Estimating Collection Size with Logistic Regression

Jingfang Xu
xjf02@mails.tsinghua.edu.cn

Sheng Wu
wu-s05@mails.tsinghua.edu.cn

Xing Li
xing@cernet.edu.cn

Department of Electronic Engineering
Tsinghua University, Beijing, 100084, China

ABSTRACT

Collection size is an important feature to represent the content summaries of a collection, and plays a vital role in collection selection for distributed search. In uncooperative environments, collection size estimation algorithms are adopted to estimate the sizes of collections with their search interfaces. This paper proposes *heterogeneous capture* (HC) algorithm, in which the capture probabilities of documents are modeled with logistic regression. With heterogeneous capture probabilities, HC algorithm estimates collection size through conditional maximum likelihood. Experimental results on real web data show that our HC algorithm outperforms both *multiple capture-recapture* and *capture history* algorithms.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information search and Retrieval

General Terms: Algorithms

Keywords: Distributed Search, Collection Size Estimation

1. INTRODUCTION

Collection size, the number of documents in a collection, is an important element to represent the content of the collection, and has great effect on resource selection in distributed search. In uncooperative environments, where collections do not export their sizes initially, the broker of distributed search needs to estimate them by itself. Several algorithms based on capture-recapture mechanism have been proposed for collection size estimation, such as *capture-recapture* (CR)[4], *multiple capture-recapture* (MCR)[5], and *capture history* (CH)[5]. Generally, these algorithms estimate collection sizes with the number of duplicate documents in different samples. Homogeneous capture probabilities across documents, that every document has the same probability to be sampled, are assumed in these algorithms. However, in fact, the capture probabilities across documents are heterogeneous, being biased towards long documents or those with high static ranks, e.g., PageRank scores[2][5]. In order to allow heterogeneous capture probabilities across documents and capture times, *heterogeneous capture* (HC) algorithm is proposed in this paper. It is inspired by a capture-recapture method with heterogeneous capture probabilities used in ecology[1][3]. Using covariates, HC algorithm models the capture probabilities with logistic regression, and estimates collection size through conditional maximum likelihood. Experimental results on real web data show that our HC algorithm outperforms existing capture-recapture based algorithms.

The rest of the paper is organized as follows. First the HC algorithm is introduced in Section 2, and then Section 3 describes the experiments and results. Finally we conclude with future work in section 4.

2. HC ALGORITHM

Supposing a collection with N documents, we capture it k times by searching k random queries through its search interface. On each capture, the returned documents are captured and recorded. Here, document's capture probability relies on both its characteristics, such as its length and static rank (query-independent), and the environments of the capture, such as the query used to sample. Using covariates (len_i,$rank_i$,tf_{ij}), the capture probability is modeled with linear logistic model, shown as Equation 1.

$$P_{ij} = \frac{\exp(\beta_0 + \beta_1 \cdot len_i + \beta_2 \cdot rank_i + \beta_3 \cdot tf_{ij})}{1 + \exp(\beta_0 + \beta_1 \cdot len_i + \beta_2 \cdot rank_i + \beta_3 \cdot tf_{ij})} \qquad (1)$$

, where P_{ij} is the capture probability that document i is captured on the jth capture , len_i is the length of the 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 β_0,β_1,β_2,β_3 are unknown parameters. Then the full likelihood of k captures is formulated as follows.

$$L = \prod_{i=1}^{N} \prod_{j=1}^{k} P_{ij}^{\delta_{ij}} (1 - P_{ij})^{1-\delta_{ij}} \qquad (2)$$

, where $\delta_{ij}=1$ if document i is captured on the jth capture and $\delta_{ij}=0$ otherwise. Unfortunately, we can not collect the values of covariates for documents never captured. Suppose a total of n documents, which we label $i=1,2,...,n$, 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 3.

$$r_{ij} = P(C_{ij} \mid C_i) = \begin{cases} \dfrac{P_{ij}}{1 - \prod_{l=j}^{k} (1 - P_{il})}, & if \ z_{ij} = 0 \\ P_{ij}, & \text{otherwise} \end{cases} \qquad (3)$$

, where z_{ij} is 1 if the document i has been captured before the jth capture and 0 otherwise. According to *Huggins-Alho* method[1][3], parameters are inferred with conditional maximum likelihood, which involves only the documents that are captured at least once, shown as Equation 4. Then, given the estimated parameters, the collection size is estimated with *Horvitz-Thompson* estimator, shown as Equation 5.

SIGIR'07, July 23-27, 2007, Amsterdam, The Netherlands.
ACM 978-1-59593-597-7/07/0007.

$$L_2 = \prod_{i=1}^{n}\prod_{j=1}^{k} r_{ij}^{\delta_{ij}} (1-r_{ij})^{1-\delta_{ij}} \quad (4)$$

$$\hat{N} = \sum_{i=1}^{n} \frac{1}{P_i}, \quad P_i = 1 - \prod_{j=1}^{k} (1-P_{ij}) \quad (5)$$

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

In addition, as the covariates in HC, the length and the term frequency of query can be acquired by parsing the content of the document, while 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 use 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 documents at the top of returned lists across many queries tend to have higher static scores than those at the bottom of the lists.

3. EXPERIMENTAL RESULTS

Three kinds of collection size estimation algorithms, HC, MCR, and CH, 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 sizes vary from 793 to 342159, and the average is 29440. In addition, the ranking function used in these site search engines is Okpai BM25 formula combined with PageRank algorithm.

Mean Absolute error ratio (MAER)[6] is adopted to measure the accuracy of collection size estimation. Let N_i be the actual size of collection i and $\hat{N}_i$ be the estimated value, and then MAER of n collections is defined as follows.

$$MAER = \frac{1}{n}\sum_{i=1}^{n} \frac{|N_i - \hat{N}_i|}{N_i} \quad (6)$$

Searching queries in these site search engines, we captured and recorded the returned documents. We collected 100 documents on each capture and selected queries randomly from the captured documents. In total, 100 captures were used in our experiments.

The performance of three algorithms with different numbers of captures is shown in Figure 1. While the MAER values of MCR with different numbers of captures fluctuate around 41%, the performance of HC and CH improves with the increase of captures. From 10 to 100 captures, the MAER value of HC varies from 37.4% to 20.0%, while that of CH varies from 38.2% to 23.9%. To sum up, HC is the best among these algorithms, and the success is due to consideration of heterogeneous capture probabilities across documents and capture times.

To explore the importance of covariates in HC algorithm, Table 1 shows the mean weights of covariates in Equation 1 across all collections with 100 captures. The values of all covariates are normalized between 0 and 1. As we can see, the weights of length, static rank and term frequency are 0.61, 4.46 and 14.24, respectively. It indicates that term frequency is the most important feature for capture probability, and static rank is more important than length.

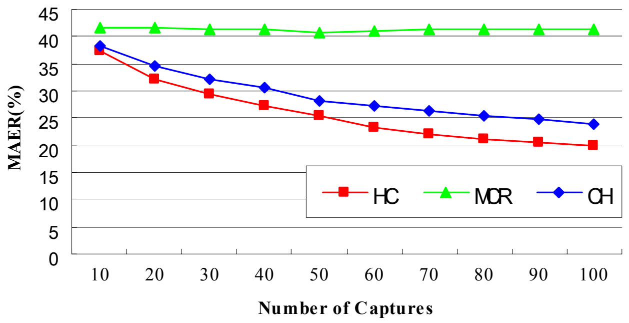

Figure 1: Performance of collection size estimation algorithms

Table 1: Weights of Covariates

constant(β₀)	length (β₁)	rank (β₂)	term frequency (β₃)
-5.40	0.61	4.46	14.24

4. CONCLUSIONS

In this paper, we propose a novel collection size estimation algorithm, HC, for distributed search. Considering heterogeneous capture probabilities, HC algorithm models capture probabilities across documents with logistic regression, and estimates collection size through conditional maximum likelihood. To the best of our knowledge, it is the first time that heterogeneous capture probabilities are applied to collection size estimation. Experimental results on real web data show that HC algorithm outperforms both MCR and CH algorithms. For the future work, we plan to use more features in HC algorithm to estimate the capture probability, and explore the effects of different features.

5. REFERENCES

[1] J. Alho. Logistic regression in capture-recapture models. Biometrics, 1990, vol. 46, 623-635

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

[3] R. Huggins. On the statistical analysis of capture experiments. Biometrika, 1989, vol. 76, 133-140

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

[5] M. Shokouhi, J. Zobel, F. Scholer, S. Tahaghoghi. Capturing collection size for distributed non-cooperative retrieval. In Proceedings of SIGIR'06, 2006, 316-323

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