j be integral, we obtain a new LP model that we refer to as *relaxed MCLP*: an interpretation of this relaxed version is that a set can be "partially taken" (when y_j is strictly between 0 and 1), and hence an item can be "partially covered".

Integrality gap (general case).

We shall use $\text{LP}(\Pi)$ to denote the value of the objective for the relaxed MCLP corresponding to the instance Π. Of course $\textsc{Opt}(\Pi) \leq \text{LP}(\Pi)$. Actually the ratio between the two (called the "integrality gap") satisfies

$$1 - \frac{1}{e} \leq \frac{\textsc{Opt}(\Pi)}{\text{LP}(\Pi)} \leq 1.$$

The lower bound is tight, in the sense that for every $\varepsilon > 0$ there is an instance whose integrality gap is less than $1 - \frac{1}{e} + \varepsilon$. This is obtained with a classical [21] construction that we shall now describe, since we shall modify it later on.

We shall build an instance Π_M for every integer $M \geq 2$; the instance Π_M has M^2 sets $\mathcal{S} = \{S_1, \dots, S_{M^2}\}$ and Ω is made of $\binom{M^2}{M}$ items: for every choice of M sets out of the M^2 available there is one item that is put in exactly those sets. If you fix $K = M$, the optimal solution to MaxCover will have value

$$\textsc{Opt}(\Pi_M) = \binom{M^2}{M} - \binom{M^2 - M}{M},$$

because whichever class of $K = M$ sets you choose, all the items that were put in a disjoint class of M sets not overlapping them will not be covered. On the other hand, the optimal solution of the primal (1) for this instance is

$$\text{LP}(\Pi_M) = \binom{M^2}{M}.$$

The fact that $\binom{M^2}{M}$ is an upper bound follows trivially from $|\Omega| = \binom{M^2}{M}$. We give a feasible solution of (1) that achieves that value, thus proving our assertion. We set $y_j = M^{-1}$ for each set S_j. By definition, each item i is contained in exactly M sets, and we can therefore choose $x_i = 1$ without violating any constraint.

The integrality gap $\textsc{Opt}(\Pi_M)/\text{LP}(\Pi_M)$ is then bounded as follows:[3]

$$\begin{aligned}
\frac{\textsc{Opt}(\Pi_M)}{\text{LP}(\Pi_M)} &= 1 - \frac{\binom{M^2-M}{M}}{\binom{M^2}{M}} \\
&= 1 - \frac{(M^2 - M) \cdots (M^2 - 2M + 1)}{M^2 \cdots (M^2 - M + 1)} \\
&\leq 1 - \left(\frac{M^2 - 2M}{M^2 - M} \right)^M = 1 - \left(1 - \frac{1}{M - 1} \right)^M \\
&\leq 1 - \frac{1}{e} + O\left(\frac{1}{M} \right).
\end{aligned}$$

Integrality gap (power-law case).

We want to show that the above construction can be modified so to prove that the lower bound is tight also for instances whose d-sequence is a power law distribution of exponent $\alpha > 2$.

Let us use Π_M to denote the instance described above. Observe that, by Stirling's approximation, the number of items $\binom{M^2}{M}$ in the instance Π_M is $\Theta\left(M^{M-\frac{1}{2}} e^M\right)$. Moreover, each set has cardinality $\binom{M^2-1}{M-1} = \Theta\left(M^{M-\frac{3}{2}} e^M\right)$.

Now suppose that we aim to create an instance Π' with a d-distribution (that is, a set-cardinality distribution) following a power law with some constant exponent $\alpha > 2$. Let $\zeta(\alpha) = \sum_{i=1}^{\infty} i^{-\alpha}$. For every integer M, the number of sets with cardinality $\binom{M^2-1}{M-1}$ needs to be at least

$$\frac{t}{\zeta(\alpha) \cdot \binom{M^2-1}{M-1}^{\alpha}} - O(1) \geq \Omega\left(t \cdot M^{\frac{3}{2}\alpha - M\alpha} e^{-M\alpha}\right),$$

where t is the number of sets in Π'.

For any given t, if we choose M to be an integer such that $t = \Theta\left(M^{M\alpha - \frac{3}{2}\alpha + 2} e^{M\alpha}\right)$ we are sure that in the instance Π' that we

[3] We are using the fact that $1 < a < b$ implies $\frac{a}{b} > \frac{a-1}{b-1}$, so $\frac{M^2-M-i}{M^2-i} > \frac{M^2-2M}{M^2-M}$.

construct, there will be at least M^2 sets of cardinality $\binom{M^2-1}{M-1}$. Let us use $\mathcal{S}$ to denote an arbitrary class of M^2 sets of cardinality $\binom{M^2-1}{M-1}$ in Π'.

Now, the largest set in Π' will have cardinality at most $O(t^{\frac{1}{\alpha}})$. We let all sets in $\Pi' - \mathcal{S}$ be subset of the largest set in Π'. Then, the total contribution of the sets not in $\mathcal{S}$ to a solution (whether integral or fractional) is at most

$$O\left(t^{\frac{1}{\alpha}}\right) = O\left(M^{M-\frac{3}{2}+\frac{2}{\alpha}}e^M\right).$$

As we already showed, if we choose $K = M$, then the solution composed of the sets in $\mathcal{S}$ has value $\Theta(n) = \Theta\left(M^{M-\frac{1}{2}}e^M\right)$. We have that $\Theta\left(\frac{t^{\frac{1}{\alpha}}}{n}\right) = O\left(M^{-1+\frac{2}{\alpha}}\right)$. Since $\alpha > 2$, the latter is $o(1)$. Therefore, the integrality gap of Π' can be upper bounded by the integrality gap of Π_M times at most $1 + o(1)$ — that is, it can be upper bounded by $1 - \frac{1}{e} + o(1)$.

4. THE DUAL PROBLEM AND ITS RELATION WITH THE GREEDY SOLUTION

In this section, we present a classical heuristic for MAXCOVER and discuss how good is the approximation ratio it provides. For this purpose, it is useful to see it in relation with the dual LP problem.

The dual of (1) is a new LP with $2n+1$ variables T, z_i, w_i, and u_j (with $1 \leq i \leq n$ and $1 \leq j \leq m$) and $n+m$ constraints:

$$\min\left(K \cdot T + \sum_{i=1}^{n} z_i\right) \qquad \textbf{subject to}$$
$$\begin{aligned} z_i + w_i &\geq 1 \quad \text{for } i = 1, \ldots, n, \\ T - \sum_{x_i \in S_j} w_i &\geq 0 \quad \text{for } j = 1, \ldots, m, \\ z_i, w_i &\geq 0 \quad \text{for } i = 1, \ldots, n. \end{aligned} \tag{2}$$

Now, let us take an optimal solution of (2) and modify it as follows.

1. Observe that optimality guarantees that $z_i \leq 1$ for each $i = 1, \ldots, n$. Indeed, if $z_i > 1$, by assigning the value 1 to z_i, we decrease the value of the objective, and we keep every constraint satisfied.

2. For every index i, if $z_i + w_i > 1$, we assign to w_i the new value $w'_i = \max(0, 1 - z_i)$. We show that this substitution keeps intact the feasibility of the system. Observe that $w_i > 1 - z_i \geq \max(1 - z_i, 0) = w'_i$, so $w'_i < w_i$, and the substitution never breaks constraints from the second group; moreover, $z_i + w'_i = \max(1, z_i) \geq 1$, so the constraints from the first group will also be satisfied, and the solution will still be feasible.

In other words, under optimality we can assume that $w_i = 1 - z_i \in [0,1]$, so the optimal solutions of (2) can be turned into optimal solutions of a simpler LP problem with $n+1$ variables only:

$$\min\left(K \cdot T + \sum_{i=1}^{n} z_i\right) \qquad \textbf{subject to}$$
$$\begin{aligned} T &\geq |S_j| - \sum_{x_i \in S_j} z_i \quad \text{for } j = 1, \ldots, m, \\ z_i &\in [0,1] \quad \text{for } i = 1, \ldots, n. \end{aligned} \tag{3}$$

The optimal value of (3) on a given instance Π of the problem will be denoted by DLP(Π); by the duality theorem

$$\text{DLP}(\Pi) \geq \text{LP}(\Pi) \geq \text{OPT}(\Pi).$$

In other words, solving either LP problems (either the primal or the dual) provides an upper bound to the value of the optimal solution, and as proved in Section 3 the gap between the optimum and the upper bound can be as large as $1 - 1/e$.

Sub-optimality of greedy algorithm (general case).

The *greedy solution* of the MAXCOVER problem is found by selecting iteratively the set (one of the sets) that cover the maximum number of yet-uncovered items, until K sets are selected. We write GREEDY(Π) for the number of items covered by the greedy solution on the instance Π.

Greedy solutions are sub-optimal, that is (looking at the overall picture) DLP(Π) $\geq$ LP(Π) $\geq$ OPT(Π) $\geq$ GREEDY(Π) but it is known that [15]

$$1 - \frac{1}{e} \leq \frac{\text{GREEDY}(\Pi)}{\text{OPT}(\Pi)} \leq 1,$$

with the lower bound being once more tight (*i.e.*, there are instances for which we can make the ratio as close to the lower bound as we want).

We now describe a classical construction that shows the tightness of the lower bound. Fix $K \geq 2$, and choose an integer $t \geq 1$. We create an instance $\Pi = \Pi(K, t)$. For each $i = 1, \ldots, K$ and $j = 0, \ldots, Kt - 1$, let us create a set Ω_i^j containing $K^{Kt-1} \cdot \left(\frac{K-1}{K}\right)^j$ unique elements. Let $\Omega = \bigcup_{i,j} \Omega_i^j$, $\Omega_i = \bigcup_j \Omega_i^j$ and $\Omega^j = \bigcup_i \Omega_i^j$.

We let the sets of the instance be Ω_i, for $i = 1, \ldots, K$, and Ω^j, for $j = 0, \ldots, Kt - 1$. Observe that picking the sets $\Omega_1, \ldots, \Omega_K$ is an optimal solution, having value:

$$\begin{aligned} |\Omega| &= \sum_{i=1}^{K} |\Omega_i| = K \cdot K^{Kt-1} \cdot \sum_{j=0}^{Kt-1} \left(\frac{K-1}{K}\right)^j \\ &= K^{Kt+1} \cdot \left(1 - \left(1 - \frac{1}{K}\right)^{Kt}\right) \geq \left(1 - e^{-t}\right) K^{Kt+1}. \end{aligned}$$

On the other hand, each Ω^j is larger in cardinality than any Ω_i, so the greedy algorithm will pick the sets $\Omega^0, \ldots, \Omega^{K-1}$ in this order, for a total value:

$$\begin{aligned} \left|\Omega^0 \cup \ldots \cup \Omega^{K-1}\right| &= K \cdot K^{Kt-1} \cdot \sum_{j=0}^{K-1} \left(\frac{K-1}{K}\right)^j \\ &= K^{Kt+1} \cdot \left(1 - \left(1 - \frac{1}{K}\right)^K\right). \end{aligned}$$

Therefore, the ratio between the value of the solutions produced by the greedy algorithm and the optimal solution is bounded by

$$\frac{\text{GREEDY}(\Pi)}{\text{OPT}(\Pi)} \geq \frac{1 - \left(1 - \frac{1}{K}\right)^K}{1 - e^{-t}}.$$

By selecting $t = t(K) = \lceil \ln K \rceil$, we obtain that the ratio can be upper bounded by $1 - \frac{1}{e} + O\left(\frac{1}{K}\right)$. Hence, the ratio converges to $1 - \frac{1}{e}$ as $K \to \infty$.

Sub-optimality of greedy algorithm (power-law case).

We once again transform the above instance $\Pi(K, \lceil \ln K \rceil)$ into an instance having a power law d-distribution (*i.e.*, cardinality distribution) with exponent $\alpha > 1$. We will build the new instance

Π' so that it contains $m = \Theta\left(K^{\alpha K \lceil \ln K \rceil + 1}\right)$ sets. An easy calculation shows that it will contain at least K sets of cardinality $|\Omega_1| = |\Omega_2| = \ldots = |\Omega_K|$, and at least 1 set for each of the cardinalities $\left|\Omega^0\right|, \left|\Omega^1\right|, \ldots, \left|\Omega^{K \lceil \ln K \rceil - 1}\right|$. We will use these sets to recreate the instance Π within Π'.

The largest set in Π' will have size $\Theta\left(K^{K \lceil \ln K \rceil + \frac{1}{\alpha}}\right)$, and we build all the other sets of Π' so that they are subsets of this largest set.

Now, picking the sets $\Omega_1, \ldots, \Omega_K$ will still cover at least $\left(1 - \frac{1}{K}\right) \cdot K^{K \lceil \ln K \rceil + 1}$ elements. The greedy algorithm, on the other hand, will cover the set of maximum size – which will account for $\Theta\left(K^{K \lceil \ln K \rceil + \frac{1}{\alpha}}\right)$ elements – and, as in the instance Π, will necessarily cover at most $K^{K \lceil \ln K \rceil + 1} \cdot \left(1 - \left(1 - \frac{1}{K}\right)^K\right)$ other elements.

Therefore, $\textsc{Greedy}(\Pi')/\textsc{Opt}(\Pi') \to 1 - \frac{1}{e}$, as $K \to \infty$.

Sub-opt. of greedy algorithm (double power-law).

The previous proof still leaves some margin of hope: the tightness of the bound may fail to hold if we require both distributions to be power-law. A more complex construction, however, shows that even in this case the usual bound holds true:

Lemma 1 *Suppose that $\alpha > 2$ and $\beta > \alpha + 1$.*[4] *Then, we can create instances Π' of increasing sizes $n = \Theta(m)$, having a q-degree α-power law distribution, and a d-degree β-power law distribution, such that*

$$\frac{\textsc{Greedy}(\Pi')}{\textsc{Opt}(\Pi')} \xrightarrow{n \to \infty} 1 - \frac{1}{e}.$$

PROOF. Let Π be any instance that has the given q-degree and d-degree distributions.

Observe that, since the sum of the degrees has to be equal to the sum of the cardinalities, and since $\alpha, \beta > 2$, Π satisfies $n = \Theta(m)$.

Let the largest cardinality be u; then $u = \Theta\left(n^{\frac{1}{\beta}}\right)$. Let $\mathcal{T} \subseteq \mathcal{S}$ be the subclass of sets having cardinality at least $\varepsilon \cdot u$, for an unspecified constant $\varepsilon = \varepsilon(\alpha, \beta) > 0$. We have $|\mathcal{T}| = \Theta\left(n^{\frac{1}{\beta}}\right)$. Also, let K be the largest integer such that there are at least K sets of cardinality at least $K^{K \lceil \ln K \rceil} \cdot \left(1 - \left(1 - \frac{1}{K}\right)^{K \lceil \ln K \rceil}\right)$. Then, $K = \Theta\left(\frac{\ln n}{\ln^2 \ln n}\right)$.

Let $\mathcal{T}' \subseteq \mathcal{T}$ be the subclass of sets having the largest K cardinalities in $\mathcal{T}$ (breaking ties arbitrarily). We have $\max_{S \in \mathcal{T}'} |S| = u$, and $\min_{S \in \mathcal{T}'} |S| \leq u - K$.

We then create another subclass of sets $\mathcal{U}$: this class will contain $K \lceil \ln K \rceil$ sets, and will be disjoint from $\mathcal{T}$. Specifically, for each $j = 1, \ldots, K \lceil \ln K \rceil - 1$, we put in $\mathcal{U}$ exactly one set S_j of cardinality $K^{K \lceil \ln K \rceil} \cdot \left(\frac{K-1}{K}\right)^j$. Moreover, $\mathcal{U}$ will contain one final set S_0 of cardinality $\sum_{S \in \mathcal{T}'} |S| - \sum_{j=1}^{K \lceil \ln K \rceil - 1} K^{K \lceil \ln K \rceil} \cdot \left(\frac{K-1}{K}\right)^j$.

For each set $S \in \mathcal{T} \cup \mathcal{U}$, and for each element $e \in S$, (i) create a new set $S' = S'(S, e)$, (ii) remove e from S, and (iii) add e to S' (which will then have cardinality 1). Observe that, at the end of this process, no old item degree changes; all the sets that had cardinality at least εu are now empty, and we have introduced at most $\sum_{i=\varepsilon u}^{\infty} \left(i^{-\beta} \cdot i \cdot m\right) + u \cdot K = O\left(n^{\frac{2}{\beta}}\right)$ new sets of cardinality 1. Since the number of sets that had cardinality 1 was $\Theta(n)$, the d-degree power-law distribution is preserved at 1.

As a second step, create a set X of $\sum_{S \in \mathcal{T}'} |S| = \Theta(u \cdot K)$ new elements — the elements in X will end up having degree 2 in our construction. That is: (i) assign each element $x \in X$ to exactly one set $S \in \mathcal{T}'$, so that each set gets a number of elements equal to its original cardinality; also (ii) assign each element $x \in X$ to exactly one set $S \in \mathcal{U}$, so that each set gets a number of elements equal to its original cardinality. In this step we also add $\Theta(u \cdot K) = o(n)$ new elements of degree 2 — again, since the number of degree 2 elements in the q-degree distribution was $\Theta(n)$, the distribution is preserved.

The third and final step of our construction fills up the sets in $\mathcal{T} - \mathcal{T}'$. Pick the largest such set S, and fill it with new elements. Make all the other sets in $\mathcal{T} - \mathcal{T}'$ subsets of S. Observe that there will be $|S| = O\left(n^{\frac{1}{\beta}}\right)$ new elements, and that their degree will be at most $|\mathcal{T}| = O\left(n^{\frac{1}{\beta}}\right)$.

The q-sequence guarantees that, for each $1 \leq d \leq O\left(n^{\frac{1}{\beta}}\right)$, the number of items having degree d is at least $\Omega\left(\frac{n}{d^\alpha}\right) \geq \Omega\left(n^{1 - \frac{\alpha}{\beta}}\right)$. We add at most $O\left(n^{1/\beta}\right)$ new nodes with that degree; since $\beta - \alpha > 1$, we have that we add at most an $o(1)$ fraction of new nodes of degree d, for each possible d. The q-sequence is therefore preserved.

Finally, let Π' be the new instance. Observe that it follows the original q-degree power law, and d-degree power law, distributions. Observe also that greedy will pick the largest K sets in $\mathcal{U}$ (plus at most one set in $\mathcal{T} - \mathcal{T}'$), while the optimal solution would pick the sets in $\mathcal{T}'$. An easy calculation then shows the result. $\square$

The instance showing the above bound is created by "embedding" the classical construction described in the previous subsection into an instance with degrees and cardinalities distributed like power laws. This has to be done in a very careful way: first we show that the largest sets in the power law instance are good enough to contain the elements in the largest sets of the classical instance. Then, we need to ensure that the other large sets in the power law instance will not change by more than a $(1 + o(1))$ factor from the value of the greedy, and optimal, solutions. To do so, we include all those sets in one large set — this, of course, changes the degree distribution: the last part of our proof is then showing that the distribution does not change significantly and, in fact, still follows the same power law.

5. BOUNDING THE APPROXIMATION OF THE GREEDY SOLUTION

The previous section does not leave much room for hope: apparently even having an instance of MAXCOVER that has a power-law distribution of degrees on both sides may cause greedy to work as badly as it can. Also, on the other hand, the LP formulations do not help much, because they are themselves away from the optimum (in the other direction) for a large gap.

Nevertheless, we will be able to prove that greedy can exploit the dual problem to "certify" the quality of its own output, that is, to provide a bound on how far the solution is from the optimum. This property will allow us to modify the greedy algorithm so to make it produce this certificate along with the solution at a moderate computational cost (in many cases, asymptotically at no cost).

The basic property is stated in the following theorem; albeit seemingly unfathomable, we shall see how this result can be put at good use in a self-certifying version of the standard greedy algorithm.

[4] Although in our data $\beta > \alpha + 0.7$, this lemma seems to still hold.

Theorem 1 *For an integer $t \geq 1$, we say that a set is t-large if its cardinality is at least t, and that an item is t-certifying if it is contained in at most one t-large set.*
Consider an instance Π and let $S_1^*, \ldots, S_K^*$ be a solution (that is, a sequence of sets) produced by greedy; define

$$\gamma = \left| S_K^* \setminus \bigcup_{j=1}^{K-1} S_j^* \right|,$$

that is, the "gain" produced by the last set. For all $j = 1, \ldots, K-1$, let $\ell_j \geq 0$ be the smallest integer such that the number of γ-certifying items contained in S_j^* is at least $\gamma - \ell_j$. Then,

$$\mathrm{DLP}(\Pi) \leq \textsc{Greedy}(\Pi) + \sum_{j=1}^{K-1} \ell_j$$

and therefore $S_1^*, \ldots, S_K^*$ is a solution of MaxCover *with an additive error of at most $\sum_{j=1}^{K-1} \ell_j$.*

Proof. Observe that, necessarily, for each $j = 1, \ldots, K$, we have $|S_j^*| \geq \gamma$ — that is, all the S_j^*'s are γ-large. Therefore, if x is a γ-certifying item, then x can be in at most one set S_j^*, $1 \leq j \leq K$.

Now, for each $j = 1, \ldots, K-1$, let T_j^* be equal to any subset of S_j^*, of cardinality $\gamma - \ell_j$, containing only γ-certifying items (observe that T_j^* is well-defined because of the assumption in the claim). By definition, the T_j^*'s are pairwise disjoint.

Consider the dual (2) and let

$$z_i = \begin{cases} 1 & \text{if } x_i \in \bigcup_{j=1}^{K-1} \left(S_j^* \setminus T_j^*\right) \\ 0 & \text{otherwise.} \end{cases}$$

Since $T_j^* \subseteq S_j^*$ and since the T_j^*'s are pairwise disjoint, we can write

$$\begin{aligned} \sum_{i=1}^{n} z_i &= \left| \bigcup_{j=1}^{K-1} \left(S_j^* \setminus T_j^*\right) \right| = \left| \bigcup_{j=1}^{K-1} S_j^* \right| - \left| \bigcup_{j=1}^{K-1} T_j^* \right| \\ &= \left| \bigcup_{j=1}^{K-1} S_j^* \right| - \gamma \cdot (K-1) + \sum_{i=1}^{K-1} \ell_j. \end{aligned}$$

The value of the objective function of the dual is then $\left| \bigcup_{j=1}^{K-1} S_j^* \right| - \gamma \cdot (K-1) + K \cdot T + \sum_{j=1}^{K-1} \ell_j$.

By the definition of γ, we have $\left| \bigcup_{j=1}^{K-1} S_j^* \right| + \gamma = \left| \bigcup_{j=1}^{K} S_j^* \right|$. Therefore, the value of the objective function of the dual is

$$\left| \bigcup_{j=1}^{K} S_j^* \right| + K \cdot (T - \gamma) + \sum_{j=1}^{K-1} \ell_j.$$

We prove that setting $T = \gamma$ gives a feasible solution for the dual, *i.e.*, we show that it satisfies every constraint in the dual program: $\gamma = T \geq |S| - \sum_{x_i \in S} z_i$ for each set $S \in \mathcal{S}$ of the instance. We rewrite the constraint as:

$$\gamma \overset{?}{\geq} |S| - \left| S \cap \bigcup_{j=1}^{K-1} \left(S_j^* \setminus T_j^*\right) \right|.$$

If $S = S_j^*$, for some $j = 1, \ldots, K-1$, then the constraint is trivially satisfied, since it simplifies to $\gamma \overset{?}{\geq} |T_j^*| = \gamma - \ell_j$. We therefore assume that S differs from each of the $S_1^*, \ldots, S_{K-1}^*$ sets. Moreover, we can assume that $|S| > \gamma$, because otherwise once more the constraint will be trivially true.

Observe that $\cup_{j=1}^{K-1} T_j^*$ has empty intersection with S — indeed each node in the former set is part of exactly one γ-large set between $S_1^*, \ldots, S_{K-1}^*$ and cannot therefore be part of a second γ-large set S. Thus, the constraint can be rewritten as:

$$\gamma \overset{?}{\geq} |S| - \left| S \cap \bigcup_{j=1}^{K-1} S_j^* \right| = \left| S - \bigcup_{j=1}^{K-1} S_j^* \right|.$$

The latter cannot be larger than γ, otherwise the greedy algorithm would have picked S instead of $S_K^* \neq S$.

Hence, we found a feasible solution of the dual whose objective is $\left| \bigcup_{j=1}^{K} S_j^* \right| + \sum_{j=1}^{K-1} \ell_j$. Since the solution of value $\left| \bigcup_{j=1}^{K} S_j^* \right|$ produced by the greedy algorithm is feasible in the primal, we obtain

$$\textsc{Greedy}(\Pi) + \sum_{j=1}^{K-1} \ell_j = \left| \bigcup_{j=1}^{K} S_j^* \right| + \sum_{j=1}^{K-1} \ell_j \geq \mathrm{DLP}(\Pi). \quad \Box$$

A special case of Theorem 1 can in fact provide a guarantee of optimality: if for all $j = 1, \ldots, K-1$, the number of γ-certifying items contained in S_j^* is at least γ, then all ℓ_j's are zero, and the solution produced by the greedy algorithm is optimal.

We can turn the additive bound of Theorem 1 into a multiplicative bound (or, if you prefer, into a bound on the approximation ratio):

Corollary 1 *If Π satisfies the hypothesis of Theorem 1 then*

$$\frac{1}{1 + \frac{\sum_{j=1}^{K-1} \ell_j}{\textsc{Greedy}(\Pi)}} \leq \frac{\textsc{Greedy}(\Pi)}{\textsc{Opt}(\Pi)} \leq 1.$$

Proof. Just recall that $\textsc{Greedy}(\Pi) + \sum_{j=1}^{K-1} \ell_j \geq \mathrm{DLP}(\Pi)$ and the latter is an upper bound for $\textsc{Opt}(\Pi)$. $\Box$

Note that the actual ratio can be much better than the one obtained by Corollary 1 simply because the upper bound of Theorem 1 is not tight. If the instance is small enough, one can try to solve the dual LP to obtain a better bound (or even to show that greedy produces the optimum for the instance under test).

The claim of Theorem 1 could be seen as being too unwieldy to be useful. In the following, we will show that (i) the theorem can be directly turned into an efficient algorithm, and that (ii) the approximation ratio that Corollary 1 guarantees for this algorithm is, on our instances, *very* close to 1.

A certifying greedy algorithm.

Theorem 1 can be turned into an algorithm that is able to certify that the solution currently produced by the greedy algorithm satisfies the theorem and it is optimal, or more generally to provide a bound on its approximation ratio. This extra computation has a moderate extra cost in time, and only requires a $O(T \log T)$ preprocessing phase (and linear extra space), where T is the size of the instance.

Given a pair $(\Omega, \mathcal{S})$, for every item $x \in \Omega$ define ξ_x to be 1 plus the cardinality of the 2nd largest set $S_i \in \mathcal{S}$ containing x; of course, for every t, x is t-certifying if and only if $\xi_x \leq t$. Now, let us store, for every $S_i \in \mathcal{S}$, an array $x_{S_i}[-]$ of $|S_i|$ entries, where $x_{S_i}[1] \leq \cdots \leq x_{S_i}[|S_i|]$ contain the values ξ_x (for all $x \in S_i$) in non-decreasing order.

By the observation above, for every t, S_i contains at least t items that are t-certifying iff $x_{S_i}[t] \leq t$. More generally, for every t, let u be the largest index such that $x_{S_i}[u] \leq t$ and let $\ell = \max(t-u, 0)$. Then, ℓ is the smallest non-negative integer such that S_i contains at least $t-\ell$ items that are t-certifying (because $t-\ell = t-t+u = u$). Index u can be found by binary search.

Armed with these observations, we can present Algorithm 1: it is a variant of the standard greedy algorithm, but at every step it provides a lower bound to the ratio between the greedy solution and the optimal one. The time required to produce the lower bound is $O(K \log n)$, where K is the size of the current solution. Since $K \leq m$, the cost per iteration is asymptotically $\log n$ larger than $O(m)$, the time required by the standard iteration of the greedy algorithm.[5] In particular, as long as $K \leq m/\log n$, Algorithm 1 is (asymptotically) not worse than the standard greedy algorithm.

Algorithm 1 Certifying the greedy algorithm with the lower bound produced by Corollary 1. Here, c represents the number of items covered by the current solution $S_1^*, \ldots, S_K^*$, γ is the gain of the last set, and L is the sum $\sum_{j=1}^{K-1} \ell_j$ using the notation of Theorem 1.

```
Input: a pair $(\Omega, \mathcal{S} = \{S_1, \ldots, S_m\})$
  $K \leftarrow 1; c \leftarrow 0$
  while true do
    choose $i \in \{1, \ldots, m\}$ maximizing $\gamma = |S_i \setminus \cup_{j=1}^{K-1} S_j^*|$
    // check that all sets contain $\geq \gamma$ items that are $\gamma$-certifying
    $L \leftarrow 0$
    for $j$ from 1 to $K-1$ do
      find the largest $u$ such that $x_{S_j^*}[u] \leq \gamma$
      $L \leftarrow L + \max(0, \gamma - u)$
    $S_K^* \leftarrow S_i$
    $c \leftarrow c + \gamma$
    Solution $S_1^*, \ldots, S_K^*$ is not worse than a $\frac{1}{1+\frac{L}{c}}$ approximation
    $K \leftarrow K + 1$
```

6. CARDINALITY AND DEGREE-BOUND RANDOM INSTANCES

In this section we introduce a natural model to produce random instances, and we show that the greedy algorithm is going to be close to optimal on them, and that our certifying greedy algorithm will be able to certify this near-optimality.

Definition 1 *Let $1 \leq q_1 \leq \ldots \leq q_n \leq m$ and $d_1 \leq \ldots \leq d_m \leq n$ be two non-decreasing sequences such that*

$$\sum_{i=1}^{n} q_i = \sum_{j=1}^{m} d_j = M,$$

and such that $q_n \cdot d_m \leq M$. We build a bipartite graph with $n+m$ nodes, and for each $1 \leq i \leq n$ and $1 \leq j \leq m$, we add an edge between the nodes corresponding to q_i and d_j independently with probability $\frac{q_i \cdot d_j}{M}$. We denote this random bipartite graph by $B(\boldsymbol{q}, \boldsymbol{d})$ (here q_i correspond to queries and d_j to documents).

We interpret this bipartite graph $B(\boldsymbol{q}, \boldsymbol{d})$ as a set system: an item q will be part of the set S iff q has an edge to S. We observe that the expected degree[6] of the node corresponding to q_i (resp., d_j) is in fact q_i (resp., d_j).

We will consider power law distributed $\boldsymbol{q}$ and $\boldsymbol{d}$, so to match what we see in our datasets. We assume that $\boldsymbol{q}$ follows a power law with exponent $\alpha > 2$, and that $\boldsymbol{d}$ follows a power law with exponent $\beta > 2$. That is, we assume that the number of items with a value q_i equal to q will be $\Theta\left(M \cdot q^{-\alpha}\right)$, and the number of sets with a value d_j equal to d will be $\Theta\left(M \cdot d^{-\beta}\right)$. A simple calculation shows that $\alpha, \beta > 2$ imply that the condition $q_n \cdot d_m \leq M$ is satisfied.

We also point out that the power law distributions imply that the tails of $\boldsymbol{q}$ and $\boldsymbol{d}$ satisfy, for every integer $t \geq 1$:

$$\sum_{\substack{i \\ q_i \geq t}} q_i \geq \Theta\left(M \cdot t^{2-\alpha}\right), \quad \text{and} \quad \sum_{\substack{j \\ d_j \geq t}} d_j \geq \Theta\left(M \cdot t^{2-\beta}\right).$$

Lemma 2 *Let Π be the instance corresponding to a sample of $B(\boldsymbol{q}, \boldsymbol{d})$, with $\boldsymbol{d}$ and $\boldsymbol{d}$ following, respectively, a power law with exponent $\alpha > 2$, and one with exponent $\beta > 2$.*

If Algorithm 1 is run on Π then, with high probability, for each K for which S_K^* produces a gain of at least $\frac{20 \ln n}{\varepsilon^2}$ new elements, then the algorithm returns, and certifies, a $(1 - O(\varepsilon))$-approximation for MAXCOVER.

PROOF. Let X_j be the number of elements having $q_i \leq Q = \varepsilon^{-\frac{1}{\alpha-2}}$ that end up inside S_j, and in no other set $S_{j'}$ such that $d_{j'} \geq L = \varepsilon^{-\frac{\alpha-1}{(\alpha-2)(\beta-2)}}$. Observe that X_j lower bounds the number of $(L+1)$-certifying elements of S_j. Then, a simple calculation shows that $E[X_j] \geq (1 - O(\varepsilon)) \cdot d_j$.

Let $G = \left\{j \mid d_j \geq \frac{10 \ln n}{\varepsilon^2}\right\}$. Since edges are inserted independently in the bipartite graph, the Chernoff bound guarantees that, with probability $1 - o(1)$, for each $j \in G$ it will hold that $X_j \geq (1 - O(\varepsilon)) \cdot d_j$.

Moreover, the Chernoff bound also guarantees that, with probability $1 - o(1)$, for each $j \in G$, it will hold that $|S_j| \leq (1 + O(\varepsilon)) \cdot d_j$. Analogously, with probability $1 - o(1)$, for each $j \in [m] - G$, we will have that $|S_j| \leq \frac{20 \ln n}{\varepsilon^2}$.

Now, let $G' = \left\{j \mid d_j \geq \frac{20 \ln n}{\varepsilon^2}\right\}$. We know that, for each $j \in G'$, it holds that $|S_j| = (1 \pm O(\varepsilon)) \cdot d_j$ and the number of $(L+1)$-certifying elements in S_j is at least $(1 - O(\varepsilon)) \cdot d_j$.

Now consider the generic iteration K of Algorithm 1 (or, of the classical greedy algorithm) . Suppose that it brings in the set S_k^*. Suppose furthermore, that it produces a gain γ satisfying $\gamma \geq \frac{20 \ln n}{\varepsilon^2}$. Then, since the gains are decreasing, we will have that $|S_1^*|, |S_2^*|, \ldots, |S_k^*| \geq \gamma$, and therefore each of them contains at least $(1 - O(\varepsilon)) \cdot \gamma$ many $(L+1)$-certifying elements. Since $L < \gamma$, this implies that we can choose $\ell_1, \ldots, \ell_{K-1} \leq \varepsilon\gamma$ in Theorem 1.

The decreasing property of the gains guarantees that the covering produced by greedy has cardinality at least $\gamma \cdot K$. Theorem 1 guarantees that the dual has value at most equal to the covering produced by greedy plus $(K-1) \cdot \varepsilon\gamma$. It follows that Algorithm 1 can return, and certify, a $(1 - O(\varepsilon))$-approximation. □

The interpretation of this result is essentially the following: under the conditions stated in the Lemma, the greedy algorithm will provide a very good approximation as long as sufficiently many items are brought in at every step — that is, as long as $\gamma = \Omega(\ln n)$. One of the key properties used in the proof is that, under the conditions in the lemma, every "large" set S contains *many items* that, excluding S, are only part of "small" sets — these elements will then be γ-certifying.

A fair point to make is that our instances are not "random". Nonetheless, we think that this result highlights some simple properties (which are sufficient for Theorem 1 to certify a good approx-

[5] If only a test for optimality is needed, it can be done in time $O(K)$, by just checking at every step that $x_{S_i}[\gamma] \leq \gamma$ for every $i = 1, \ldots, K-1$, avoiding the binary search.

[6] This random construction does not guarantee that the degree sequences are exactly given by the q_i's and d_j's: this is true only in expectation. However, there is a trivial (albeit cumbersome) proof that the degree distributions will follow the original power laws.

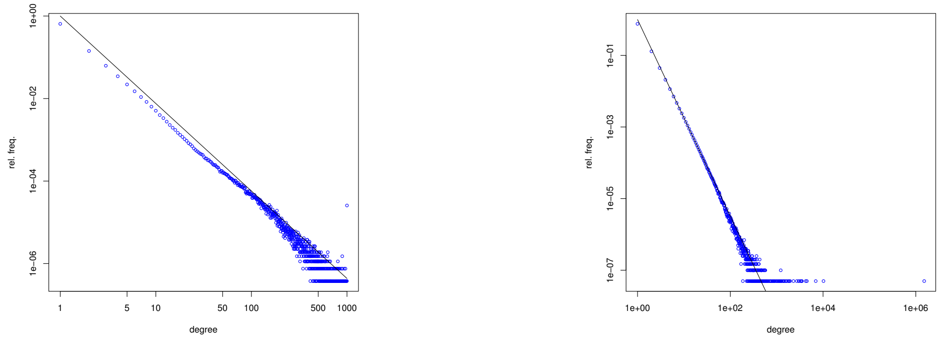

Figure 1: Plot in loglog-scale of the q-distribution (left) and d-distribution (right) of one of our largest datasets, and the corresponding power-law curve (with the exponent obtained by `plfit`). Note that in this example $k = 1\,000$, which caps the q-degrees (because a query cannot appear in more than $1\,000$ web pages).

imation) that are shared by real instances and random ones. This result is intended as a partial explanation of the surprisingly high quality of the greedy approximation.

7. EXPERIMENTAL RESULTS

Description of the datasets.

The bulk of our experiments involved four samples taken from the query logs of the Yahoo! search engine, covering different time periods, from few hours to many months. For each of the four query logs, we considered all (query,URL) pairs that corresponded to a click (the so-called click graph [12]). In order to reduce at the same time the size of the dataset and the level of noise, we performed the following cleaning operations: (i) we did not consider queries submitted less than c times, with $c \in \{5, 10, 20\}$; (ii) we did not consider (query, URL) pairs appearing less than f times, with $f \in \{5, 10, 20\}$; (iii) of the remaining pairs, for each query we only considered the (at most) top k URLs that were clicked most frequently for that query, with $k \in \{10, 50, 100, 1000\}$.

As a result, we worked on 144 click graphs, with a number of queries ranging from $10,935$ to $8,730,941$, and a number of URLs ranging from $15,393$ to $19,990,574$. The graphs were produced using Hadoop (from the distributed HFS containing the query logs) and were then compressed using WebGraph [6]: the largest of them (the one relative to a whole semester, with $c = f = 5$ and $k = 1\,000$) required 814 MB of storage.

To check the general applicability of our result we have also run our algorithm on a Twitter based instance (which we obtained from `http://an.kaist.ac.kr/traces/WWW2010.html`.) This Twitter social graph was translated into a MaxCover instance by creating one element, and one set, for each user account. The set S_u of user u contained u and all the users followed by u. Since the graph consists of more than 41M nodes and 2.5B edges, we obtained more than 41M elements and more than 41M sets. The sum of the cardinalities of these sets exceeded 2.5B.

Degree distributions.

For each of the click-graphs dataset, we computed the q- and d-distributions; recall that the d-distribution is the distribution on the sizes of the sets (*i.e.*, of the number of queries related to a specific clicked URL), whereas the q-distribution is the distribution of the number of sets in which an item appears (*i.e.*, of the number of URLs that were clicked for a specific query): because of the way data were filtered, the q-distribution is capped by k (no query will ever appear in more than k sets), whereas there is no upper bound on the largest possible value appearing in the d-distribution.

	min	1st q.	median	mean	3rd q.	max
q-degrees	1.991	2.090	2.124	2.141	2.180	2.398
d-degrees	2.571	2.837	2.892	2.937	2.982	3.258

Table 1: Statistics of the exponents of the power-law distributions for the graphs corresponding to our 144 datasets.

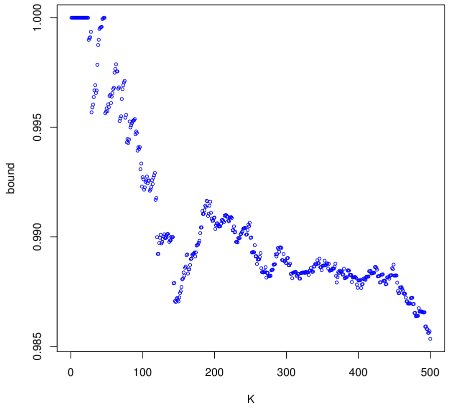

Figure 2: Bound output by Algorithm 1 for $K = 1, \dots, 500$ on the largest of our click-graph datasets.

For every degree distribution, we used the techniques of [10] (as implemented in the `plfit` tool) to fit them to a power-law and to estimate the corresponding exponent. An example is given in Figure 1 and the statistics for all the exponents of the two distributions are reported in Table 1.

Running algorithm 1.

We ran the first 500 iterations of the certifying greedy algorithm (with $K = 1, \dots, 500$) on each dataset, keeping track of the solution found and of the bound on its optimality, as produced by the algorithm. The running time was about 29 ms per iteration on an Intel Xeon CPU X5660 at 2.80GHz: this datum is averaged across 1000 iterations and includes the pre-processing time.

The overall behavior observed is the same across all the click-graph datasets, and it is drawn in Figure 2 for our largest dataset; the bound with $K = 500$ is still 0.9853, which means $< 2\%$ with respect to the optimum.

Independently on the choice of the parameters (c, f and k), the bound on the error for the largest dataset was never larger than 2%

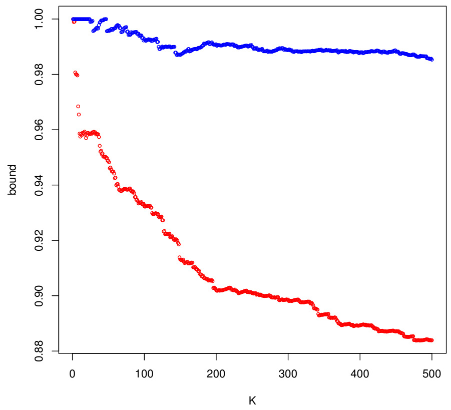

Figure 3: In blue, the bound output by Algorithm 1 for $K = 1, \ldots, 500$ on the largest of our click-graph datasets; in red, the trivial bound.

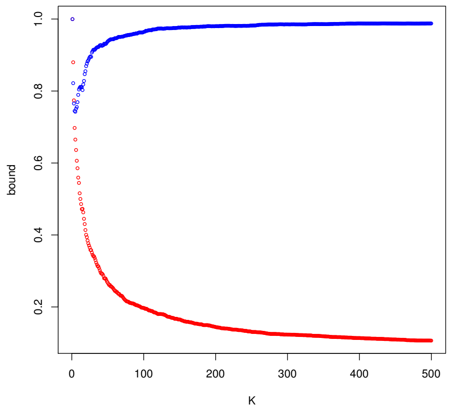

Figure 4: In blue, the bound output by Algorithm 1 on our Twitter dataset; in red, the trivial bound.

for all $K = 1, \ldots, 500$. It is generally observed that increasing c and f or decreasing k leads to larger error bounds, simply because it makes the sets smaller: this is true across all datasets.

We compared our bound on the approximation ratio with a baseline (the ratio between the number of items covered by the greedy solution and the sum of the cardinalities of the sets selected). The baseline bound on the largest dataset is much weaker than our bound (the baseline gives ratios between 0.883 and 0.933): the comparison is shown in Figure 3.

It is worth noting that the impact of c introduces a different bias on the queries considered depending on the size of the time slot: on a one-hour log, disregarding queries appearing less than, say, 20 times means focussing on “hot topics”, whereas the same value of c would produce many tail queries on a six-month log. The fact that there is no substantial difference in the results for the same value of c across different datasets means that our technique is not sensible to this bias. Moreover, we expect that considering even very infrequent queries (e.g., $c = 1$) might only further reduce the error bounds.

Interestingly our algorithm has a different behavior on the Twitter-based dataset (Figure 4): the guaranteed approximation ratio increases with K (and it converges quickly to $\approx 99\%$) — our click-graph datasets, instead, have a guaranteed approximation ratio that decreases with K. In both cases, though, our algorithm guarantees approximation ratios very close to 1. It would be extremely interesting to see if our algorithm still guarantees very good approximation ratios on other large-scale web, social or networking datasets.

The tightness of our bound.

The bound produced by the algorithm seems to witness that the greedy algorithm behaves much better than expected, especially on large datasets, which is the case for web search; in fact, the real approximation ratio may be even *better* than that!

We have tested this hypothesis by explicitly solving the linear program (using the IBM `cplex` optimizer[7]) on (each of) the smallest datasets for $K = 1, \ldots, 500$. We could not run it on larger datasets for performance-related reasons (solving the LP would have taken too much time, and too much memory).

We were surprised to observe that, in each of these test cases, the LP value was *at most 8 additive units more than* the value of the solution produced by greedy; moreover, in each test case, the approximation ratio was at least 99.79%. In many cases the two values did coincide, showing that greedy was in fact producing the best solution. We find this observation extremely appealing. We are still lacking a completely satisfying explanation of this phenomenon — we see Theorem 1 and Section 6 as just a first step in this direction.

Rewiring the datasets.

Since we are interested in the way the degree distributions influence the behavior of our algorithm, for each of the 144 datasets we produced two rewired variants: (i) *q-scrambling* (the web page cardinalities were kept unchanged, but the queries contained in each web page were chosen u.a.r.); (ii) *d-scrambling* (the numbers of web pages where each query appears were kept unchanged, but the actual web pages to which each query belongs were chosen u.a.r.). In other words, the two variants are designed so that one of the two degree distributions matches the one summarized in Table 1 whereas the other becomes uniform.

We then ran Algorithm 1 on the scrambled datasets; the outcome was more or less the same for all cases: (i) On the q-scrambled datasets (same cardinalities but items chosen u.a.r.), the bound produced by the algorithm is extremely good for all $K = 1, \ldots, 500$: essentially the greedy algorithm produces the *optimal solution* in all cases. Intuitively, the q-scrambled dataset resembles the *best possible world*: the largest sets are disjoint with high probability, and choosing the K largest ones provides a solution that is very close to the optimum. (ii) On the d-scrambled datasets (items have the same degrees but are assigned to sets u.a.r.), greedy performs much worse than in the original datasets. An intuitive reason is that sets are much smaller than in the original case, and therefore it is much harder for Algorithm 1 to find certifying elements.

Ratio of coverage.

One important aspect that we have not yet discussed is how many queries of the universe we are, in fact, covering. We observe that the number of queries covered at $K = 500$ is negligible (only 2.5% in our largest dataset). In Figure 5, we present Algorithm 1’s results on our largest dataset for K up to $140,000$.

Recall that Algorithm 1 serves two complementary needs. First, it tries to select K web pages to maximize the number of queries covered. Second, it gives an upper bound on the maximum number of queries that can be covered with K web pages. Figure 5 shows the algorithm performance in these two tasks: the blue line repre-

[7] `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`

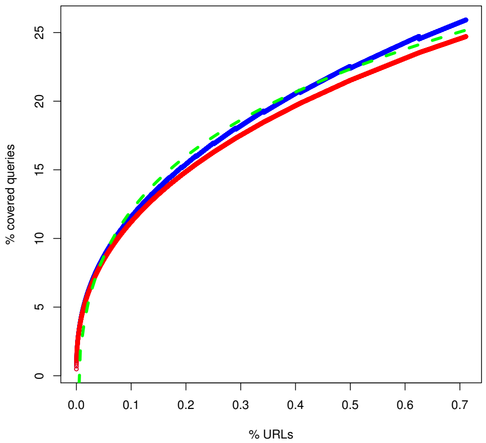

Figure 5: The fraction of queries (red) covered by Algorithm 1 with respect to the fraction of web pages on the largest of our datasets; for comparison, we show the upper bound on the fraction (blue) produced by Theorem 1, as well as the polynomial model (dashed).

sents the fraction of queries that Algorithm 1 could cover with a given K, while the red line represents the aforementioned upper bound.

For concreteness we mention that, for our largest dataset, 24.5% of the query universe can be covered with just less of 140,000 URLs (0.7% of the URLs!), and that it is impossible to cover the same percentage with fewer than $123,000$ URLs. We also observe that the MaxCover approximation that Algorithm 1 can certify is quite good even at $K \approx 140,000$. At such a large K, the algorithm certifies an approximation of 0.95; the trivial sum-bound certificate at the same K would not be better than 0.74. In fact, up to $K \approx 360,000$ (1.8%) we can ensure that the solution is almost optimal.

If we fit a power law to the increments of unique queries covered by Algorithm 1 in Figure 5, we obtain an exponent of -0.76. This is not surprising, because usually the interaction of power laws gives another power law. In fact, this exponent is very close to the difference in the power law exponents of the q- and d-sequences. Hence, we fitted a model of the form $\alpha \cdot x^{0.24} - \beta$ to the coverage curve obtaining an error of just $5.7 \cdot 10^{-3}$ for $\alpha = 1.148$ and $\beta = 0.098$ (see dashed line in Figure 5). Notice that for $x = 1$ we obtain a coverage of 1.05! (a perfect model should have coverage 1). From this model we can estimate a coverage of 50% of the queries with just 6.6% of the URLs and a coverage of 87% when using 50% of the web pages (these estimations are pessimistic as the model grows slower than the real curve). This is quite remarkable considering that typically half of distinct queries are singletons (that is, they appear only once in the query log) and their overall volume is significant (say 25%).

8. CONCLUSIONS AND FUTURE WORK

We have shown that we can find an almost optimal small set of web pages that can cover a large subset of all possible unique queries — moreover, our algorithmic guarantee seems to hold even for other large-scale (non web page-related) data settings. We plan to extend our results to the weighted version (that is, to maximize the coverage of the query volume), and to other kinds of datasets. We believe that maximizing this case might be a harder task than our original one. However, the coverage results will be better. We expect the coverage to be at least 50% of the query volume with 5% of the web pages, because most important URLs are good answers for a large fraction of the query volume (*e.g.*, see [3]). We also plan to change the definition of relevance (*e.g.*, use all top answers and not only the clicked pages or weight the clicked pages by the dwell time to control for novelty and spam) and of coverage (*e.g.*, a query is covered only if the index contains at least t relevant pages).

Our results partly depend on a double power-law, one for queries and another for web pages. The power law of web pages do depend on the ranking function of the search engine. However, apart from simple ranking schemes such as PageRank that can be shown to be a power law, there is little on the study of score distributions for ranking functions (*e.g.*, see [18]). Nevertheless, in practice ranking score distributions should follow a power law, as it happens in typical features used for ranking functions, such as terms or clicks frequencies. Although in practice web indices contain hundreds of billions of pages, our findings show that our approach is invariant to scale (the results are similar from thousands to tens of millions), as expected given the power-law distributions involved. Hence, we believe that our results can be extrapolated to hundreds and billions of web pages if we have the right power law exponents of the distributions involved.

Another issue is that the query distribution of web search engines is not static. However, the changes in time are small. In fact, in [3] they find that the pairwise correlation among all possible 3-week periods of the query distribution for 15-weeks was always over 99.5%. This implies that daily or weekly updates to the set of essential web pages should be enough. These updates are not only necessary due to changes in the query stream, but also because in practice there will be changes in the set of web pages available for indexing. Hence, our approach can be another way to define periodical updates to the index of a web search engine.

Regarding the secondary index problem, we can design a crawling algorithm driven by the query distribution, as mentioned in [2]. This approach would gather relevant pages for the secondary index and should also improve the pool of pages available for selecting the essential web pages for the main index.

The problem we have solved not only can be used to reduce the size of the main index of a web search engine. Indeed, if we can predict the query intention, we could use the click knowledge of the overall web search engine to build optimized indices for vertical search engines tailored to a particular intention. The same idea applies to other query subsets such as queries coming from a given country or language. In these vertical search engines we can have tailored ranking functions as well as tailored user experiences.

Other problems where our results can be used include:

- Optimize the selection of the cache of web pages in web search engines, if there are limited memory resources.

- Optimizing the indexed web pages in each local server of a distributed web search architecture given finite memory resources [4, Section 10.6.2].

- Optimizing document caching in a distributed web search scheme [4, Section 11.4.5] where each local server caches a small set of documents (in principle, just the most frequently accessed) to reduce the query search time.

- Optimizing the indexed documents in any P2P distributed search scheme, given the local resources defined by every peer [4, Section 10.8].

9. REFERENCES

[1] A. Anagnostopoulos, L. Becchetti, S. Leonardi, I. Mele, and P. Sankowski. Stochastic Query Covering. In *Proc. of WSDM 2011*, pages 725–734, 2011. ACM.

[2] R. Baeza-Yates. Information retrieval in the web: beyond current search engines. *Int. J. Approx. Reasoning*, 34(2-3):97–104, 2003.

[3] R. Baeza-Yates, A. Gionis, F. Junqueira, V. Murdock, V. Plachouras, and F. Silvestri. Design trade-offs for search engine caching. *TWEB*, 2(4), 2008.

[4] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison-Wesley, Harlow, UK, second edition, 2011.

[5] R. Baeza-Yates, V. Murdock, and C. Hauff. Efficiency trade-offs in two-tier web search systems. In *Proc. of SIGIR 2009*, pages 163–170, 2009.

[6] P. Boldi and S. Vigna. The WebGraph framework I: Compression techniques. In *Proc. of WWW 2004*, pages 595–601, Manhattan, USA, 2004. ACM Press.

[7] D. Carmel, D. Cohen, R. Fagin, E. Farchi, M. Herscovici, Y. S. Maarek, and A. Soffer. Static index pruning for information retrieval systems. In *Proc. of SIGIR 2001*, pages 43–50, 2001.

[8] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proc. of KDD 2009*, pages 199–208, New York, NY, USA, 2009. ACM.

[9] F. Chierichetti, R. Kumar, and A. Tomkins. Max-cover in map-reduce. In *Proc. of WWW 2010*, pages 231–240, New York, NY, USA, 2010. ACM.

[10] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Rev.*, 51(4):661–703, Nov. 2009.

[11] R. Cohen and L. Katzir. The generalized maximum coverage problem. *Inf. Process. Lett.*, 108(1):15–22, 2008.

[12] N. Craswell and M. Szummer. Random walks on the click graph. In *Proc. of SIGIR 2007*, pages 239–246, New York, NY, USA, 2007. ACM.

[13] A. Dasgupta, A. Ghosh, R. Kumar, C. Olston, S. Pandey, and A. Tomkins. The discoverability of the web. In *Proc. of WWW 2007*, pages 421–430, New York, NY, USA, 2007. ACM.

[14] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[15] D. S. Hochbaum. Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. In D. S. Hochbaum, editor, *Approximation algorithms for NP-hard problems*, pages 94–143. PWS Publishing Co., Boston, MA, USA, 1997.

[16] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proc. of KDD 2003*, pages 137–146, New York, NY, USA, 2003. ACM.

[17] K.M. Risvik, Y. Aasheim, and M. Lidal. Multi-Tier Architecture for Web Search Engines. In *Proc. of LA-WEB '03*, pages 132–, 2003. IEEE.

[18] S. Robertson. On score distributions and relevance. In *29th European Conference on IR Research*, LNCS, pages 40–51, Rome, Italy, April 2007. Springer.

[19] G. Skobeltsyn, F. Junqueira, V. Plachouras, and R. Baeza-Yates. Resin: a combination of results caching and index pruning for high-performance web search engines. In *SIGIR*, pages 131–138, 2008.

[20] A. Swaminathan, C. V. Mathew, and D. Kirovski. Essential pages. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '09, pages 173–182, Washington, DC, USA, 2009. IEEE Computer Society.

[21] V. V. Vazirani. *Approximation algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.