Estimating Collection Size in Distributed Search

Jingfang Xu Sheng Wu Xing Li

xjf02@mails.tsinghua.edu.cn wu-s05@mails.tsinghua.edu.cn xing@cernet.edu.cn
Department of Electronic Engineering, Tsinghua Unviversity, Beijing, 100084, China

Abstract

Distributed search is an effective way to search information over thousands of information collections available on the web. As an important feature in distributed search, collection size plays a vital role in resource representation and selection. This paper proposes two novel algorithms to estimate collection size in uncooperative environments. *Sample high frequent resample* (SHFRS) algorithm firstly samples collections with random queries and then resamples with highest frequent queries in sample sets. Considering different capture probabilities across documents, *heterogeneous capture* (HC) algorithm estimates collection size with conditional maximum likelihood. Both algorithms are evaluated on real web data. Experimental results show that our algorithms outperform significantly both sample-resample and capture-recapture algorithms.

Keywords: Multiple capture-recapture, Sample-resample, Collection size estimation

1. Introduction

In the World-Wide Web today, there is a wealth of information "hidden" in the information collections such as digital libraries and web databases which are called *deep web*. Unlike information in the surface web, rather than crawled by general search engines, e.g. Google, information in these collections can only be accessed through collections' search interfaces. One way to integrate information in these collections automatically is through distributed search engines, which search multiple collections in distributed environments. Given a query, a distributed search engine ranks the underlying collections and selects some of them to forward the query (resource selection) based on their content summaries (resource representation), and then merges the results returned from the selected collections (results merging).

As an important feature in resource representation, collection size, the number of documents in a collection, is considered in many resource selection algorithms [Si and Callan (2003)][Liu et al. (2002)]. In cooperative environments, collections follow some special protocols, e.g. STARTS [Gravano et al. (1997)], providing their content sum-maries initiatively, including their sizes. However, in practice, collections are

control-led by different organizations, so acquiring resource representation with protocols will fail when collection controllers do not cooperate or misrepresent their contents. Consequently, it is important for distributed search engine to estimate the content summaries of underlying collections in uncooperative environments. Some collection size estimation algorithms in uncooperative environments have been proposed, such as *sample-resample* [Si and Callan (2003)] and *capture-recapture* [Shokouhi et al (2006)][Liu et al. (2002)], but their estimation accuracy need to be improved.

This paper presents two novel collection size estimation algorithms in uncooperative environments: *sample high frequent resample* (SHFRS) and *heterogeneous capture* (HC). SHFRS firstly collects sample documents by running some random queries in a collection, and then resamples with high-frequent queries in the sample set. On the other hand, allowing different capture probabilities, HC estimates collection size with conditional maximum likelihood. In addition, the capture probability is formulated with linear logistic regression. Experimental results show that our algorithms outperform previous algorithms, which indicates that our algorithms are more suitable to estimate collection size.

The rest of the paper is organized as follows. First previous work is briefly reviewed in section 2. Then we present SHFRS algorithm in section 3, and HC algorithm in section 4. Section 5 describes the experiments and results. Finally we conclude with future work in section 6.

2. Previous Work

In cooperative environments, it is specified in some protocols, e.g. STARTS, that collection size is one important element for describing collections, which need to be exported by collections initiatively. Correspondingly, in uncooperative environments, *sample-resample* and *capture-recapture* are proposed for estimating collection size.

Sample-resample, proposed by Callan et al., consists of two steps: sample and resample. First, the algorithm collects a sample set by running some random queries in the collection's search interface. Second, some random terms in the sample set are sent to the search interface to investigate their document frequencies in the complete collection. Assuming that the probability of a random term occurring in the sample set equals to that in complete collection, the collection size can be estimated.

Capture-recapture is firstly proposed by Liu et al., and then refined by Shokouhi et al. to *multiple capture-recapture*. In *multiple capture-recapture*, the returned documents of each query are viewed as a sample set, and T sample sets with the same size k are needed. Assuming random capture, the expected number of duplicate documents in two random samples is k^2/N, where N is the collection size. Using T samples, the total number of pairwise duplicate documents D can be formulated as

$T(T+1)k^2/2N$. Hence, the collection size is estimated as $T(T+1)k^2/2D$. The basic assumption in capture-recapture methods is random capture, which means that capture probabilities across documents in the collection are the same. Unfortunately, in practice, capture probabi-lities are usually biased towards long documents or documents with high static ranks. Moreover, capture probabilities are also related with ranking functions used in the co-llections [Shokouhi et al (2006)][Bar-Yossef and Gurevich (2006)]. To eliminate the influence of the bias, Border et al. [Border et al.(2006)] and Bar-Yossef et al. [Bar-Yossef and Gurevich (2007)] introduce *importance sampling*, which is to estimate collection size by uniform distribution using observations from a biased distribution.

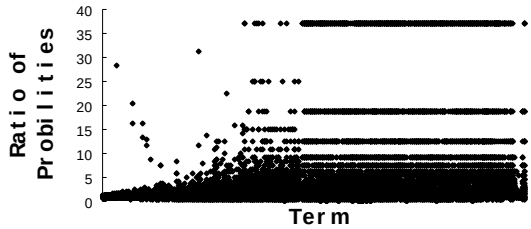

Figure 1. ***Ratio of terms' occurring probabilities in sample set and in collection***

3. Sample High Frequent Resample Algorithm

In sample-resample method, the probability $P_{sample}(t)$ of term t occurring in the sample set and $P_{collection}(t)$ of term t occurring in the collection are formulated as Equation 1.

$$P_{sample}(t) = \frac{df_{sample}(t)}{N_{sample}} \quad P_{collection}(t) = \frac{df(t)}{N} \tag{1}$$

, where $df_{sample}(t)$ and $df(t)$ are document frequencies of term t in the sample set and in the complete collection, N_{sample} and N are referred to as size of the sample set and the collection. It is assumed that the two probabilities are almost the same. However, we did some experiments to compare the two probabilities across terms and found that is not exact. We built a site search for a real web site, and collected a sample set with *sample-resample* method. Figure 1 shows the ratios of term occurrence probabilities in two sets. The horizontal axis represents terms, sorted according to their document frequencies in the collection, while the vertical axis shows $P_{sample}(t)/P_{collection}(t)$. As illustrated in the figure, for most of the terms, the ratio values are far different from 1, which means the two probabilities are vastly different. However, on the most left of the figure, referred to as the highest frequent words, the ratio values are close to 1. In other words, the approximate occurrence probabilities assumption in s*ample-resample* method is only correct for highest frequent terms. Therefore, most of the words are unsuitable for collection size estimation except highest frequent terms. Maybe it is

because that highest frequent terms are probably stop-words or kernel words of collection's topic, which are used in most of the documents in the collection.

According to this observation, SHFRS is proposed. Different from *sample-resample* algorithm, which selects random terms to resample, only highest frequent terms are used in SHFRS, which is described as follows in detail.

1. Run some random queries to search the collection, and collect top k documents returned for each query to build a sample set with predefined size N_{sample}.

2. Calculate document frequencies of terms occurring in the sample set, and use n highest terms to resample. The numbers of matches reported by the search interface are referred to as their document frequencies in the collection.

3. With document frequencies of terms resampled, the collection size is estimated with Equation 1, 2.

$$P_{sample}(t) = P_{collection}(t) \qquad then \quad \hat{N} = \frac{1}{n}\sum_{i=1}^{n}\frac{df(t_i) * N_{sample}}{df_{sample}(t_i)} \tag{2}$$

4. Heterogeneous Capture Algorithm

Both *sample-resample* and SHFRS need the numbers of matches reported by search interfaces. So they will fail when search interfaces do not report the match numbers or lie. Though the capture-recapture mechanisms do not require the number of matches, their assumption about random capture is questionable as mentioned in Section 2.

Different from the effort of eliminating capture bias in Border et al. and Bar-Yossef et al.'s work, considering different capture probabilities of documents, HC algorithm is inspired by a kind of *capture-recapture* methods, which allows heterogeneous capture probabilities, used in ecology first [Alho (1990)][Huggins (1989)]. In HC, the capture probability of a document is related with its characteristics, such as length and static rank, as well as the capture environments, such as sample query. Using covariates x_{ij}, the capture probability is estimated with linear logistic model, shown as Equation 3.

$$P_{ij} = \frac{\exp(\beta' x_{ij})}{1+\exp(\beta' x_{ij})} = \frac{\exp(\beta_0 + \beta_1 \cdot len_i + \beta_2 \cdot rank_i + \beta_3 \cdot tf_{ij})}{1+\exp(\beta_0 + \beta_1 \cdot len_i + \beta_2 \cdot rank_i + \beta_3 \cdot tf_{ij})} \tag{3}$$

, where P_{ij} is the capture probability that document i is captured on the jth capture, len_i is the length of document i, $rank_i$ is the static rank of document i, tf_{ij} is the term frequency of the jth query in document i, and $\beta_0, \beta_1, \beta_2, \beta_3$ are linear coefficient.

Supposing a collection with N documents, we sample it k times with k random queries. On each sample, we capture some documents and record them. Then the full likelihood is formulated as follows.

$$L = K \prod_{i=1}^{N} \prod_{j=1}^{k} P_{ij}^{\delta_{ij}} (1 - P_{ij})^{1-\delta_{ij}} \quad \textbf{(4)}$$

, where K depends on none parameters other than N, $\delta_{ij}=1$ if document i is captured on the jth capture and $\delta_{ij}=0$ otherwise. Unfortunately, we can not collect the values of covariates, such as len_i and $rank_i$, for documents never captured. Suppose a total of n documents, labeled $i=1, 2, ..., n$, are captured. Let C_i be the event that document i is captured at least once and C_{ij} be the event that document i is captured on the jth capture. Given C_i, the conditional capture probability r_{ij} of document i on the jth capture is formulated as Equation 5. According to *Huggins-Alho* method [Alho (1990)] [Huggins (1989)], we infer the parameters with conditional maximum likelihood, involving only the documents captured at least once, shown as Equation 6. Then we estimate the collection size with *Horvitz-Thompson* estimator with Equation 7.

$$r_{ij} = \begin{cases} \dfrac{P_{ij}}{1 - \prod_{l=j}^{k} (1 - P_{ij})}, & if \ z_{ij} = 0 \\ P_{ij}, & otherwise \end{cases} \quad \textbf{(5)}$$

, where z_{ij} is 1 if document i has been captured before the jth capture and 0 otherwise.

$$L = \prod_{i=1}^{n} \prod_{j=1}^{k} r_{ij}^{\delta_{ij}} (1 - r_{ij})^{1-\delta_{ij}} \quad \textbf{(6)}$$

$$\hat{N} = \sum_{i=1}^{n} \frac{1}{P_i}, \ P_i = 1 - \prod_{j=1}^{k} (1 - P_{ij}) \quad \textbf{(7)}$$

, where P_i is the probability that document i being captured at least once.

5. Experimental Results

5.1 Data

The algorithms described in this paper were evaluated with real web data. We crawled 50 Chinese web sites and built a site search engine for each of them. These site search engines are viewed as distributed collections, whose size we need to estimate. In addition, the ranking function used in the site search engine is static score, calculated by PageRank algorithm, combined with dynamic score, calculated by Okapi BM25 formula. Their actual sizes vary from 793 to 342,159 documents, and the average size of them is 29,440.

5.2 Metric

Absolute error ratio (AER)[Si and Callan (2003)] is adopted to measure the accuracy of collection size estimation. Let N be the actual size of a search engine and $\hat{N}$ be the estimated value, and then AER is defined as follows.

$$AER = \frac{\left| N - \hat{N} \right|}{N} \quad \textbf{(8)}$$

5.3 Implementation

Four kinds of collection size estimation algorithms were evaluated in our experiments, *sample-resample*, SHFRS, *multiple capture-recapture*, and HC. In the *sample-resample* and SHFRS algorithms, we selected the top 10 returned documents for every query to build a sample set, until its size reached the predefined size. While the *sample-resample* method selected 10 random terms in sample set to resample, our SHFRS algorithm chose 10 highest frequent terms. If the collection does not support the retrieval of some queries selected in SHFRS, e.g. stop-words, we just skipped them and selected other high frequent terms.

In *multiple capture-recapture* and HC methods, we collected 100 documents on each capture with random terms occurring in the sample set. In addition, as the covariates in HC, the document length and term frequency of query can be acquired by parsing the content of the document, but the static rank of the document, e.g. PageRank score, is hard to obtain directly without the link graph of all the documents in the collection. So we used the average position of the document in the returned lists across all the queries to approximate its static rank. It is based on the assumption that the static ranks of the documents which usually occur in the front of the returned lists are probably higher than those of the documents occurring in the end of the lists.

5.4 Results and Discussions

For *sample-resample* and SHFRS algorithms, we evaluate their performances with sample sets of different sizes, from 300 documents to 2000 documents. As shown in Figure 2, across all the sample sets, the AER values of SHFRS vary from 31.8% to 36.8%, and the minimum is achieved with the sample set containing 1000 documents. However, the performance of *sample-resample* is poor, ranging from 49.4% to 413.9%, which is similar to the results shown by Shokouhi et al.[Shokouhi et al (2006)]. The results indicate that *sample-resample* is unsuitable for size estimation.

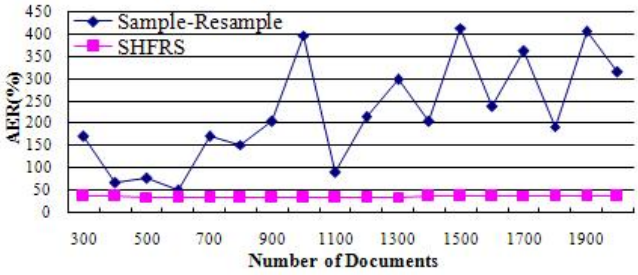

Figure 2. ***Performance of Sample-Resample and SHFRS***

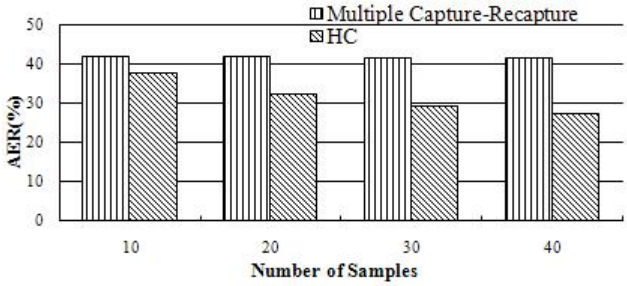

Figure 3. ***Performance of Multiple Capture-Recapture and HC***

On the other hand, the performance of HC and *multiple capture-recapture* algorithms with different number of samples is shown in Figure 3. While the AER values of *multiple capture-recapture* fluctuate around 41% with different number sample sets, the performance of HC varies from 37.4% to 27.3%, decreasing with the increase of the number of sample sets. To sum up, the performance of *multiple capture-recapture* is robust but is poorer than that of HC algorithm.

The best performance of these algorithms is shown in Table 1. HC is the best among all the algorithms, 14.5% better than SHFRS, and 33.9% better than *multiple capture-recapture*, due to considering the different capture probabilities across documents. Depending on match number reported by the collection, SHFRS has a little advantage over *multiple capture-recapture*. So SHFRS is more accurate than *multiple capture-recapture* when the collection reports the number of matches correctly.

Table 1. ***Best performance of all algorithms***

Algorithm	Sample-Resample	SHFRS	Multiple Capture-Recapture	HC
AER	49.4%	31.8%	41.3%	27.3%

6. Conclusions and Future Work

In this paper, we propose two novel collection size estimation algorithms: SHFRS and HC. After collecting a sample set with some random queries, SHFRS resamples a collection with some highest frequent terms in the sample set. Taking advantage of highest frequent terms, our SHFRS algorithm outperforms the well-known *sample-resample* algorithm. Considering the different capture probabilities of documents, HC algorithm estimates the collection size with conditional maximum likelihood based on the documents that have been captured. To the best of our knowledge, it is the first time that heterogeneous capture probabilities are allowed in the collection size estimation algorithms. All the algorithms are evaluated on the real web data. Experimental results show that our algorithms outperform significantly both t*sample-resample* and *multiple capture-recapture* algorithms. For the future work, we plan to use more features in HC algorithm to estimate the capture probability, and the effects of different features will also be explored.

References

A. Broder, M. Fontoura, V. Josifovski, R. Kumar, R. Motwani, S. Nabar, R. Panigrahy, A. Tomkins, and Y. Xu. (2006), Estimating corpus size via queries. Proc. 15th CIKM, 2006.

Alho, J. (1990), Logistic regression in capture-recapture models, Biometrics, vol. 46, pp.623-635

Bar-Yossef, Z., Gurevich, M. (2006), Random sampling from a search engine's index, In Proceedings of WWW'06, pp. 367-376.

Bar-Yossef, Z., Gurevich, M. (2007), Efficient Search Engine Measurements, In Proceedings of WWW'07.

Callan, J., Connell, M. (2001), Query-based sampling of text databases, ACM Transactions on Information Systems, vol. 19, pp. 97-130.

Gravano, L., Chang, C.-C.K., Garcia-Molina, H., Paepake, A. (1997), STARTS: Stanford proposal for internet meta-searching, In Proceedings of SIGMOD'97, pp. 207-218.

Huggins, R. (1989), On the statistical analysis of capture experiments, Biometrika, vol. 76, pp. 133-140.

Liu, K., Yu, C., Meng, W. (2002), Discovering the representative of a search engine, In Proceedings of CIKM'02, pp. 652-654.

Si, L., Jin, R., Callan, J., Ogilvie, P. (2002), A language modelling framework for resource selection and results merging, In Proceedings of CIKM'02, pp. 391-397.

Si, L., Callan, J. (2003), Relevant document distribution estimation method for resource selection, In Proceedings of SIGIR'03, pp. 298-305.

Shokouhi, M., Zobel, J., Scholer, Falk., Tahaghoghi, S. (2006), Capturing collection size for distributed non-cooperative retrieval, In Proceedings of SIGIR'06, pp. 316-323.