Predicting New Search-Query Cluster Volume

Jacob Sisk, Cory Barr

December 14, 2007

1 Problem Statement

Search engines allow people to find information important to them, and search engine companies derive their profit from delivering paid advertisements in response to user queries as well as "organic" results. The paid advertisements are matched to the user's query, usually by way of shared similar keywords or topics. This poses many problems for search-engine companies and their advertisers, almost all of which stem from the long tail of the distribution of user queries. Search-engine companies have spent tremendous effort monetizing this tail by providing imaginative techologies to match low-frequency queries ("Riemannian manifold") to relevant advertisements ("Springer book sale").

Novel queries are an under-monetized segment of this long tail. The world changes rapidly. New products, news, gossip, memes, stories and ideas consistently emerge. Predicting the volume of queries about these novel topics is the subject of this report.

If we observe a novel query, the likelihood of never seeing that query again is 67.1% (measured one month past the query's initial appearance). On the other hand, if we observe that novel query occuring a few times (even better if by a few different people), it becomes more probable the query is about some new idea or thing, and we are more likely to see it again in the future.

2 Data

2.1 Query Logs, Dataset Construction

To build a corpus of novel queries, we constructed a Bloom filter 30 gigabytes in size with an estimated false positive rate of less than 0.01% containing over 25 billion queries issued to the Yahoo! search engine in 2005 and 2006. We then sampled 2.5% of the search traffic from January 2007, retaining only queries not issued in 2005 or 2006. There are 16.8 million unique queries in this sample. Since we are looking for new topics and believe queries about new topics may take many different lexical forms, we Porter-stemmed the queries. Then, for each novel query q occuring at time t_0, we built a regular time series beginning at t_0 using a period of one minute that recorded 1) the number of times q was re-issued in each subsequent minute (for up to 28 days), 2) the number of new users in each minute $t_0 + i$ who issued q but never issued q in

$f(q^{(i)}, t_0, t_0+5)$	$\lvert q_i \rvert$	$\Delta = 5$	10 min.	30 min.	60 min.	3 hrs	1 day	7 days	28 days
1	16.5m	0.0033	0.0049	0.0071	0.008	0.009	0.0116	0.0184	0.0275
2	213k	0.0193	0.0264	0.0347	0.0373	0.0392	0.0429	0.0506	0.0595
3 to 9	12k	0.0662	0.0775	0.0886	0.092	0.0952	0.1006	0.1124	0.122
≥ 10	155	0.1355	0.1355	0.1355	0.1419	0.1419	0.1484	0.1484	0.1548

Table 1: Probability of query repetition given frequency in first 5 minutes

$u(q^{(i)}, t_0, t_0+5)$	$\lvert q_i \rvert$	$\Delta = 5$	10 min	30 min	60 min	3 hrs	1 day	7 days	28 days
1	16.8m	0.0036	0.0052	0.0075	0.0084	0.0094	0.012	0.0189	0.028
2	2472	0.0825	0.1331	0.1913	0.2091	0.2229	0.2573	0.2876	0.3139
≥ 3	200	0.22	0.325	0.41	0.445	0.47	0.495	0.5	0.515

Table 2: Probability of query repetition given user count in first 5 minutes

$t_0, ..., t_0 + i - 1$, and 3) the number of repeat users who issued q in minute $t_0 + i$ and also issued q at some point prior to $t_0 + i$.

2.2 Descriptive Statistics

To help design features, we examined how informative tallies of query frequency were in small time windows early in the history of a novel query. Specifically, if a novel query $q^{(i)}$ is first issued at t_0 and has frequency $f(q^{(i)})$ in $[t_0, t_0 + \delta)$–which we will denote $f(q^{(i)}, t_0, t_0 + \delta)$–it is instructive to empirically estimate the likelihood that $q^{(i)}$ is issued in some larger, later time window $[t_0 + \delta, t_0 + \delta + \Delta]$. It is equally instructive to do this considering the number of novel users issuing $q^{(i)}$, which we denote $u(q)$, in $[t_0, t_0 + \delta)$. This gives us an estimate of

$$P\left(f(q^{(i)}, t_0 + \delta, t_0 + \delta + \Delta) > 0 \| f(q^{(i)}, t_0, t_0 + \delta\right) \quad (1)$$

Table 1 shows an estimate for the conditional probability of repeated query issuance given the frequency we observe for that query in the first five minutes of its lifespan. This table demonstates that seeing a novel query more than once in five minutes greatly increases the chances we will see that query again, both in the next few minutes as well as up to a month in the future. Table 2 demonstates we can estimate the same reoccurrence for $u(q, t_0 + \delta, t_0 + \delta + \Delta)$, and the effect is even stronger.

2.3 Clustering: From Queries to Topics

After examining the recorded queries, we felt clustering semantically and temporally related queries would provide aggregate cluster statistics and much more informative training data for a supervised learning problem than examining individual query behavior. In addition, a substantial portion of novel queries are not useful to search engine advertisers, including navigational queries (13.7% of queries in our sample), DNS errors, etc. We hypothesised a cluster-inclusion criterion could be designed that many of these unwanted queries would not pass, and eliminating unclustered queries would improve our training set.

label	coverag execut hussein videotap jazeera	carbon collector nikki spe ne
queries	coverag of saddam execut al jazeera coverag of saddam execut videotap of saddam hussein execut saddam hussein execut full coverag saddam hussein hospit bed execut	ne for spe carbon nikki ne spe free download of ne for spe carbon ne for spe carbon g carbon collector nikki spe ne ne for spe carbon collector locat p

Table 3: Example Query Clusters and Their Labels

We clustered using an agglomerative algorithm based on the Jaccard distance between two queries. This distance was extended to a Jaccard distance between a query and a cluster by means of comparing a query to a computer-generated cluster label of tokens in the cluster selected by a tf.idf criterion. This technique did discard most non-informative queries in addition to ameliorating data sparsity issues. Unfortunately, despite implementing an inverted index to eliminate comparing queries to others with no token intersection, the algorithm was computationally expensive, and we had to subsample down to 0.025% of the queries from January 2007. Table 3 presents examples of some clusters and their labels.

It should be noted that the scope of this study is to examine the feasibility of predicting search-query volume through supervised learning. However, focusing on clustering lies outside our present scope. Consequently, our experiments are designed to confirm the feasibility of predicting search-query cluster volume and persistence given a reasonably well-clustered training set, which our initial clustering method achieved.

3 Predicting Future Query Volume

3.1 Experimental Framework

For every experiment, our design matrix was constructed by breaking up the first $T1$ of a cluster's query volume history into a regular time series of n pieces some ε apart, recording query volume (or $log(1+volume)$) for each of the time slices. The dependent variable was the total query volume observed in the next $T2$ units of time for that cluster.

3.2 Regression

Due to data sparsity, logistic regression performed unpredictably. Linear regression faired better. Regressions were compared to a baseline model of predicting zero future query volume. For almost all choices of T_1, T_2 and ε, regression outperformed the baseline, sometimes radically. Table 4 shows root mean-squared-error for the regression with the baseline in parentheses. Statistics are the result of 10-fold cross validation.

3.3 Support Vector Machine Prediction

We had constraints on computing time, but preliminary experiments showed an SVM with a polynomial kernel could significantly outperform the linear model. Furthermore,

	$T_2 = 1440$ (1 day)	$T_2 = 10080$ (1 week)	$T_2 = 40320$ (28 days)
$T_1 = 30$ ($\frac{1}{2}$ hour)	$\varepsilon = 5 : 0.22(0.23)$	$\varepsilon = 5 : 0.40(0.52)$	$\varepsilon = 5 : 0.32(0.85)$
$T_1 = 60$ (1 hour)	$\varepsilon = 5 : 0.21(0.22)$	$\varepsilon = 5 : 0.40(0.52)$	$\varepsilon = 5 : 0.32(0.84)$
$T_1 = 240$ (4 hours)	$\varepsilon = 5 : 0.21(0.21)$ $\varepsilon = 30 : 0.20(0.21)$ $\varepsilon = 60 : 0.20(0.21)$	$\varepsilon = 5 : 0.41(0.52)$ $\varepsilon = 30 : 0.40(0.51)$ $\varepsilon = 60 : 0.40(0.52)$	$\varepsilon = 5 : 0.34(0.84)$ $\varepsilon = 30 : 0.32(0.84)$ $\varepsilon = 60 : 0.33(0.84)$
$T_1 = 1440$ (one day)	$\varepsilon = 5 : 0.24(0.22)$ $\varepsilon = 30 : 0.20(0.22)$ $\varepsilon = 60 : 0.20(0.22)$ $\varepsilon = 360 : 0.20(0.21)$	$\varepsilon = 5 : 0.47(0.51)$ $\varepsilon = 30 : 0.40(0.51)$ $\varepsilon = 60 : 0.39(0.50)$ $\varepsilon = 360 : 0.40(0.51)$	$\varepsilon = 5 : 0.72(0.82)$ $\varepsilon = 30 : 0.36(0.82)$ $\varepsilon = 60 : 0.35(0.82)$ $\varepsilon = 360 : 0.34(0.82)$
$T_1 = 10080$ (one week)	$\varepsilon = 30 : 0.17(0.18)$ $\varepsilon = 60 : 0.17(0.18)$ $\varepsilon = 360 : 0.17(0.18)$ $\varepsilon = 1440 : 0.17(0.18)$	$\varepsilon = 30 : 0.40(0.45)$ $\varepsilon = 60 : 0.39(0.45)$ $\varepsilon = 360 : 0.38(0.45)$ $\varepsilon = 1440 : 0.38(0.45)$	$\varepsilon = 60 : 0.51(0.70)$ $\varepsilon = 360 : 0.48(0.70)$ $\varepsilon = 1440 : 0.47(0.70)$

Table 4: Linear Regression for Future Query Volume Prediction

the SVM did a very good job of predicting outliers. This is important, since outlier queries are likely the most easily monetizable.

Figure 1 provides evidence of the model's accuracy. The graph displays a distinctly modal tendency. Our model either predicts correct search volume extremely well, or predicts no volume. However, above a certain volume threshold the SVM performs with exceptional precision. Since our goal is to ultimately predict novel, persistant large-volume clusters for search-engine monetization, performance below some volume threshold is likely to be unimportant or perhaps entirely irrelevant. Therefore, the performance of this model implies fitting a model using an SVM provides satisfactory application performance.

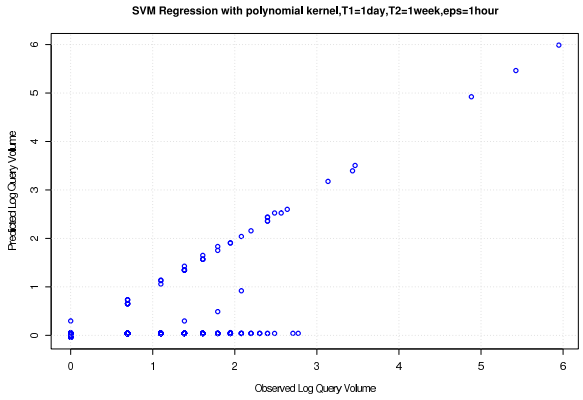

Figure 1: SVM Regression (T_1 =1 day, T_2 =1 week) (3rd degree polynomial kernel)

3.4 Markov Model

We felt a Markov-model prediction system could be appropriate for our time-series data. We defined states as discretized volume levels with the following ranges: 0,

	START	0	1i	1d	2i	2d	10i	10d	20i	20d	100i	100d
START	0.07	0.13	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
0	0.00	0.89	0.03	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1i	0.00	0.92	0.02	0.05	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00
1d	0.00	0.92	0.03	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2i	0.00	0.80	0.05	0.08	0.05	0.02	0.00	0.00	0.00	0.00	0.00	0.00
2d	0.00	0.86	0.04	0.05	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.00
10i	0.04	0.08	0.04	0.08	0.20	0.08	0.16	0.08	0.04	0.04	0.04	0.04
10d	0.04	0.11	0.04	0.04	0.04	0.36	0.04	0.14	0.04	0.04	0.04	0.04
20i	0.03	0.03	0.06	0.06	0.03	0.03	0.09	0.03	0.34	0.20	0.03	0.03
20d	0.03	0.08	0.03	0.03	0.03	0.03	0.08	0.13	0.13	0.34	0.03	0.03
100i	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.13	0.07	0.07	0.07
100d	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.19	0.06	0.06

Table 5: Markov Model State-Transition Probabilities

1, 2-9, 10-19, 20-99, 100-999, and 1000+. In addition, each volume-bucket state is partitioned into an "increasing" and "decreasing" state, where "increasing" is defined as having at least 50% of the query volume in the latter half of the time span. In our experiment, each state covers one day. Table 5 presents state-transition probabilities.

A phenomenon occurs when the volume reaches 20 queries in a day. The state transition indicates the probability of maintaining at least the current volume increases 30%, which suggests a possible metric for cluster persistence.

Our Markov-model prediction system consistently predicts a daily total volume of 0 if the previous day had volume less than 10. In our experiments, the prediction for higher-volume clusters was always several states too low (but never 0). We believe a second-order Markov model could offer greater prediction. The Markov model also has two advantages over the SVM. First, it can readily model query volume per day of the week, which differs radically. Second, it offers the distinct advantage of being intuitively understandable for non-quantitive employees. However, we believe the SVM model remains the superior choice.

4 Future Work

Though tangential to our supervised regression study, it seems clear that increasing performance of query clustering is the most likely means to increase the ultimate utility of the supervised-learning research. Consequently, current efforts are focused on unsupervised clustering techniques that incorporate cluster-inclusion criteria and automatic determination of the optimal number of clusters.

It would also be worthwhile to add richer features. Specifically, including standard *ARMA* time series features (second order, differenced, etc) and to comparing their performance to that of an SVM on the first order features with a non-linear kernel might prove informative.

The primary motivation for this research is to quickly alert online advertisers about emerging topics for which people are searching. Advertisers "bid" on phrases they think will be issued by users who want their products or services. Consequently, a query (or query in a cluster) becoming bidded by an advertiser is as useful a dependent variable as future query volume. Behavior encoded in the time-series may help predict biddedness. However, biddedness might prove best handled as an addition classification problem once high-volume, persistant clusters are identified.