Incremental Sampling in Query Logs

Ricardo Baeza-Yates
Yahoo Labs
Sunnyvale, USA
rbaeza@acm.org

ABSTRACT

We introduce a simple technique to generate incremental query log samples that mimics well the original query distribution. In this way, editorial judgments for new queries can be consistently added to previous judgments. We also review the problem of how to choose the sample size depending on the types of queries that need to be detected as well as the conditions needed to get a good sample.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval

1. INTRODUCTION

In evaluating search engine quality, we are typically interested in the average performance over a large set of queries. Unfortunately, estimating this performance based on individual queries is not trivial because query frequencies follow a very biased distribution that can be modeled with a power law [2]. Hence, choosing the right sample size as well as the right sample is crucial and can have a dramatic impact on performance estimations as well as in any other measure inferred from the query sample.

Many times sample queries are often selected in an arbitrary fashion, due to many different reasons. For example, the queries are those where we have judgement evaluations and hence we can measure quality performance. However, most of the time those queries do not represent well the original query stream (*e.g.* biased towards popular queries, filtered for adult queries and/or identity information, etc.). Even when the sampling is done correctly, the resultant distribution may not have the same characteristics of the original query stream.

To measure search engine quality, many times editorial quality judgments are used. However, as the editorial judgment of large samples is expensive, we have a trade-off between cost and the error in the performance estimation. For this reason, the editorial work is spread over time and instead of generating a large sample to be used for several months, it is much better to generate incremental samples that reflect the changes in the query distribution but at the same time are consistent with each other. Therefore, we design a new sampling algorithm that addresses these requirements. To the best of our knowledge, we are the first to tackle this case.

The rest of the paper is organized as follows. In the following section we characterize query logs while in Sections 3 and 4 we address the issues of sample size and sample generation, discussing how to measure the goodness of a sample with respect to the original distribution. In Section 5 we present the new sampling algorithm that is able to generate a series of incremental samples, followed by some concluding remarks in Section 6.

SIGIR '15, August 09 - 13, 2015, Santiago, Chile.

ACM 978-1-4503-3621-5/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2766462.2776780.

2. MODELING QUERY FREQUENCIES

A query log is a time-ordered sequence of queries submitted to a search engine. Each query can have one or more attributes associated to the query, such as time, location, etc. The query frequency is usually modeled by a power law (there are other distributions that can be used but they are more complex, see [10]). That is, the query frequencies follow a distribution proportional to $f_1\ r^{-\alpha}$ where f_1 is the frequency of the most popular query, r is the rank of the query when frequencies are sorted from largest to smallest, and α is the exponent of the power law.

The exponent α depends on the specific search engine, how many languages are used in it, etc. For example, in [11], for the case of a Brazilian search engine, α was 0.59, while for another one that uses Spanish was 1.42 [7]. In [3, 8], for the case of a generic search engine, α was 1.84 and 2.38, respectively, more similar to the case of words in text documents. In this paper we use for our examples mobile query logs from 8 different countries obtained in the same period of time from a large web search engine. In our case α is below but close to 1. An example is given in Figure 1 where α is 0.88 if we force f_1 to be the actual value.

An important characteristic of query logs is that the long tail does not match well the power law model, because the tail is much longer than the one that corresponds to the power law fitting the head distribution. Indeed, singleton queries are a large percentage of the query vocabulary (number of different queries), typically 50% in a generic web search query log and in our mobile query logs they are in the range of 61% to 69% (in the case of Figure 1 is 68.8%). In the later case this implies more than 26% of the query

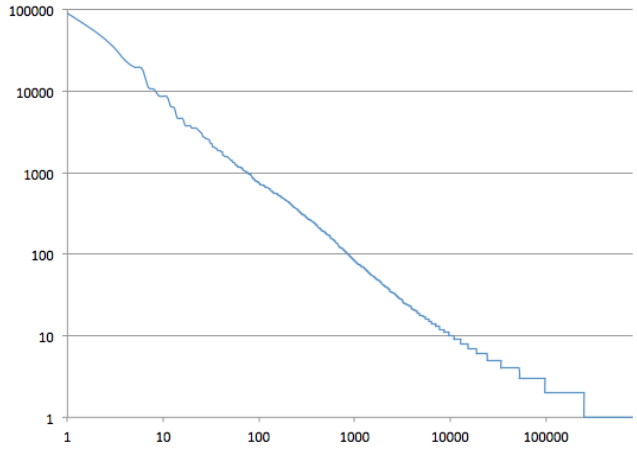

Figure 1: Mobile query log example.

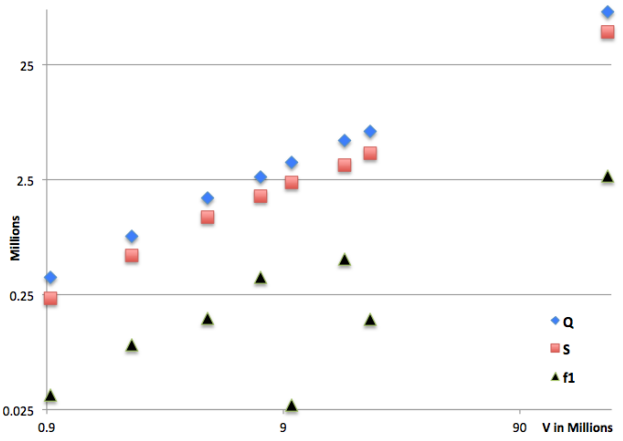

Figure 2: Characteristics of mobile query logs for 8 different countries.

volume (in our mobile samples this percentage ranges from 20% to 27%).

Let Q be the total number of unique queries in the query log and S the number of singleton queries. Hence, a better model for the query frequency is:

$$f_r = \begin{cases} \lceil f_1/r^\alpha \rceil & 1 \le r \le Q - S \\ 1 & Q - S < r \le Q \\ 0 & r > Q \end{cases}$$

Using this model we obtain the following formulas for the total volume of queries V:

$$V = \begin{cases} f_1(Q-S)^{1-\alpha}/(1-\alpha) + S + O(1) & 0 < \alpha < 1 \\ f_1 \ln(Q-S) + S + O(1) & \alpha = 1 \\ f_1/(\alpha - 1) + S + O(f_1 Q^{-(\alpha-1)}) & \alpha > 1 \end{cases}$$

As in practice $S = O(Q)$, we have the following order of magnitude relations with respect to the volume of queries:

$$\begin{array}{lll} Q = O(V) & f_1 = O(V^\alpha) & 0 < \alpha < 1 \\ Q = O(V) & f_1 = O(V/\log V) & \alpha = 1 \\ Q = O(V^{1/\alpha}) & f_1 = O(V) & \alpha > 1 \end{array}$$

In our mobile query logs we have the first case that works well even across countries as shown in Figure 2. Notice how f_1 grows a bit more slowly as α is below 1. The two outliers are countries that use Kanji, so as expected, language also plays a role.

3. SAMPLE SIZE

Most research works use very small query sets, partly due to lack of data, or as we mentioned before, for lack of judged queries. On the other hand, in [9] it is shown that a random query sample of 650 queries is sufficient to reliably estimate if a search engine is better than another search engine. However, this work does not say anything regarding the right sample size if we want to estimate a measure in the query log itself, for example, the fraction of queries that mention a location or a given topic.

Let us assume that we want to be able to detect in the query log sample a certain class of queries that appears with probability p in the query stream. To estimate the sample size we could use the standard error formula for a binomial distribution of parameter p. However the standard error is not good for small values of p. In [4] and the associated comments, several techniques to estimate the error in a Binomial distribution are studied. For large samples, they recommend the Agresti-Coull [1] technique that also works well for values of p closer to 0, which is the case for many applications (it is certainly the case for click through rates in ads or results below the top ranked ones). Although not recommended, for small p also the Clopper-Pearson technique works well [5], but is a bit conservative. At the end we settled for the recommended technique because it also allows to find a formula for the sample size n. Indeed, we obtain

$$n \ge Z^2_{1-\alpha/2}\left(\frac{(1-p)}{\epsilon^2 p} - 1\right)$$

where ϵ is the maximum relative error that we want to have, $1-\alpha$ is the confidence interval and Z is the inverse of the standard normal distribution. For example, if we want to estimate a class of queries that appears 10% of the time with a confidence interval of 90% and a relative error of at most 10%, we need a sample size of at least 2,433 queries. With 650 queries the relative error would be almost 20% or then we can only measure with a 10% error a class of queries that appears 29% of the time or more. In practice, if we want to measure rare queries, such as using the search engine as a calculator, we will need much larger sample sizes.

4. SAMPLING TECHNIQUES

Query samples are weighted sets, in the sense that the same query can be sampled multiple times. Let $\mathcal{S}$ be a sample query set. For every $q \in \mathcal{S}$, we have a sample frequency $f_q(\mathcal{S})$, *i.e.*, the frequency of q in $\mathcal{S}$. From here we can estimate the prior probability p_q of q in the entire query log, if $\mathcal{S}$ is an i.i.d. sample of the distribution of queries:

$$p_q \approx \widehat{p}_q(\mathcal{S}) = \frac{f_q(\mathcal{S})}{\sum_{q' \in \mathcal{S}} f_{q'}(\mathcal{S})} .$$

As query frequencies are discrete and very biased, standard sampling techniques do not work well.[1] Indeed, the two

[1]The alias method of Walker or efficient techniques tailored to discrete distributions do not work well with large and biased data sets.

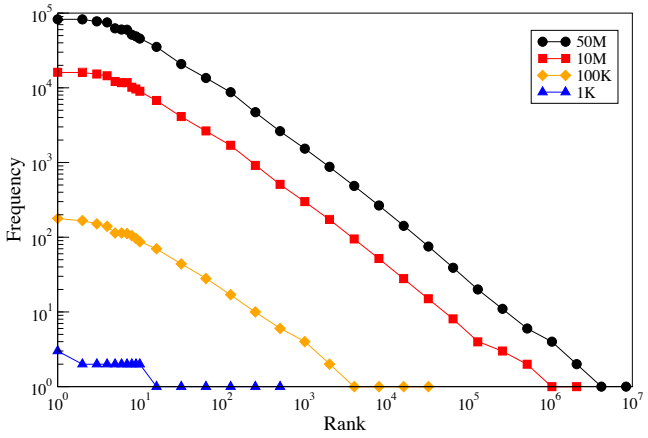

Figure 3: Query frequency distributions with varying sample sizes in a logarithmic scale.

main problems related to our query frequency characterization are:

(1) the frequencies do not mimic well the original distribution, specially for frequent queries, and

(2) the relative size of the set of singletons is quite different.

A good query sampling algorithm should solve the two problems above.

In [12] they analyze the two most used sampling techniques, which suffer of the two drawbacks just mentioned. The simplest one is obtained by removing repeated queries from the sample (but then there are no frequencies and any performance measure will be wrongly estimated). The second and most used one uses $\widehat{p}_q(\mathcal{S})$ above to estimate the real frequencies. However, in power law distributions, the $\widehat{p}_q(\mathcal{S})$ estimator is not accurate unless the sample is very large, or the α coefficient is close to zero (making the distribution close to uniform). We show this problem in Figure 3 by using a query log of 50M queries and construct samples (i.i.d. with replacement) of sizes 1K, 100K, and 10M. Observe that as the sample becomes smaller, the sample distribution approaches the uniform distribution, and then any performance measure underestimates the effect of frequent queries. They propose to partially solve issue (1) by using a much larger sample to estimate the query frequencies, for example, by using the frequencies of the 10M sample to better estimate the frequencies of the 1K sample in Figure 3. This technique allows to alleviate the underestimation of frequent queries. Nevertheless, issue (1) is not completely solved as the most frequent queries are not necessarily the same and (2) still remains. Indeed, the singletons set size for the query log of Figure 1 using the original frequencies to correct the sample frequencies is about half the original (34%).

5. A NEW SAMPLING TECHNIQUE

A solution to issue (1) is sampling via binning in a query log sorted by query frequency. That is, we first put all the most frequent queries, second the second most frequent queries, and so on. As a result, all identical queries will be contiguous as shown in Figure 4. To sample, we divide this frequency sorted query sequence in (almost) equal bins and

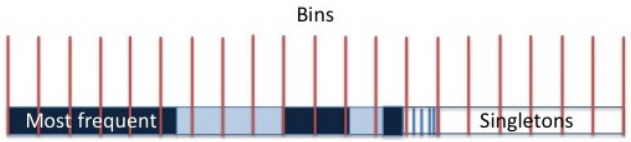

Figure 4: Sampling by binning.

we choose randomly one query of each bin as shown in Figure 4. For this, we just choose at random a number from 0 to the bin size minus 1, selecting the query in that position. Then we jump to the next bin adding the bin size, iterating until we reach the end of the distribution. If there are more than one bin for the same query, we will have repetitions that will give us the frequency of that query in the sample. In practice this algorithm can be simulated by only using the queries and their frequencies.

What is the right binning size? If we want to obtain M queries in the sample, the bin size will be $b = V/M$ for a query log with a volume of V queries. However, as many samples will be repeated the number of unique queries is less than the target sample size M. If we want to obtain M distinct queries in the sample, we need to adjust the bin size to get the closest possible to M.[2] For this we can use binary search to find the right sample size (we need to decrease the bin size to obtain more unique queries). This implies in the worst case $O(\log M)$ iterations.

Notice that all the queries that have original frequency larger or equal than the bin size will be represented in the sample. Hence, the order the queries in the head of the distribution is equal to the original distribution up to that point. In addition, the frequency in the sample will be scaled by the bin size (up to the value of 1). In fact, in the head of the distribution this is similar to stratified sampling, as we are sampling from groups that in most cases have the same query (that is, they are different strata as shown in the left of Figure 4). Hence, we do not need to correct the sample frequency as this frequency is just a scaling of the original frequency (modulo bin size).

The procedure above does not solve issue (2) as now the sample is heavily biased towards the head. To fix this we change the bin size when we reach the singletons in the sample. That is, when the query frequency in the original distribution is less than $b/2$, we stop and we recompute the bin size to match the singletons size ratio. If we want a ratio of $\beta = S/V$ of singletons volume, we now use $b' = (1-\beta)(Q-Q')/(\beta V')$ to achieve the desired ratio, where V is the total query log volume, Q is the number of distinct queries in the query log, and V' and Q' are the current values for the same variables in the partial sample already generated. Notice that in the set of original singletons we are actually doing a uniform sampling and that $b' > b$.

In Figure 5 we show the frequencies of a sample of 750 queries for the query log of Figure 1 using the new technique. In this case the volume ratio of the singletons is the same as in the original query log (26.6%) and as the sample is small, all non singletons queries are identical to the head of the original query log. As we are matching the ratio of singletons, the percentage of singletons with respect to unique queries is a bit higher (75% instead of 69%), so this sample

[2]Depending on M, the final number might be not equal to M due to rounding.

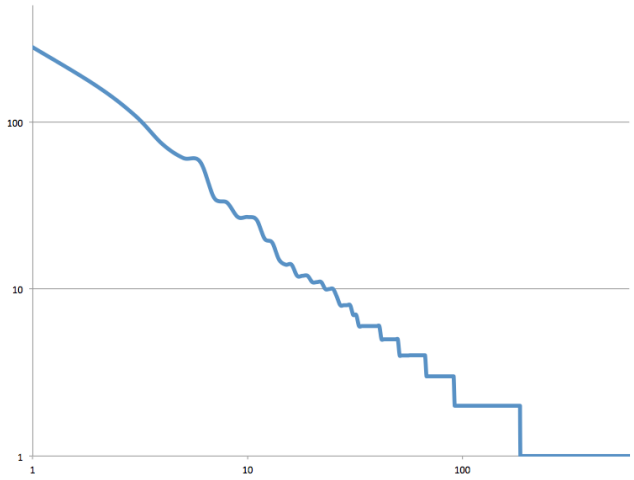

Figure 5: Sample example in logarithmic scale.

has a slight bias to the tail. Notice that the resemblance of both curves is very good, with a similar α.

A simplified deterministic variant of the previous algorithm is to use the queries at the right end of each bin. In the head of the distribution there are no changes as the queries are the same in most bins. In the tail, any query is as good as any query as the order is arbitrary, so in practice it is random. The same argument partially holds in the middle of the distribution and as we are using the right side query, getting again a sample biased towards the tail, being a bit harder than a completely random sample (which implies that we will not overestimate any measure that is biased to the head, *e.g.* commercial queries).

Now we show how we can do incremental samples with this algorithm. First, we select the largest b that is a power of 2 and that satisfies $b = V/M$. This will give us a sample of size M or larger. Now, if we double the sample size b and we use the same algorithm to generate the new sample, most of the queries will be the same (in the head this is trivially true), providing that we use the same initial random value (we can store this parameter as part of the sample). Hence, if we are enlarging a set of judged queries, about half of them will be already judged, so in practice the technique is almost incremental. This is even true if we use a different query log, as the correlation in time of query logs is very high. In [3] they show that the correlation of windows of a few weeks along a whole year was all the time larger than 0.99.

6. CONCLUDING REMARKS

We have presented a new sampling algorithm in query logs that mimics well the long tail of the distribution as well as the most frequent queries. An issue that we do not address, is that sometimes we want to make sure that additional attributes are well represented in the query sample (*e.g.* locations from where a given query is issued). However, the attributes of queries in a sample, in general, will not follow the distribution of a given attribute as the attribute distribution might not be uniform nor necessarily correlated with the query distribution. So, if a sample for a given attribute is needed, we can apply the binning technique on that attribute, independently of the queries. Otherwise, if we need all attributes associated to a query, we should apply the binning technique for each attribute separately, according to the different strata groups that can be defined in the query (e.g. top queries, singletons, etc.). This will give a more truthful random sample (notice that in general mimicking two or more non-correlated distributions is not possible). Another possibility is to apply the same algorithm to an expanded query log where the frequencies are taken over unique tuples that contain queries and the values of attributes of interest (continuous attributes such as time must be binned, *e.g.* by hours).

Another conclusion of our study, is that we may need different query frequency models for different alphabets and/or languages. Also, if the query distribution has a longer tail, we may need to take in account duplets (queries that appear twice), triplets, etc.

7. REFERENCES

[1] A. Agresti and B.A. Coull. Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician* 52(2):119–126, 1998.

[2] R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval: the concepts and technology behind search (2nd ed.). *Pearson Education*, 2011.

[3] R. Baeza-Yates, A. Gionis, F. Junqueira, V. Murdock, V. Plachouras and F. Silvestri. Design Trade-Offs for Search Engine Caching. *ACM Transactions on the Web*, 2(4), Article 4, 2008.

[4] L.D. Brown, T.T. Cai, and A. DasGupta. Interval estimation for a binomial proportion. *Statistical Science*, 16(2):101–133, 2001.

[5] C. Clopper and E.S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika* 26: 404–413, 1934.

[6] W. G. Cochran. Sampling Techniques (third ed.) Wiley & Sons, 2007.

[7] . R. Baeza-Yates and F. Saint-Jean. A Three Level Search Engine Index based in Query Log Distribution. In *SPIRE 2003*, Springer LNCS, Manaus, Brazil, October 2003.

[8] R. Baeza-Yates, A. Tiberi. Extracting Semantic Relations from Query Logs. In *ACM KDD 2007*, San Jose, California, USA, August 2007, 76–85.

[9] E. C. Jensen, S. M. Beitzel, O. Frieder, and A. Chowdhury. A framework for determining necessary query set sizes to evaluate web search effectiveness. In *Special Interest Tracks and Posters of the 14th Int'l Conf. on World Wide Web*, pages 1176–1177, 2005.

[10] M.E.J. Newman. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics* 46 (5): 323–351, 2005.

[11] C. Saraiva, E. de Moura, N. Ziviani, W. Meira, R. Fonseca, and B. Ribeiro-Neto. Rank-preserving two-level caching for scalable search engines. In *Proc. 24th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 51–58, 2001.

[12] H. Zaragoza, B.B. Cambazoglu, R. Baeza-Yates. Web search solved? All result rankings the same?. In *Proc. CIKM 2010*: pages 529–538, 2010.