Causal Relation of Queries from Temporal Logs

Yizhou Sun[1], Ning Liu[2], Kunqing Xie[1], Shuicheng Yan[3], Benyu Zhang[2], Zheng Chen[2]

[1]State Key Laboratory on Machine Perception, Department of Intelligent Science, Peking University
{pekingsun, kunqing}@cis.pku.edu.cn

[2]Microsoft Research Asia
Beijing 100080, P.R.China
{ningl, byzhang, zhengc}@microsoft.com

[3]University of Illinois
405 N. Mathews Ave.
Urbana, IL 61801
scyan@ifp.uiuc.edu

ABSTRACT

In this paper, we study a new problem of mining causal relation of queries in search engine query logs. Causal relation between two queries means event on one query is the causation of some event on the other. We first detect events in query logs by efficient statistical frequency threshold. Then the causal relation of queries is mined by the geometric features of the events. Finally the Granger Causality Test (GCT) is utilized to further re-rank the causal relation of queries according to their GCT coefficients. In addition, we develop a 2-dimensional visualization tool to display the detected relationship of events in a more intuitive way. The experimental results on the MSN search engine query logs demonstrate that our approach can accurately detect the events in temporal query logs and the causal relation of queries is detected effectively.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: *Time series analysis*

General Terms: Algorithms, Measurement.

Keywords: Search engine query log, time series, and causal relation.

1. INTRODUCTION

Owing to the great potential to improve the performance of search engine and uncover the personal behavior habits of the users, query log analysis has attracted much attention in both academic and industrial area. However, to the best of our knowledge, no work has noticed the fact that a sharp increase of one query's submitting frequency may indicate the increase of another query's, namely, there may exist causal relation between different queries. In this paper we study a new problem of mining causal relation between different queries based on the time series data extracted from query logs. If a query is submitted much more often than normally, it usually means that a specific event related to this query is happening. To uncover the underlying causal relationships, we present a system, in which the events are extracted from each query time series (QTS) followed by the Event Causality Test (ECT), and then the causal relation is re-ranked by Granger Causality Test (GCT).

2. CAUSAL RELATIONS OF QUERY

2.1 Causal Relations

Causal relation is also referred to as cause-and-effect relation, which is one of the basic relation types in ontology. If one query's change always causes another query's change within a time period, we say that there is a causal relation between the two queries. More formally, the causal relation between two objects *A* and *B* can be defined as,

Definition 1. *A* causally precedes *B* if the following conditions hold:
(1) *A* and *B* are in the same process and *A* was executed before *B*;
(2) There exists *C* such that *A* causally precedes *C* and C causally precedes *B*.
And we note this relationship as $A \prec B$.

Granger Causality Test (GCT) was first proposed by Clive W. J. Granger in 1969 [1] and popularized by Sims at [2] which provided a statistic method to judge if two variables have causal relation. Although GCT may provide causal relations with high confidence, we cannot directly use it due to its high computational complexity. The complexity is $O(mn^2)$ for building the whole causal relation graph, where *m* is the length of the time series and *n* is the query number. In this way, we propose an efficient method called *Event Causality Test* (ECT) method to filter a large number of unrelated queries for a given query, which calculates the causal relation coefficient between queries by the queries associated with events.

2.2 Event Model of Queries

The sharply increasing of query frequency in time series means that some events happened. For instance, after Microsoft released Windows Vista, the query "Vista" is used much more frequently within a period of time. There appears a peak in the time series of query "Vista". Thus the burst in the query frequency time series is often an intuitive representation of some real events. Mathematically, we define events as:

Definition 2 (Events): an event e^q on a query q is a period in which its search frequency is above a given threshold f_b, and at least the frequency at one time point during the period is above a given threshold f_c. Climaxes of an event are query frequencies that are above f_c, and whose values are the maximal within a certain neighborhood. We denote the i^{th} event on q as e_i^q.

The first threshold f_b is a baseline that can determine the event period, and the second threshold f_c is the climax threshold that can pick out the periods with large influence. We represent events as: $e^q = < id, t_b, t_e, t_c, ts_c >$, where id is the ID of the event; t_b and t_e are the start time and end time of the event; t_c is the time that climax happens; and ts_c is the time series value at t_c. Thus $t_e - t_b$ is the time period that the event lasts, and $(t_e - t_b) \times ts_c$ is a good indicator of the importance of the event.

WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

We call $S(e^q)=\frac{1}{2}(t_e-t_b)\times ts_c$ as the area of e^q to measure the importance of e^q .The definition is illustrated in Figure 1: we can see that there are two events (as shown in shadow) in the query "Mother's day".

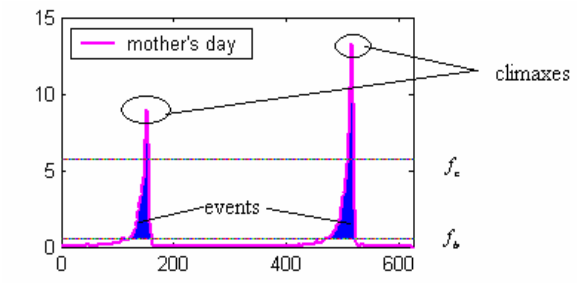

Figure 1. Illustration of events of "Mother's day"

The processing of transforming time series into event series can be shown in the following Table 1.

Table 1. Time series transform into event series

Step 1. Finding: using 3σ rule of Gaussian distribution to detect a group of event candidates based on the **Definition 2**.
Step 2. Merging: if they share the same appearing period, we will combine them.
Step 3. Splitting: if the frequency at t_0 between two climaxes below a threshold f_s , the ending time of the first event will be set as t_0 and the second event's beginning time will also be set as t_0 .
Output: An event series of the query $Q=\{e_1^Q,e_2^Q,...,e_M^Q\}$ and $e^q=<id,t_b,t_e,t_c,ts_c>$

2.3 Causal Relation between Queries

Based on the definitions of the Causal Relations and Events, we first give the definition of the **Probability of Event** e_1 **Cause Event** e_2 based on the geometrical framework as (4.1):

$$P(e_1 \prec e_2)=\begin{cases} 0 & if\ t_b(e_1)\ \text{after}\ t_b(e_2) \\ \dfrac{S(e_1)\cap S(e_2)}{\max(S(e_1),S(e_2))} & else \end{cases} \quad (2.1)$$

where $S(e_i)$ is the area of the event e_i and it can be calculated by $S(e_i)=(t_e-t_b)\times ts_c/2$ which can be calculated by the feature of $e^q=<id,t_b,t_e,t_c,ts_c>$. Intuitively, the longer the time for e_i lasts, the more important it is. The larger frequency of e_i is, the more important it is. Given the causal relation between events, the causal relation between queries is defined as follows:

Definition 3. Given two queries $A=\{e_1^A,e_2^A,...,e_{M^A}^A\}$ and $B=\{e_1^B,e_2^B,...,e_{M^B}^B\}$, the probability that a query A causes a query B is

$$P(A\prec B)=\frac{\sum_{i=1}^{M^A}S(e_i^A)P(e_i^A\prec B)}{\sum_{i=1}^{M^A}S(e_i^A)}=\frac{\sum_{i=1}^{M^A}S(e_i^A)\max_{1\le j\le M^B}P(e_i^A\prec e_j^B)}{\sum_{i=1}^{M^A}S(e_i^A)} \quad (2.2)$$

Intuitively, for an event e_i^A of query A, the longer and the larger frequency is, the more important it is. Then the importance/weight of event in a query can be denoted by the area $S(e_i^A)/\sum_{i=1}^{M^A}S(e_i^A)$.

The probability of e_i^A cause query B is measured by the maximum probability values among all $P(e_i^A\prec e_j^B)$. Finally, the **Probability of Query A Cause Query B** is determined by $P(A\prec B)$ in (2.2).

3. EXSPRIMENTS

The dataset we used here was collected from MSN search engine (http://search.msn.com/). The dataset is a daily aggregation of users' submitting frequency for each query, from December 2nd, 2003 to August 8th, 2005, 625 days in total. To get the overall quality comparison, we sample 150 queries from the dataset, and apply ECT and ECT-GCT methods on them. We ask eight people to label the top 100 results (include the input query) obtained by the ECT method, and thus we can get the accumulated precision graph as in Figure 2. The *x*-coordinate of Figure 2 means the top-*k* position, where *k* is from 1 to 99 (the input query is excluded, for we can't apply GCT to the totally the same two time series as this will cause singular matrix). The y-coordinate is the accumulated precision at a top-k position, which is calculated as: $\frac{\#(\text{label as 1 in top-}k\text{ results})}{k}$

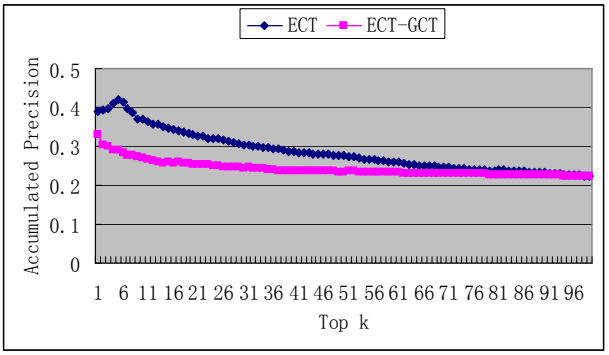

Figure 2. Accumulated precision of ECT and ECT-GCT

4. CONCLUSION

In this paper, we studied a new problem of detecting causal relation between queries based on query time series data extracted from search engine query logs. An effective and efficient method was proposed to solve this problem by using the event information involved in each query time series. To further improve the precision of the causal relation, GCT was implemented to re-rank the top-*k* results. The experimental results show that the integrated ECT-GCT approach provided encouraging results, and it greatly outperformed ECT with a reasonably higher complexity.

5. ACKNOWLEDGMENTS

This work is supported by the National High-Tech Research and Development Plan of China (863) under Grant No.2006AA12Z217

6. REFERENCES

[1] C. W. J. Granger, "Investigating Causal Relations by Econometric Models and Cross-spectral Methods", Econometrica, vol. 37, pp. 424-438, 1969

[2] C. A. Sims, "Money, Income, and Causality", the American Economic Review, vol. 62, pp. 540-552, 1972.