The Pennsylvania State University

The Graduate School

College of Engineering

ANALYSIS AND PREDICTION OF WEB USER INTERACTIONS USING TIME SERIES ANALYSIS

A Thesis in

Electrical Engineering

by

Vijay Vyas Mohan

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2009

The thesis of Vijay Vyas Mohan was reviewed and approved* by the following:

Bernard J. Jansen
Assistant Professor of Information Sciences and Technology
Thesis Co-Adviser

George Kesidis
Professor of Electrical Engineering and Computer Science and Engineering
Thesis Co-Adviser

John Metzner
Professor of Electrical Engineering and Computer Science and Engineering

Kenneth W. Jenkins
Professor of Electrical Engineering
Head of the Department of Electrical Engineering

*Signatures are on file in the Graduate School

Abstract

In this research, the methodology of time series analysis is studied and adapted to analyze the temporal facets of individual user interaction with search engines as recorded in search logs. A massive search engine query log with more than 3.5 million queries over a period of three months is first enhanced with factors which identify each user query by user intent, type of query, and other aspects. Temporal characteristics are used to obtain additional factors such as the elapsed time between query searched and result clicked along with tracking seasonal components like daily and weekly cycles for each query. Two popular approaches to time series analysis are explored – the Box-Jenkins ARIMA method and the regression method. A framework is provided for using the methodology of time series analysis to predict the future actions of the individual user. Time series regression models are obtained for every active user to predict the rank of the results clicked one-step ahead of time. The aggregate statistical analysis of the obtained time series models are used to recognize similarities in user behavior for Web search and identify significant predictors of rank clicks. Predicting Web search engine users' future actions and analyzing their searching behavior could be very useful for optimizing online advertisements and web service providers.

Table of Contents

List of Figures

List of Tables

Acknowledgement

First and foremost, I would like to offer my sincerest gratitude to my advisor, Dr. Jim Jansen, who has supported me with his knowledge, guidance, patience and encouragement during the course of my research. He gave me a very interesting research topic and steered me in the right direction whenever I needed it and without him, this thesis would not have been possible.

I would also like to thank my committee members, Dr. George Kesidis and Dr. John Metzner for supporting my thesis and research.

Finally I would like to thank my parents for their strong support in whatever I do. I thank my girlfriend, Nandini, for her constant motivation and faith in me, and my friends at Penn State, without whom, I would not have been able to complete this research study.

Chapter 1

Introduction and Overview

1.1 Introduction

The World Wide Web (Web) contains at least 25.1 billion pages as of 2009 [1] and is still growing. The consequence of its growth is the increasing importance of Web search engines that are used to search and locate relevant information. One of the main purposes of using search engines is to easily find relevant information on the Web without having to know the exact address of the Website or Webpage. Therefore, it is important for search engines to display worthwhile results at the top of the results listing in response to a user query.

For Web searching, a search log is defined as an electronic record of interactions that have occurred during a searching episode between a Web search engine and users searching for information on that Web search engine [2]. Search logs usually contain data such as the query submitted, time of search, rank of the result clicked, and other related fields. These fields are logged every day for millions of users who use the major search engines, and the logs contain information that could potentially be extremely useful for learning more about the users and predicting trends in their usage. This is important because it is directly related to developing search engines with personalization features to the users. Agichtein et al [3] state that accurate modeling and interpretation of user behavior have key applications to ranking, click spam detection, and web search personalization. With major search engines

increasingly playing a bigger role in online advertising, it is crucial for them to understand its users and the manner in which they use its services.

Most of the research using Web search logs have traditionally focused on descriptive aspects of the search engine logs at three common levels of analysis, namely *term*, *query*, and *session* [4]. A variety of statistical results such as number of terms in a query, average number of queries per day, or average number of results clicked per session for a user can be obtained provided the log contains the appropriate data. However, search logs indirectly contain time series data because the time of query search is typically stored. Data like the query searched and rank of the result clicked are linked to the time of query search; therefore, one can view the data as a time series process. If some sort of user identification is also logged, then one could potentially generate time series formulas that model the searching pattern for every user on the engine. Analyzing the temporal facets of user interactions stored within the search logs could be used to develop forecasting models for Web searching at the individual level.

1.2 Motivation

As search engines get more advanced, they are becoming more personalized for each user. Most current search engines have the means to identify every unique user by having them sign in to customizable Web pages to personalize the content they want to see. Examples include My Yahoo (my.yahoo.com), My AOL (my.aol.com),

and iGoogle (www.google.com/ig) wherein the user can personalize and choose what to see, such as email, news, weather, stock prices etc. Having the user sign in to his/her Web search page has the inconspicuous advantage for the search engine to monitor their personal search usage even if they are using a public computer where the Internet Protocol (IP) address cannot be used to keep track of a particular user's search history. The arrival of desktop search bars also has the same advantage of being able to maintain a history of a particular user's searching activities spread over different sessions.

This gives rise to an aspect of log research focusing specifically on the individual user-system interaction, which was not possible before. Each user has his own characteristic that makes him unique compared to other users of the search engine. For example, some users may always click on the first uniform resource locator (URL) after a query search irrespective of whether that result is relevant to them or not while others read through the snippets of the results in the first page and then click the one they deem to be most relevant. Research focusing on the individual user behavior can lead to forming predictive equations that define the individual's searching patterns.

There are a number of interesting factors in a unique user-system interaction that could have a useful predictive value. User clicks on the hyperlinks are a source of endorsement and are general indicators that the document is of interest to that user.

The rank of the document clicked is another valuable factor that has predictive importance. Even if the predicted rank is the one of first result on the results page, the analysis would be important because we can directly pre-load the landing page the user would have clicked, for example. Predicting the number of terms in the user's query is another factor that would be useful because it has been shown that the query length has a positive effect on clickthrough which implies that a bigger query length increases the clickthrough rate (CTR) [5], which increases the value of that search engine results page (SERP) to the user and advertisers.

Most of the user queries reflect a particular user intent, which can be broadly classified as *informational*, *navigational*, or *transactional* [6]. Web search engines can help people in finding the resources they are looking for by more clearly identifying the intent behind the query. Even though the user intent has been shown to have no impact on the clickthrough rate [5], it could possibly be a significant factor in predicting the rank of the result clicked to the search engines for optimizing their search results. Predicting the change in the search pattern of the user queries could also be of extreme importance in the development of intrinsic automated assistance to searches. If we are able to predict the next type of query modification one step ahead for the individual user, it could potentially be used to optimize Web search, as shown in [7].

In this thesis, we develop models to identify predictive characteristics of the individual user for some of the above factors by analyzing each user's unique past history using time series analysis. A time series is the collection of quantitative observations of an entity usually at regular intervals, and Web search logs contain many such entities that are important factors in search research. We use a transaction log from AOL search (www.aol.com), which is a top 5 ranked search engine [8]. The log contains more than 3.5 million queries sorted by anonymous user identification (ID) and sequentially arranged over a period of three months. In our research, we obtain equations that define an individual user's search patterns in order to predict the user's future actions. We believe that forecasting the individual user's actions could be very useful for optimizing online advertisements and Web search providers. It has been shown that searchers' behavior across different search engines is very similar [9]; hence, we believe that the methodology used in this research and the results obtained will be applicable to a wide range of search engines.

1.3 Thesis Overview

This thesis is structured as follows. In chapter 2, we first provide an overview of key concepts and previous work related to Web search transactional log analysis. In chapter 3, we introduce the methodology of time series analysis from a statistical point of view and explore two of the best known approaches, the time series regression method and the Box-Jenkins [10] or AutoRegressive Integrated Moving Average (ARIMA) method to develop our equations. Although each method has its

own advantages, we opt for the regression method as it could handle the technically discrete unevenly spaced time series data of the individual user that is contained in the search engine transaction logs. In chapter 4, we present our research design and analysis consisting of three stages: data collection, data preparation, and data analysis. A detailed walk-through of the analysis for a sample user is also given. In chapter 5, we analyze the results obtained from the time series analysis and present the clustered statistical results and aggregate model significance. Chapter 6 provides the consolidated summary of this thesis along with future work.

Chapter 2

Background and Literature Review

There have been several studies related to Web search log analysis but not as many as one might expect. Jansen [2] states that the reason could be because there are not enough published works that provide an organized study of how to use the search logs to support the study of Web searching and none of the works give a comprehensive explanation of the methodology used. His study addressed the use of search log analysis for the study of Web searching and defined a three-stage process comprising of data collection, preparation, and analysis.

1.1 Descriptive Studies

The first studies on search engine user behavior started in the early 1990s. Belkin [11] in 1993 stated that one can classify searching episodes in terms of (1) *goal* of the interaction, (2) *method* of interaction, (3) *mode* of retrieval and (4) *type of resource* interacted with during the search. Although the study was conducted with library systems as its perspective, it could be associated with Web searching which shared a similar viewpoint. The initial studies as well as the bulk of research on transaction log analysis of Web search logs has been primarily descriptive in nature [9], [12], [13]. For example, Jansen et al. [9] studied the characteristics and changes in Web searching from nine different search engine logs and found that users are viewing fewer pages than before and the use of US based search engines differ from searching

on European based search engines. Nancy et al. [12] analyzed the queries for subject content based on co-occurrence of terms within multi-term queries using hierarchical cluster analysis and found similar relationships among different subject categories. A large number of studies were statistical in nature and involved the statistical characteristics of the user queries like finding the average number of terms in the query or number of queries per session.

The other dimension to descriptive studies of search logs was content-based behavior. One approach was to conduct this analysis of the search engine logs at three common levels of analysis, namely *term*, *query*, and *session* [4]. At the *term* level, Silverstein et al [14] studied the interaction of terms within queries and showed that web users differ significantly from the user assumed in the standard information retrieval literature. The analysis of query terms and analysis of query topics were at the *query* level and analyzed specific types of search engine queries such as multimedia queries and textual queries. The *session* level analysis was mainly concerned with analysis of search behavior within a session or across multiple searching sessions. The study of what constituted a session and detection of session boundaries were also performed [15].

1.2 Predictive Studies

However, log research is now moving towards more predictive aspects. As initial efforts in this area, Beitzel, Jensen, Chowdhury, Grossman, and Frieder [16] reviewed

a log of hundreds of millions of queries and found that query traffic from particular topical categories differed both from the query stream as a whole and from other categories. This analysis provided valuable insight for improving retrieval effectiveness and efficiency. Jansen, Booth, and Spink [17] automatically classified queries as *informational*, *navigational*, or *transactional*, and achieved an accuracy of 74 percent. They provided Web search engines with the knowledge for more precisely associating user goals with queries and thereby providing more targeted content.

Recent research has focused on exploring different methodologies that can help predict future actions based on analyzing the user-system interactions from a search engine log. For example Zhang, Jansen, and Spink [5] use neural networks analysis as their methodology to identify factors that significantly affect the clickthrough of Web searchers. Their results show that high occurrences of query reformulation, lengthy searching duration, longer query length, and the higher ranking of prior clicked links correlate positively with future clickthrough. Time-based study of Web search logs has already been investigated by some researchers and has proven to be a viable approach. Özmutlu, Spink, and Özmutlu [18] conducted a comparative time-based study of US-based Excite and Norwegian-based Fast Web search logs and their findings suggest that Web user behavior fluctuates from the beginning of a day to the end of a day. Beitzel et al [16] examined query traffic on an hourly basis by matching it against list of queries that had been topically pre-categorized by human editors and

investigated changes in the query stream over time by examining the nature of changes in popularity of particular topical categories.

Evaluating predictive scenarios from search engine logs by adopting time series analysis as the preferred methodology has been studied by Zhang, Jansen, and Spink [19]. They perform a one-step-ahead prediction of the average rank with average query length as the input and found that searchers who typed the fewest query terms one period ahead were more likely to click higher ranked links. The work by Liu et al [20] presents a unified model to predict the Web query trend. They classify the queries into general, periodic, and accidental queries and attempt to unify the time series model, periodic model, and correlation model for different categories of queries respectively.

However, the above research works mainly focused on analysis of the general user-system interactions at the aggregate level (i.e., all the users in the data set). We found only a few studies which examine the individual user behavior. Piwowarski and Zaragoza [21] propose different models for predicting user clicks based on click-through history for a particular query. They combine the naïve baseline model with Bayesian probabilistic models to achieve high prediction accuracy over a high subset of query sessions. Dupret and Piwowarski [22] estimate the probability of examination of a result given the rank of the result and the distance (in ranks) to the last clicked result by developing a user browsing model. They find that a user usually

views the result that is located directly below a clicked result, which explain why documents located after a relevant document are clicked more frequently than others.

To our knowledge, none of the studies have used time series analysis as a method for developing forecasting models for individual user behavior. Even though the time series obtained from search logs can contain enormous amount of data and seem very random, filtering out the time series for the individual user can exhibit features like a pronounced seasonal pattern or a general trend in their search patterns. The unique user behavior can be modeled more extensively using click-through history of the concerned user by calculating additional attributes from the transaction log. Query modifications by the individual user can be predicted if the user's behavior can be described by a well-fit model. There are different ways to model this user behavior, and we aim to apply time series analysis to individual user searching data to develop predictive equations that define the individual's searching character.

In this research, we explore two of the best known approaches, the regression method and the Box-Jenkins [10] or AutoRegressive Integrated Moving Average (ARIMA) method to develop our equations. Each method has its own advantages in this research. We opted for the regression method as it could handle the technically discrete event (time ordered) series data of the individual user that is contained in the search engine transaction logs, while ARIMA is typically used for slightly longer

term predictions or predictions based on the analysis of the aggregate user activities [19].

Chapter 3

Methodology

3.1 Search Log

The dataset that we study is adapted from the query log of AOL search engine (www.aol.com). The entire collection consists of around 3.5 million query records. These records contain distinct queries submitted from about 65k users over three months (from March to May 2006). Each record is in the same format: {AnonID, Query, QueryTime, ItemRank, ClickURL}.

The descriptions for these elements are listed below:

- **AnonID**: An anonymous user ID number, usually corresponding to a real search engine user 2.
- **Query**: The query issued by the user, case shifted with most punctuation removed.
- **QueryTime**: The time at which the query was submitted to the search engine by the user for fulfilling his particular information needs.
- **ItemRank**: If the user clicked on a search result, the rank of the item on which they clicked is listed.
- **ClickURL**: If the user clicked on a search result, the domain portion of the URL in the clicked result is listed.

Below is a sample query log segment of an anonymous user in the AOL query log data.

AnonID	Query	QueryTime	ItemRank	ClickURL
36723	scientific notation	3/14/2006 9:27:02 PM	3	http://janus.astro.umd.edu
36723	scientific notation	3/14/2006 9:27:02 PM	1	http://www.nyu.edu
36723	scientific notation worksheet	3/14/2006 9:29:49 PM	1	http://www.fordhamprep.org
36723	converting metric prefixes	3/14/2006 9:51:49 PM	2	http://www.aaamath.com
36723	scientific notation	3/14/2006 10:12:15 PM		
36723	scientific notation	3/14/2006 10:16:42 PM		
36723	scientific notation worksheets	3/14/2006 10:16:57 PM	1	http://www.fordhamprep.org

Figure 1: Snapshot of the AOL search log

3.1.1 *Event Series*

By looking at the sample query log segment of an anonymous user's search history in the AOL search log, if the real-time spacing between consecutive records is discarded, then the search log can be viewed as an event stream of time ordered records. The "event" is the query searched by the user and the result URL clicked. Each event is identified by a discrete (integer valued) time index which gives us an event series ordered in time. The raw data in each record from this event series is the query term that was searched and the rank of the result that was clicked.

3.1.2 *Defining a Session*

One can define a user session on a Web search engine as a temporal series of interactions between the user and the search engine within a specific time period. During a session, the user may take several actions like searching for queries and clicking on URLs. In the context of our search log data, we define a user session as a

sequence of time ordered records grouped together per user which is localized in 'real' time. The real-time spacing between consecutive queries searched by a user is used to identify the start of a new session for each user. The QueryTime element is used to calculate the elapsed time between consecutive records for a user which will be used to determine the different searching sessions.

3.2 Research Question

The following research question is addressed in this thesis: *How can we apply the methodology of time series analysis to a search engine transaction log and use it to develop models that define an individual's searching patterns?*

We aim to identify the direct and indirect temporal factors that one can use to predict the searching patterns of an individual search engine user. In particular, we want to obtain a set of equations for the individual user that are characteristic of that user and use it to predict the user's future actions on the search engine. We provide a framework for using time series analysis in the study of Web search transaction logs. Our main driving force is the fact that Web browsing behavior differs from user to user, and a general browsing model for the entire set of users may be insufficient to predict an individual user's actions.

3.3 Overview of Time Series

The general description of a time series is a set of observations obtained by measuring a single variable regularly over a period of time. Examples of time series are the daily inventory levels measured for a period of time in a manufacturing industry or a series of average sales figures over many years that consist of one observation per month. There are two important points that differentiate a time series from other observational data methods. First, the typical time series comprises of observations taken at regular intervals of time. The other noticeable difference is that the observations in a time series are not mutually independent. A single event can potentially affect all the later observations in the series.

The search log data from AOL used in this research contains more than 3.5 million queries from 65,516 users over a period of three months. The foremost concern with the series in this search log is the fact that the observations are not at regular intervals of time. Rather, the time taken for the clicking of results is at the complete discretion of the user. For example, a user might issue a query and immediately click on a result, wait for sometime before clicking, or not click at all. Therefore, the second property of a typical time series comes into serious contention here. Every click by a typical user can be assumed to be dependent on his previous clicks during a searching episode according to the cascade model [23]. The same series can also be transformed to a typical time series by sampling the data at regular intervals of time. The time interval could be hourly, daily, weekly, or a custom-

defined time slot. However, sampling leads to loss of data; in this case it is the loss of flow of user behavior. Therefore, this standard approach is not feasible for our research.

Aris et al [24] discussed the challenges of event series and developed different methods of representing them. We use one of these methods, the Event Index method of representing the event series. The event occurrences are not scheduled beforehand and can take place at any time; in our case, the event is the click of a result. This method of representation distorts the x-axis (which typically represents time) and separates every event by an equal amount of space regardless of the elapsed time between events. The objective here is to represent the sequential user behavior taking place in a session of Web searching without the presence of physical time. The elapsed time between two queries could be as small as few seconds or as large as few days. Thus, it is impractical to include it in the representation of the concerned time series.

3.4 ARIMA vs. Time Series Regression

Our goal is to emphasize methods that are appropriate and useful for finding patterns that will lead to suitable models for our time series data. Time series analysis enables us to generate forecasts of a dependent time series that is based upon the information of its own history, explain events that happened in the past, and provide insight into the dynamical interrelationships between variables. There are a number of

methods to conduct time series analysis, but we explore two popular approaches, the regression method [25] and the ARIMA model. Box and Jenkins [10] developed the ARIMA model and defined three major stages for building a model: *identification*, *estimation*, and *diagnostic checking*. These three stages are essential for any statistical modeling. Identification involves selecting a tentative model type to work with the time series data. Estimation is the process of fitting the selected model to the data and estimating its parameters. Diagnosis is the stage in which the selected model is studied on how well it fits the data.

ARIMA models are classified as ARIMA(p,d,q) models where p is the number of non-seasonal autoregressive terms, d is the number of non-seasonal differenced terms, q is the number of non-seasonal moving average terms. For example, if a series of the user's rank clicks R(t) is modeled as ARIMA(1,1,0), which is a differenced first order autoregressive model, it would lead to the following equation:

$$R(t) - R(t\text{-}1) = \beta_0 + \beta_1(R(t\text{-}1) - R(t\text{-}2))$$

, where β_0 is the constant and β_1 is the differenced term coefficient.

The basic requirement for ARIMA modeling to work is that the sequences of data points are separated at regular intervals of time. This is necessary because ARIMA can handle the periodic cycles only if the time series data is spaced at uniform intervals. For example, a periodicity factor of 12 indicates that the time series under analysis consists of monthly data where there are 12 periods in a season. In our

situation, the user is under no obligation to conduct his browsing activities in a uniform manner. It is possible that we can add specific regressors to the ARIMA model as categorical variables to explicitly specify when there is a seasonal change, but the results obtained have not been satisfactory in practice [26]. Most statistical software packages also do not have support for adding categorical variables to an ARIMA time series model.

Therefore, we investigated the regression method, which has the advantage that the data need not necessarily be spaced out at uniform intervals of time because we explicitly define the predictor variables that are going to be used to estimate the dependent variable. For time series analysis using regression, the previous entries of the dependent variable are used as the predictor variables to find a formula that predicts the future entries in the series. This is the characteristic of an autoregressive model which is a special case of the general ARIMA model. The regression approach offers the flexibility that is not present by the ARIMA approach.

3.5 Regression method

In our case, suppose **R** is the rank of the URL clicked, and denotes the dependent variable. Consider possible ways of forecasting $\mathbf{r_i}$ from the previous points. One way is to simply use each point as the estimate of the next point. This would actually give the best possible prediction in a simple random walk. A different way would be to average the last 4 points before each point $\mathbf{r_i}$ and use that average as the estimate of $\mathbf{r_i}$.

In both the above methods, the prediction is a linear function of the data points preceding $\mathbf{r_i}$. Autoregression provides a way of examining an extremely broad class of linear functions and selecting the one that works best from one class [25].

For example, consider the same time series R(t) that represents a particular user's rank click-through history. The differenced first order autoregressive ARIMA(1,1,0) model can be modeled by regression as:

$$R(t) = \beta_0 + R(t\text{-}1) + + \beta_1 (R(t\text{-}1) - R(t\text{-}2))$$

When there are two time series of different variables that are related to each other, autoregression can be used to forecast one time series variable from the other. For example, the query length series may be used to predict the rank clicked by the user if an applicable model is able to be fitted to the data.

If Q(t) represents the time series of the query length (number of terms in the query searched), then it can be added to the time series regression equation as a regressor:

$$R(t) = \beta_0 + R(t\text{-}1) + + \beta_1 (R(t\text{-}1) - R(t\text{-}2)) + \alpha_0 Q(t)$$

If X is a categorical variable with two categories (for example, it logs whether the query was searched on a weekday or a weekend) which have an impact on the time series R(t), they can be coded as indicator variables X_1 and X_2 and included in the time series regression using the leave-one-out procedure:

$$R(t) = \beta_0 + R(t\text{-}1) + + \beta_1 (R(t\text{-}1) - R(t\text{-}2)) + \alpha_0 Q(t) + \alpha_1 X_1$$

The approach to solve the above equation is similar to the problem of multiple linear regression, which characterizes the relationship between independent and dependent factors of a system. The problem is to fit a model of the following form to the available data, which characterizes a hyper plane in a k-dimensional space [2], [27].

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots + \beta_k x_k + \varepsilon$$

, where there are k independent factors, β_i , i=1,.....k is the coefficient of the i^{th} independent factor and β_0 is a constant value.

The coefficients of the regression equation are determined using the least squares method. The objective is to minimize the squared error that occurs between the fitted equation and the actual data.

The coefficients of the regression equation are calculated using the following equations and matrices [27]. Consider the matrices shown in Figure 1, where **y** is a vector of the response (or values of dependent factors obtained as a result of experiments), **X** is a matrix of the values of the independent factors, x_{ij} is the value of the i^{th} independent factor, i=1,…k, at the j^{th} experiment or data point, j=1,…n, β is the vector of the regression coefficients and **ε** is the error vector.

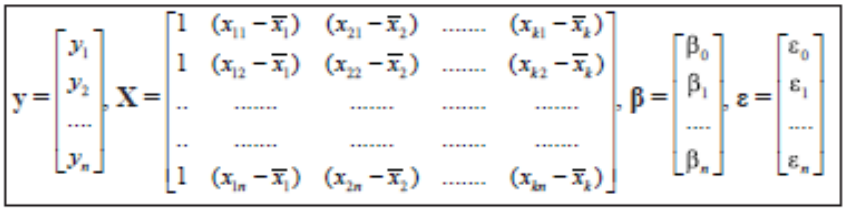


Figure 2: Regression Matrices

In this case, the least squares estimator for the regression coefficients is [27]:

$$\boldsymbol{\beta} = (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

, where $\mathbf{X}^{\mathbf{T}}$ is the transpose of matrix **X**.

The analysis of variance indicates whether the developed regression equation effectively explains the dependent factor, as well as which independent factor has a statistically significant effect on the dependent factor.

The regression method for time series is often good for shorter term forecasts and one-step-ahead predictions and is flexible in usage. There are different statistical software packages to conduct time series analysis like SAS, SPSS, StatGraphics, and Minitab. We used both Minitab and SPSS for conducting our experiments.

Chapter 4

Research Design and Data Analysis

4.1 Research Design

4.1.1 *Data Collection*

The log used in our research is an AOL search transaction log collected from 01 March 2006 to 31 May 2006. The transaction log contains 3,558,390 records of user search activity for 65,516 distinct users. We imported the transaction log into a relational database for initial pre-processing and cleaning [4]. Table 1 shows the fields included in this log.

Table 1: *Fields in AOL transaction log*

Field	Description
User Identification	An anonymous user ID number.
Query	The query issued by the user, case shifted with most punctuation removed.
QueryTime	The time at which the query was submitted for search.
Item Rank	If the user clicked on a search result, the rank of the on which they clicked is listed.
Click URL	If the user clicked on a search result, the domain portion of the URL in the clicked result is listed.

We also calculated additional attributes for the log after importing to the relational database. The additional fields are shown in Table 2.

Table 2: *Additional calculated fields in the AOL transaction log*

Field	Description
Query Identification	A unique id for every record (query) issued in the transaction log and is the primary key.
User Intent	User Intent can be classified as *Informational, Navigational, or Transactional* that reflects the user's desired intent.
Query Identification	A unique id for every distinct query searched in the transaction log.
URL Identification	A unique id for every distinct URL clicked in the transaction log.
Query length	The number of terms contained in a particular query.
Character length	The number of characters contained in a particular query.
Reformulation pattern	There are seven categories of query reformulation [28]

The User Intent field was calculated using an algorithm developed by Jansen et al. [17] that automatically classifies queries into *informational, navigational,* or *transactional*. The algorithm was originally used in datasets which identified the users using IP addresses and cookies. We adapted it and implemented it to our dataset to classify the searcher queries.

Assumptions:

1. Transaction log is sorted by user ID and time (ascending order by time).

2. Search engine result page requested are removed.

3. Null queries are removed.

4. Queries are primarily English terms.

Input:

Record R_i with User ID U_i, query Q_i, and query length QL_i.

Record R_{i+1} with User ID U_{i+1}, query Qi+1, and query length QL_{i+1}.

I: conditions of information query characteristics

N: conditions of navigational query characteristics

T: conditions of transactional query characteristics

Variable: B: Boolean//(if query matches conditions, ‘yes’ else ‘no’)

Output: Classification of User Intent, C

begin

Move to R_i (this module establishes the initial boundary condition)

Store values for U_i, Q_i, and QL_i

Compare (U_i, Q_i, and QL_i) to N

    If B then C = N

Elseif Compare (U_i, Q_i, and QL_i) to T

    If B then C = T

Elseif Compare (U_i, Q_i, and QL_i) to I

    If B then C = I

While not end of file

Move to R_{i+1}

Compare (U_i, Q_i, and QL_i) to N

    If B then C = N

Elseif Compare (U_i, Q_i, and QL_i) to T

    If B then C = T

Elseif Compare (U_i, Q_i, and QL_i) to I

    If B then C = I

(R_{i+1} now becomes R_i)

Store values for R_{i+1} as U_i, Q_i, and QL_i

end loop

The query search pattern was calculated using another algorithm developed by Jansen et al [28], [29], which automatically classified queries into new, duplicate, reformulation, generalization, specialization, generalization with reformulation, or specialization with reformulation categories.

The terminology that we use in this research is similar to that used in other Web transaction log studies [30].

- *Term*: a series of characters within a query separated by white space or other separator.
- *Query*: string of terms submitted by a searcher in a given instance of interaction with the search engine.
- *Query length*: the number of terms in the query (which includes the traditional stop words).
- *Session*: a series of interactions submitted by a user during one interaction with the search engine.

4.1.2 *Temporal factors*

To begin the time series analysis, we calculated important temporal factors (shown in Table 3) after importing the transaction log into SPSS.

Table 3: *Temporal factors*

Factor	Description
Hour	The hour of day in which the query was issued.
Day	The day in which the query was issued with March 1st as day 1.
Time Slot	One of the four time slots during which the query was searched, with the time slots roughly corresponding to *morning*, *afternoon*, *evening*, and *night* periods.
Elapsed Time	The time between two queries searched by the same user.
Session	Identifies the episodes of interaction between the user and the Web search engine.

The reason for choosing four time slots was the typical user activity that fell into one of the time slots in a periodic manner. Using one full day as a time slot resulted in loss of information about user browsing activities during different times of the day. However, an hour was too short to be used as a time slot since there was not enough data contained in an hour of searching activities per user to conduct the analysis. The elapsed time was calculated to determine the different browsing sessions for each of the users. We combined both method 2 (time based) and method 3 (query-content based) for detection of the session boundaries as described in [15]. If there is a 30 minute gap between two queries for the same user, and the second query is classified as a new query, then it constitutes a new session starting with the second query. If the time period between interactions exceed 30 minutes but the subsequent query is related to the previous query, then it does not qualify as a new session.

4.1.3 *Cycles and Coded Variables*

The time series in this transaction log typically followed a daily cycle and a weekly cycle. Usual time series programs using the ARIMA can handle only one

cycle at a time, but a coded variable approach using regression can handle several at once. The daily cycle was tracked by the four time slots of six hours each, which captured the approximate user behavior in the mornings, afternoons, evenings, and nights. The weekly cycle is tracked by coding the day of the week from one to seven. Additionally, categorical variables like the user intent and search pattern were incorporated into the initial regression model by using indicator variables. We use the "leave one out" method to avoid the difficulty arising from linear dependency which would have made it hard to estimate the individual coefficients [31].

4.2 Data Analysis

Modeling the behavior of the individual users turned out to be somewhat challenging because we both wanted to include as many users as possible and needed enough information on individual users to do the modeling. We explain our approach for identifying an appropriate set of users, since it is essential to understanding our research findings. First, we needed to find the *activeness* of the users, defined as the number of episodes of searching over the three-month period. The histogram in Figure 2 shows the percentage of users who searched a given number of queries over the period of three months.

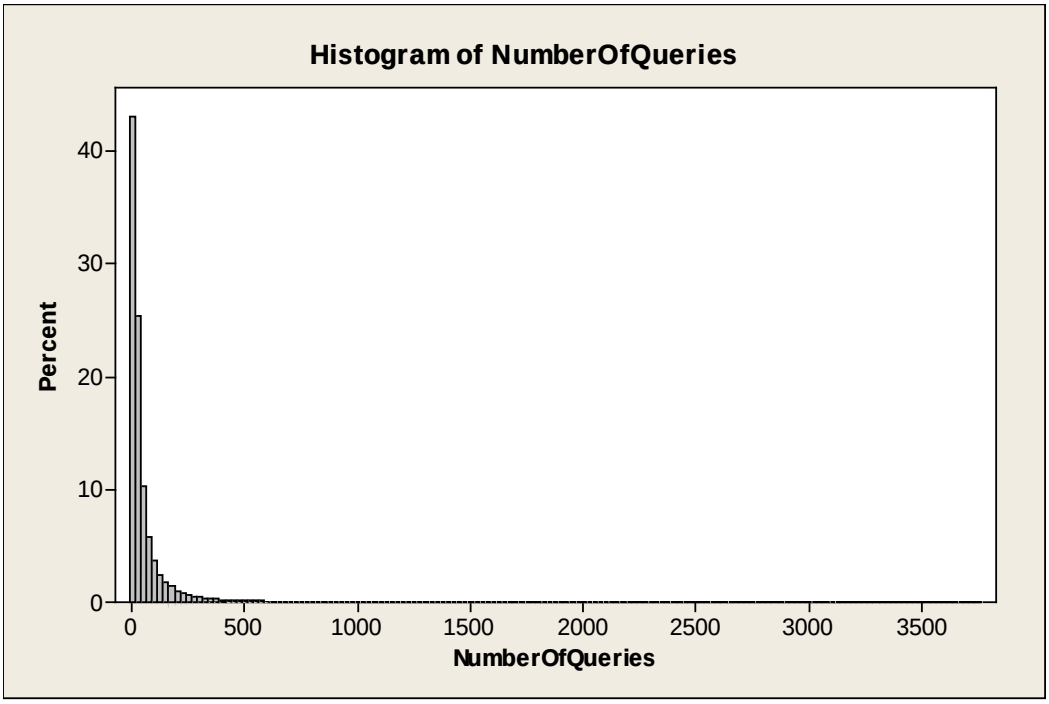


Figure 3: Number of Searchers by Number of Queries

Figure 3 shows that more than 40 percent of users searched just 1 to 10 queries over the three-month period, and an additional 25 percent of users searched between 10 and 39 queries in the entire span of three months. This implies that approximately 68 percent of users have searched for less than 40 queries individually. At first glance, it looks like there are not enough data points to fit a time series regressive model. The plot in Figure 4 gives a better idea of the user activity in this transaction log.

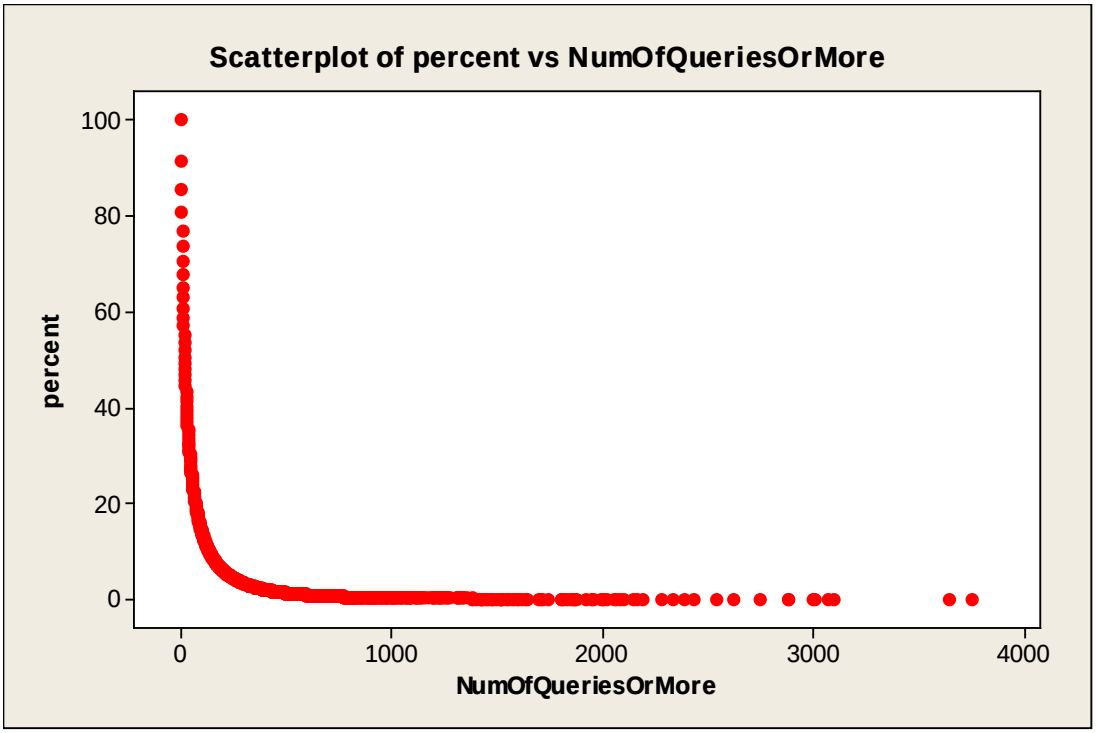


Figure 4: Graph of Number of Searchers By Number of Queries

The plot describes the percentage of users who have submitted at least the given number of queries specified in the x-axis. For example, 100 percent of users had issued at least 1 query, 91 percent had issued 2 or more queries, 85 percent had searched for 3 or more queries, and so on. The number of users show a rapid exponential decay as the number of queries issued increases.

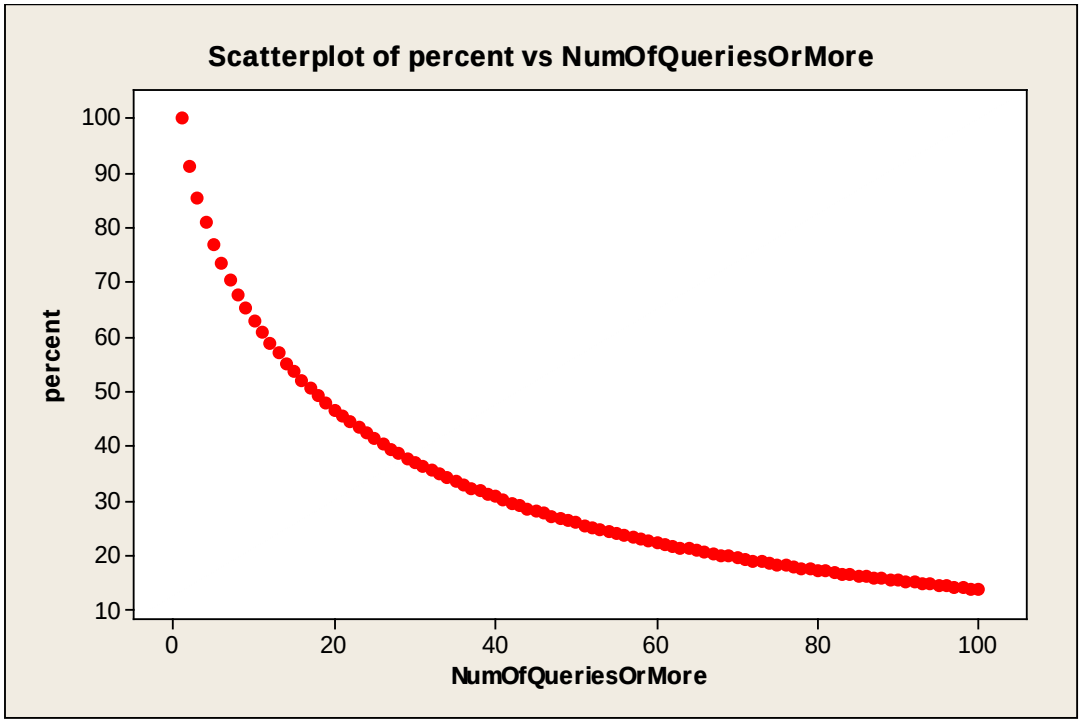


Figure 5: From 1 to 100 or More Queries by Percent of Users

If we restrict the plot to display the activity distribution (see Figure 5) for users who have searched for up to 100 or more queries, we can see that only approximately 26 percent of users have issued 50 or more queries, and 13 percent of users have searched for 100 or more queries.

However, we had to determine the distribution of queries issued by a user in different sessions. For example, a user may have searched 50 queries in a single session, or the user may have searched for a single query in 50 different sessions. The user's click-through behavior could potentially vary depending on whether the issued query was in the same session or implied the start of a new session.

We did a basic analysis test to find out the number of queries that were required to have been searched by a user to be able to model is characteristic searching behavior. We performed a time series regression for predicting the rank clicked by a user using the user's previous clicks up to 3 periods behind and new session identifier as the independent variables. Users who had searched for 500 or more queries had a mean adjusted R^2 = 33 percent; users who searched for 100 or more queries had a mean adjusted R^2 = 21 percent; users who searched for 50 or more queries had a mean adjusted R^2 = 13 percent. The adjusted value of R^2 is one of the methods of cross-validation of the model. While R^2 indicates how much of the variance in the dependent variable is explained by the regression model from our sample data, the adjusted R^2 indicates how much variance in the dependent variable is accounted for if the model had been derived from the population from which the model sample was taken [32]. Given the fact that the independent variables chosen were not the optimum choices for all the users (i.e. we hoped to improve on the results obtained with more number of variables in conjunction with a selection method instead of using the same set of variables for every user), we made the cut-off at 50 queries or more. This cut-off signifies that even though we are going to be analyzing only 26 percent of the users, they have constituted 80.9 percent of the total queries searched in the time period.

4.2.1 *Choosing the independent variables*

The fundamental objective of applying autoregression to a time series is to fit an equation to a set of independent variables that is able to forecast each point accurately from the previous points. The time series analysis will have significant meaning only if we can find the relationship between different fields of data. Table 4 lists the complete set of attributes that we investigated as significant predictor variables for predicting the next rank clicked for the individual users.

Table 4: *Predictor variables*

Predictor Variable	**Description**
Ranklag1	The rank of the result clicked one step behind
Diff_Ranklag1	The difference in time series of Ranklag1 and is equivalent of distance (in ranks) to rank clicked one step behind.
Diff_avgranklag2and3	The difference between two consecutive queries of the average of ranks clicked two and three steps behind.
Diff_avgranklag4and5	The difference between two consecutive queries of the average of ranks clicked four and five steps behind.
Qlength	The number of terms in the query.
Informational	The query is an informational query.
Navigational	The query is a navigational query.
New	The query is classified as a new query.
Duplicate	The query is classified as a duplicate query.
Reformulation	The query is classified as a reformulated query.
Generalization	The query is classified as a generalized query.
Specialization	The query is classified as a specialized query.
Gen_with_reform	The query has been reformulated to a generalized query.
New_session	The query is the start of a new session for the user.
Weekend	The query is searched during the weekend.
Night	The query is searched during the night (00:00 to 05:59).
Morning	The query is searched during the morning (06:00 to 11:59).
Afternoon	The query is searched during the afternoon (12:00 to 17:59).
Short_elapsed_time	If the elapsed time between consecutive queries is between 1 to 59 seconds.
Moderate_elapsed_time	If the elapsed time between consecutive queries is between 1 to 5 minutes.

Since it is the individual user we are concerned about, the time series for ItemRank clicked for each user should be stationary (i.e., their statistical properties like mean and variance should be constant over time to be able to predict with confidence the ItemRank that would have been clicked). This is because we can simply predict that the statistical properties will be the same in the future as they have been in the past for a stationary time series. There are different ways to stationarize a time series – by differencing, logging, deflating, and so on. For our time series, the first difference of the time series of *ItemRank* is able to render it approximately stationary.

The first difference of a time series is the series of changes from one period to the next. In our case, if R(t) represents the rank clicked at time period t, and R(t-1) is the rank clicked at time period t-1, then R(t) – R(t-1) is the first difference. Since we are going to autoregress the time series on lagged values of the rank clicked, we would stationarize ItemRank and lagged values of ItemRank by differencing, and then use the lagged values of the stationarized variables to fit a model.

According to [25], an *exponentially weighted forecast* (EWF) forecasts each point from a weighted average of previous points in which earlier points get less weight than later points because they are further from the target point. We could approximate EWF by lowering the number of autoregressive terms used in the model by averaging adjacent terms with minimal loss of predictive power. From our initial analysis for a

small random subset of users, we found that the average of lags of rank 2 and 3 periods and the average of lags of ranks 4 and 5 periods before performed as well as it would have by including the individual lags. The lags of period 6 and greater were not found to be significant factors.

After differencing to stationarize our series and using average ranks, our autoregression equation is:

$$Diff(ItemRank) = Diff\_Ranklag1 + Diff\_avgranklag2and3 + Diff\_avgranklag4and5 + \text{additional variables}$$

where $Diff(ItemRank) = ItemRank - Ranklag1$

This implies that

$$ItemRank = Ranklag1 + Diff\_Ranklag1 + Diff\_avgranklag2and3 + Diff\_avgranklag4and5 + \text{additional variables}$$

It is interesting to note that the difference in the lagged values of the rank is equivalent to distance (in ranks) between two ranks clicked in two successive periods. The additional variables with the exception of *query length* are categorical variables with binary values of 1 if the event has occurred and 0 if it has not occurred. We use the leave-one-out method as explained earlier which elucidates the fact that every category has one instance of its conditions missing from the predictor variables list.

4.2.2 *Choosing the number of regressors*

Once we decided on the requirement that a user should have searched for a minimum of 50 or more queries, we had to choose the number of regressors (i.e., predictor variables) that would be used to perform the time series regression. Even though it is widely believed that N/P > 10, where N is the sample size and P is the number of predictors, the standard errors of regression slopes are determined more by N-P rather than by N/P [25]. For example, if a user has searched for 50 queries, then N = 50 and if we choose 10 predictor variables, we have N/P = 5 and N-P = 40. If P = 5, then N/P = 10 and N-P = 45. The difference between 40 and 45 is not huge and many simulations have been shown to verify that an adequate N-P value is satisfactory even if it does not satisfy the "ten times" rule [25].

For predicting the rank of the results to be clicked one-step ahead, we have a total of 19 possible predictors to choose from. *Stepwise regression* is a process of adding or removing the variables based on the *t*-statistics of their estimated coefficients. SPSS has a semi-automated feature to implement stepwise regression. The basic direction of the steps is to add statistically significant variables to the model, but if any of the variables becomes non-significant, it is removed from the equation. In a forward step, the best available variable is added only if its associated *p*-value is less than the specified α-level, which is 0.05 in our case. Thus, we use stepwise regression in choosing the number of regressors for each individual user because every user has

a unique searching behavior and a set of predictors which might fit a model for one user might not fit a model for another user.

4.2.3 *Sample Analysis of a user*

We decided to perform the analysis for all users who were determined active by using a cut-off of minimum of 50 queries. There were a total of 17,066 users out of the 65,516 users who were active users and constituted 80.9 percent of the total queries searched during the three month period. Since we are interested in fitting a time series regression model for each of those users using their log history, it will lead to a maximum of 17,066 equations for predicting the rank that will be clicked one-step ahead. Since it is not possible to realistically explicitly specify each of those models, we present a sample analysis for one of those users in this section and then present clustered statistical results and aggregate model significance in the results section.

The users in this analysis searched for a minimum of 50 queries and a maximum of 3,755 queries over the three-month period. In this sample analysis, our one user is identified with a unique user identification number, identification number 30,011, and the transaction log history shows that the user submitted a total of 1,422 queries in 306 distinct sessions from 1 March to 31 May 2006.

Using the Event Index method of representing unevenly spaced time series, Figure 6 represents the time series of the ItemRank clicked by this user. The value 0 indicates that the user did not click on any result for that particular query. We can see that the rank clicks have a distinctive trend where the ranks clicked seem to numerically follow their predecessors. We can also see the presence of few outliers, indicating that the time series may not be stationary.

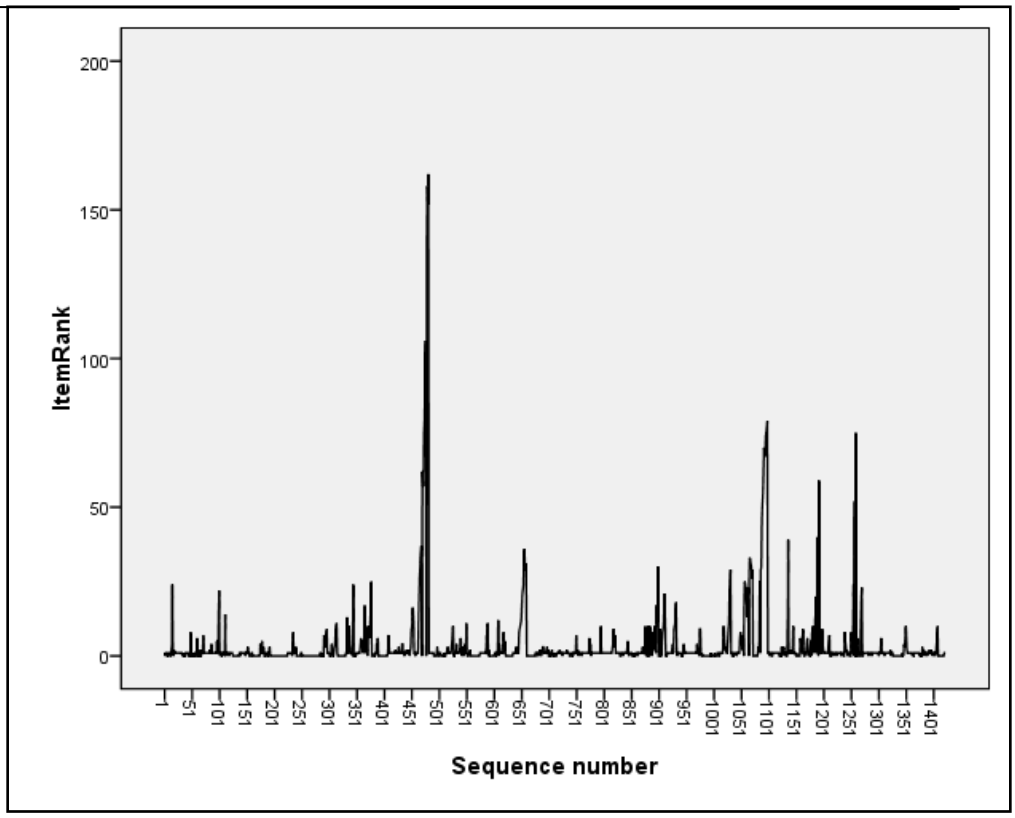


Figure 6: Time series plot of ItemRank clicked by user number 30,011

To verify this, we plot the autocorrelation for ItemRank as shown in Figure 7.

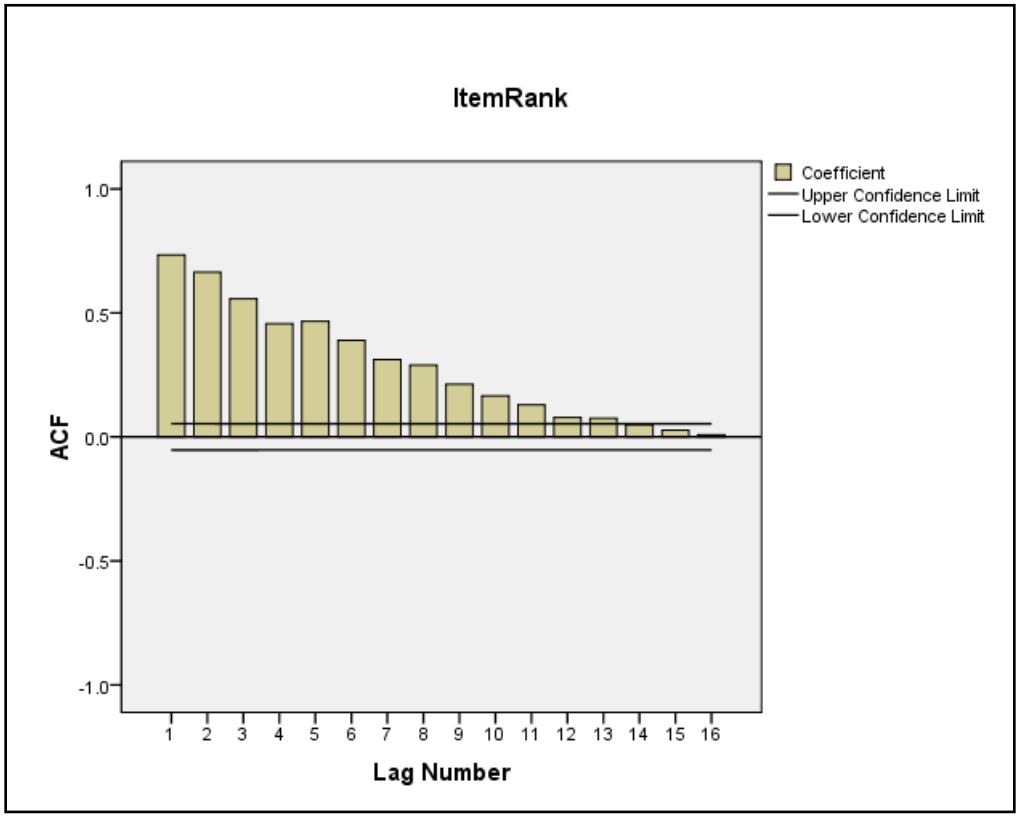


Figure 7: Autocorrelation of ItemRank

The autocorrelation function (ACF) shows a very slow, approximately linear decay which is typical of a nonstationary time series. The time series needs at least one differencing to stationarize the series. Figure 8 shows the stationarized time series of ItemRank.

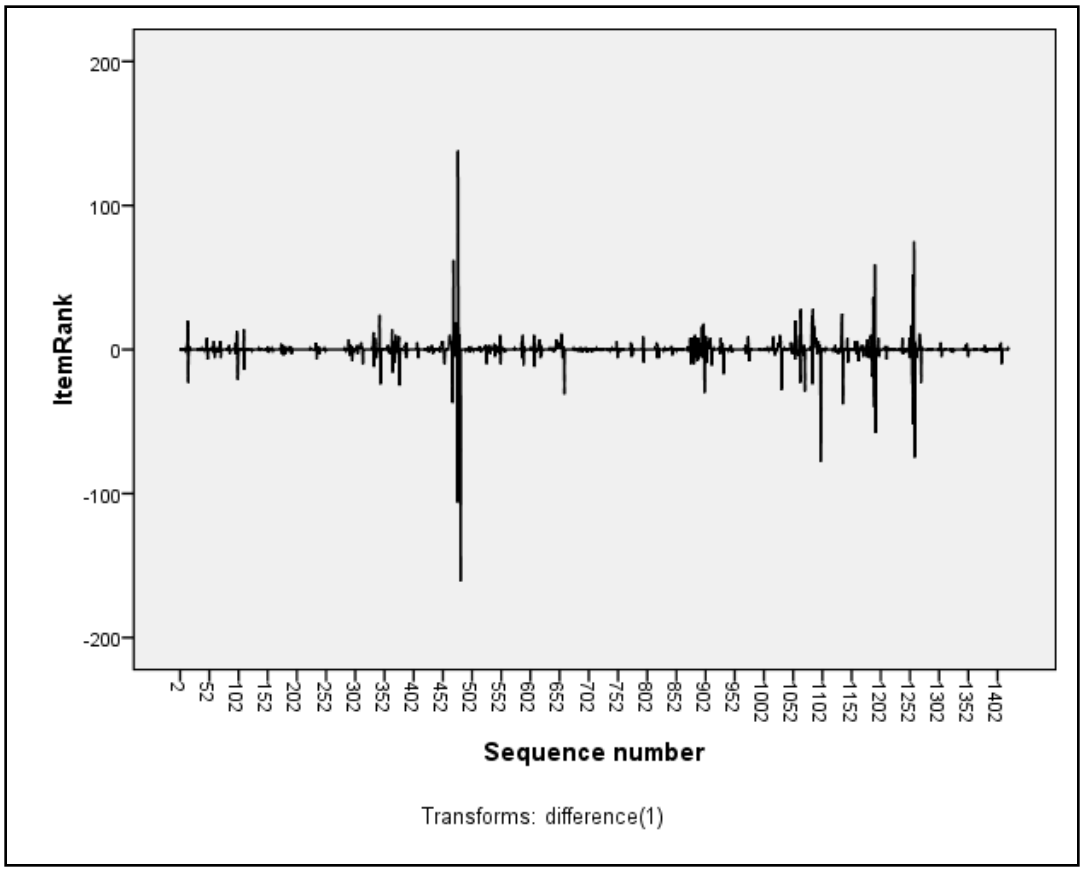


Figure 8: Differenced Time Series

The series now seems to be approximately stationary and shows a tendency to come back to its mean without any long-term trend, although there might still be a few outliers as indicated by the presence of a few spikes of non-constant variance. Figure 9 shows the ACF of the differenced series, and the signs of differencing are evident by the alternating signs from one entry to the next and also by the negative correlation at lag 1.

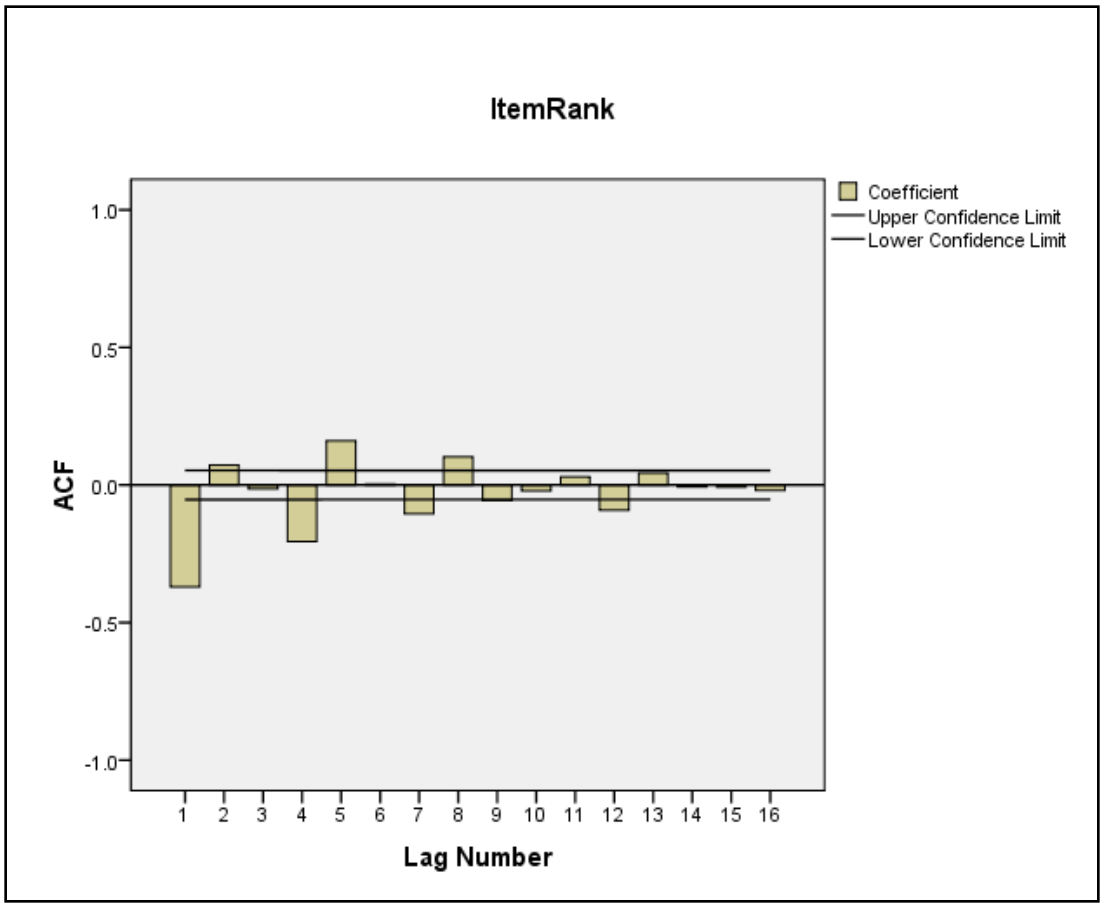


Figure 9: Autocorrelation of Differenced ItemRank

The changing signs from one observation to the next could imply that the series may have been over-differenced, but it could also be due to presence of outliers that have not been removed. As stated earlier, from initial analysis of a subset of random users, the first difference of the user's ItemRank time series sufficed to render it approximately stationary. We want to regress the stationarized ItemRank on lags of itself. Looking at the ACF for Diff(ItemRank), we see significant spikes till lag 5 and small spikes at lags 7, 8 and 12 that may be due to outliers. We use SPSS to model the time series regression equation for the differenced time series using the stepwise method of choosing the individual variables and obtained the following model:

$$ItemRank = - .709 + RankLag1 - .290\ Diff\_ranklag1 - .121 Diff\_avgrank4and5 + 1.242 moderate\_elapsed\_time + 2.974 duplicate + Error$$

The model has an adjusted R^2 value of 0.587 which implies that the model is roughly able to explain about 58 percent of variance seen in ItemRank. The *F*-statistic = 403.276 with *p*-value of 0.000 indicating the statistical significance. But the standard error of the estimate is found to be 8.100, which is on the higher side. It might be due to presence of outliers.

After removing the outliers and differencing, Figure 10 shows the autocorrelation function for the differenced ItemRank.

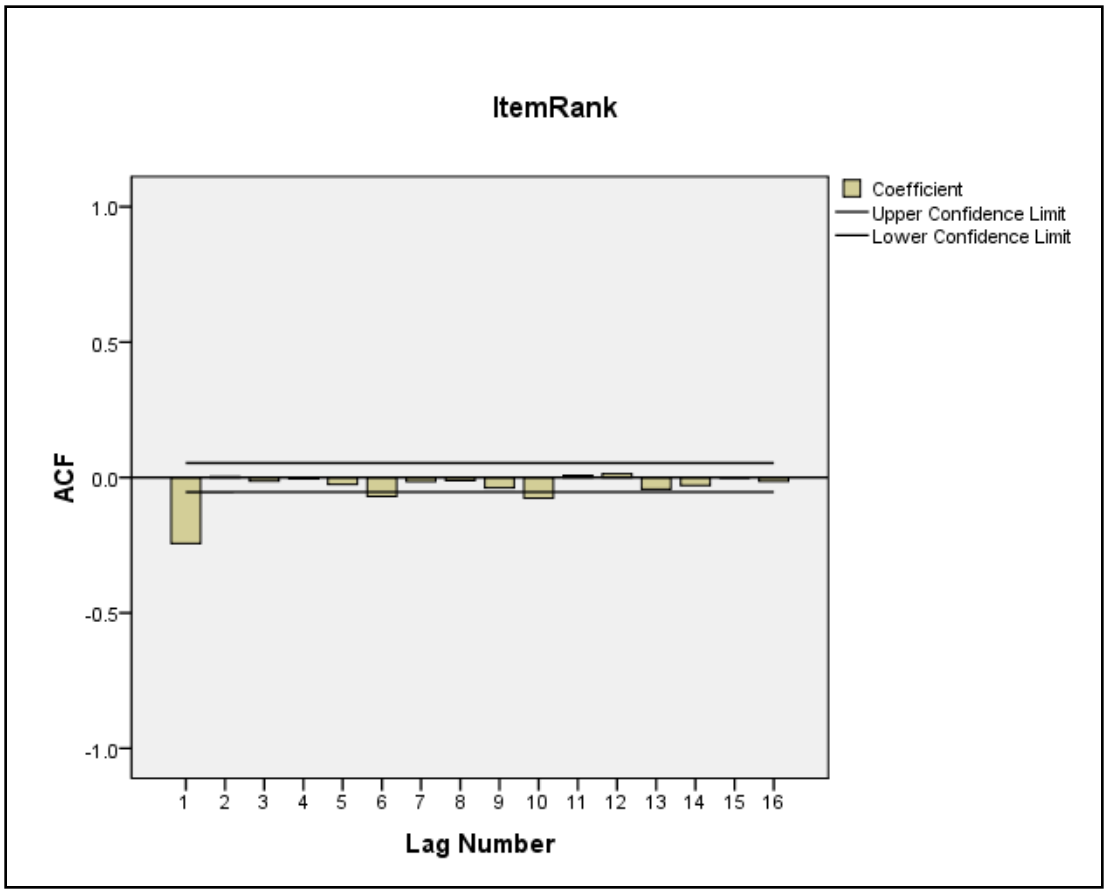


Figure 10: Autocorrelation of differenced ItemRank after removing outliers

The ACF has a negative spike on lag 1 and then shows little correlation at higher lags indicating that the differenced series is now stationary. Using the same stepwise procedure to perform the time series regression, we obtain the following model:

ItemRank = .899 + RankLag1 -.135 Diff_ranklag1 -1.130 new - .766 navigational + Error

with Adj.R^2 = 0.790; Std. Error of the estimate = 3.414; *F*-statistic = 1302.772 with a *p*-value of 0.00.

At first glance, we can immediately notice that removing the outliers caused the adjusted R^2 to jump from 0.587 to 0.790, which implies that approximately 79 percent

of the variance seen in ItemRank is explained by the independent variables identified using stepwise regression. The ANOVA table indicates that the *F*-statistic is statistically significant which implies that the variation explained by the time series regressive model is not due to chance. The standard error of the estimate has now dropped from 8.100 to 3.414. Having a low standard error is essential to be able to make a good prediction with tighter confidence intervals. The number of significant independent variables has changed from 5 to 4. Thus, after removing the outliers, the final time series model for this particular user is much more accurate than the previous model.

Looking at the predictor variables that have been fit to this model, we can see that other than its dependence on its own history, it includes two indicator variables – *new* with a coefficient of -1.130, and *navigational* with a coefficient of -.766. This tells us that this particular user tends to click first at the top-ranked results when searching for a new query. If the query is found to have a navigational intent, i.e. if the user has issued a query that is a navigational query, it again leads to a reduction in the value of the rank of his click-through.

There are certain diagnostic tests that we must perform to be confident about our time series regression model. For maximum confidence in this forecast, we need to plot the forecasting errors against time to check if the past success of the prediction equation was uniform across time. A slowly undulating time series plot (long

sequences of residuals on the same side of zero) indicates a correlation between lagged residuals (e_t and e_{t-1}). Figure 11 shows the time series plot of the standardized residuals and there does not appear to be correlation between the lagged residuals.

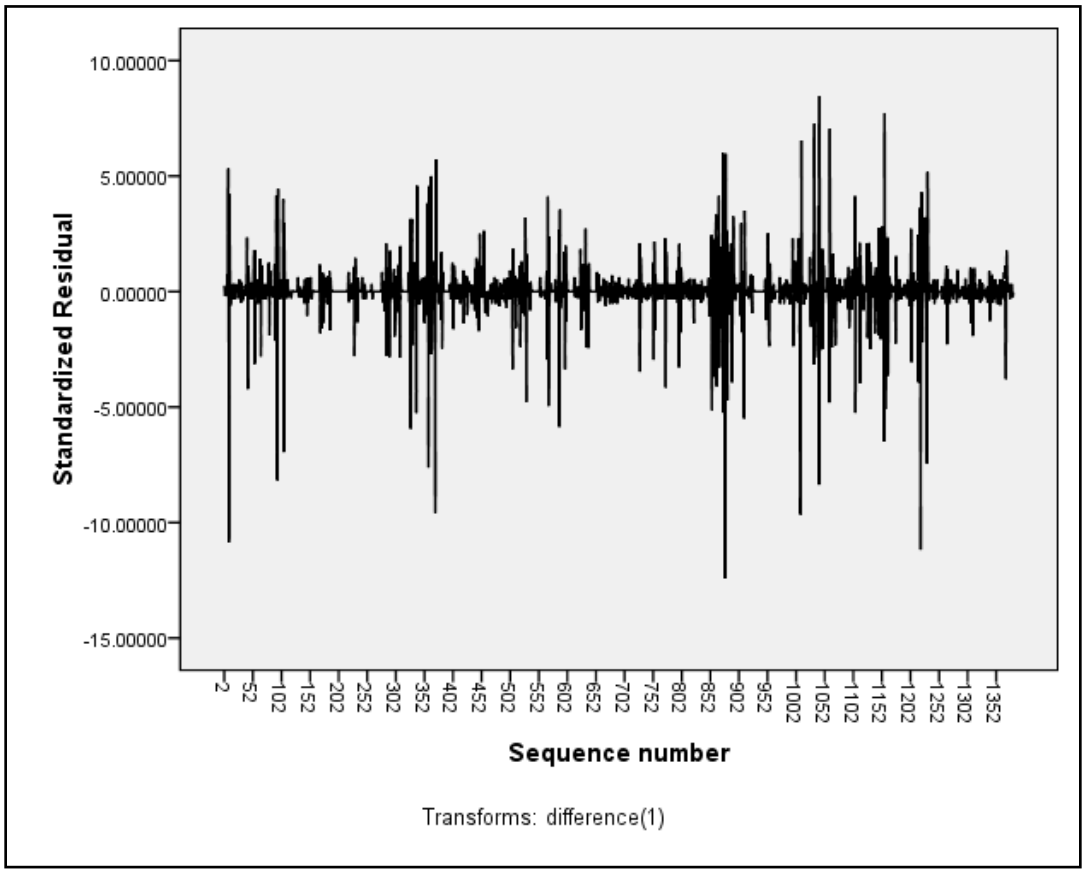


Figure 11: Time Series Plot of the Standardized Residuals

To confirm this, we plot e_t versus e_{t-1} as shown in Figure 12. A linear pattern would indicate correlation, but we obtained a random pattern indicating that there is no correlation among the residuals. This is important because serial correlation in the residuals suggests that there is room for improvement in the model, and extreme serial correlation is often a symptom of a badly mis-specified model.

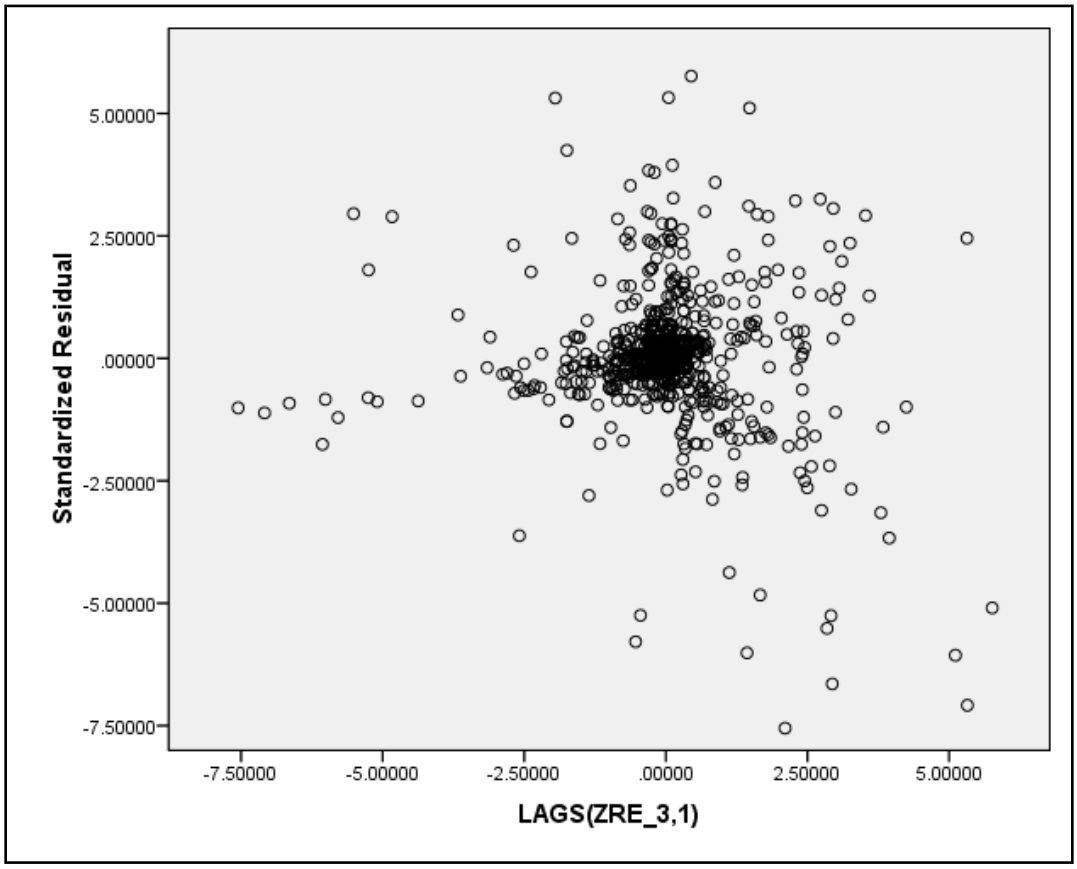


Figure 12: Scatter Plot between the Standardized Residual (e_t) and Lag of Itself (e_{t-1})

Figure 13 shows a plot of residuals versus the predicted values, and it has a very mild sideways cone pattern which indicates that there is some non-constant variance. The principal consequences of non-constant variance are the prediction intervals for individual y values which may be wrong because they are determined assuming constant variance. There is a small effect on the validity of t-test and F-test results, but generally regression inferences are robust with regard to the variance issue. Hence, this does not hurt our regression estimates much.

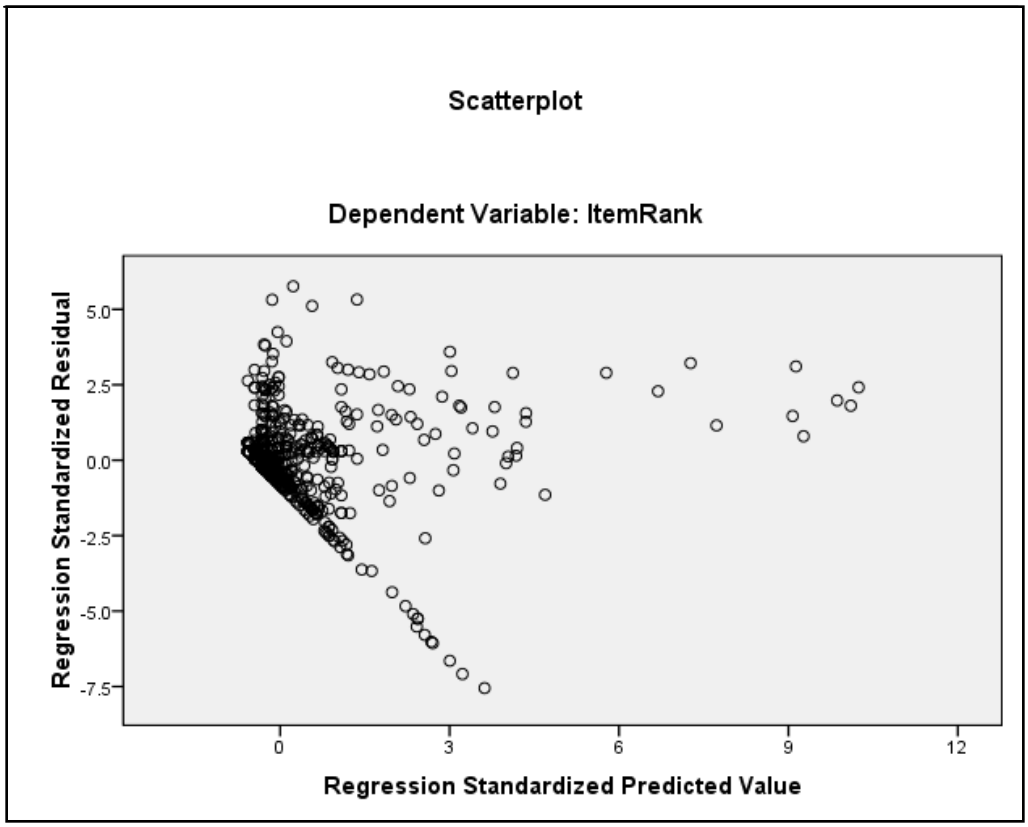


Figure 13: Residuals versus Fits

It is important to follow the steps listed in this sample analysis because a flawed time series analysis could lead to spurious results. To summarize, the following criteria must be satisfied in order to have maximum confidence in our prediction.

- It is imperative to transform both the dependent and independent variables to a stationary form in order to have a balanced equation. In our case, we found that taking the first difference of the *ItemRank* and its lagged values renders the time series roughly stationary.

- The time series regression model has to be fitted with the right dynamics in order to get an accurate prediction. Our preference is to build the autoregressive part first and then add the dynamics of several interesting indicator variables later using the method of stepwise regression to obtain the best fitting model.
- Our model can be considered dynamically correct when the residuals of our regression model are uncorrelated. If the above two steps have been followed, then most often than not, the model is found to contain the right dynamics. Removing the outliers improves the overall model. We can check for correlation in the errors by using the steps listed in the sample analysis or we can use the *Durbin-Watson* statistic to check for significant residual autocorrelation at lag 1.

We performed the time series analysis for all 17,066 active users and obtained unique time series models for all of them. Viewing the aggregate results of all the users and analyzing them can help in identifying certain similarities among users and relative significance of the independent variables.

Chapter 5

Results and Discussion

5.1 Aggregate Model Analysis

We used SPSS to do the time series analysis of the entire set of active users. We performed the analysis for 17,066 users who had collectively searched for approximately 2.8 million queries over the three months period.

Table 5 shows the aggregate adjusted R^2 values of the models fitted for all the users in the analysis. As mentioned before, while R^2 indicates how much of the variance in the dependent variable is explained by the regression model from our sample data, the adjusted R^2 indicates how much variance in the dependent variable is accounted for if the model had been derived from the population from which the model sample was taken and is used as a method of cross-validation of the time series regression model. The adjusted R^2 values ranged from as low as .047 to as high as 1.000 and the average value was 0.574. The percentile distribution of the adjusted R^2 values for the users is also displayed in the table and a median of 0.57 implies that 50 percent of the users were fitted by models which could explain at least 57 percent of the variations observed in their rank clicks.

Table 5: *Aggregate Adjusted R^2 Values according to percentile distribution of users*

N	Valid	17066
	Missing	0
Mean		0.57426
Median		0.57158
Minimum		0.0470
Maximum		1.000
Percentiles of Active Users	10	0.34600
	20	0.43602
	25	0.46690
	30	0.49337
	40	0.53791
	50	0.57158
	60	0.60907
	70	0.65622
	75	0.68066
	80	0.70940
	90	0.80241

Table 6 shows the aggregate standard error of the estimate values of the models fitted for all the users in the analysis. The "standard error of the estimate" in a regression model is the root-mean-squared error of the residuals, adjusted for the number of the coefficients estimated and is used to calculate the confidence intervals for the predicted values. Larger values of the standard error lead to wider confidence intervals. Hence, it is desirable for the model to have a small value of the standard error of the estimate. The values ranged from 0.00 (no standard error) to as large as 46.66, although the average value was 2.582. The percentile distribution of the standard errors of estimates for the users is also displayed in the Table 6 and a median of 1.86 implies that 50 percent of the users were fitted by models that had a standard error of less than 1.86. It corresponds to stronger prediction confidence of the users' rank clicks.

Table 6: *Aggregate standard error of the estimate values according to percentile distribution of users*

N	Valid	17066
	Missing	0
Mean		2.58227
Median		1.85700
Minimum		0.000
Maximum		46.659
Percentiles of Active Users	10	0.62505
	20	0.94322
	25	1.10757
	30	1.24787
	40	1.51623
	50	1.85700
	60	2.17266
	70	2.62421
	75	2.92505
	80	3.36850
	90	5.01366

Table 7 shows the aggregate number of independent (predictor) variables in the final model for the active users in the analysis. The numbers range from a minimum of 1 predictor to a maximum of 17 predictor variables in a single model for a user. The average was found to be 4.13, and the median of 4 predictors indicate that 50 percent of the users were fitted by models with less than 4 independent variables and 50 percent had more than 4 independent variables in their models.

Table 7: *Aggregate number of predictor variables according to percentile distribution of users*

N	Valid	17,066
	Missing	0.0
Mean		4.13
Median		4.00
Minimum		1.0
Maximum		17.0
Percentiles of Active Users	10	2.00
	20	2.00
	25	3.00
	30	3.00
	40	3.00
	50	4.00
	60	4.00
	70	5.00
	75	5.00
	80	5.00
	90	7.00

There is no clear-cut rule as to how many predictor variables are ideal in a model. If there is no appreciable increase in the adjusted R^2 value by adding an independent variable, then one could possibly remove it from the model. In our case, most of our independent variables are categorical variables, which are used as indicator variables for a specific event occurrence during that specific period of Web search. It is beneficial to include them if they are shown to be significant as it helps to explain the user's unique search behavior.

Table 8 gives the average values for the above statistics according to the user activeness (number of queries searched by the user) in the three-month period.

Table 8: *Aggregate Statistics by User Activeness*

Number of Queries searched by the user	Mean Adjusted R^2	Mean standard error of the estimate	Mean number of Predictor variables
50 – 100	0.49856	2.61324	3.89
100 – 500	0.63447	2.53783	4.02
500 – 1000	0.72458	2.59245	4.82
1000 and above	0.79206	2.26756	5.20

The average adjusted R^2 value increased as the number of queries searched by the user increased. This suggests that as we keep building up the history of a user, we are able to develop better models for them. The mean standard error of the estimate fluctuated around the mean of 2.58 for the aggregate set of users, but we found that the most active users who searched for more than 1,000 queries had a value 2.26, which is slightly less than the average. This again indicates that having a larger user history results in finding a more accurate time series model. Finally, the average number of predictor variables in the final model increased as the user activeness increased.

5.2 Aggregate Predictor Variables Analysis

We identified a total of 20 factors that could be used as potential predictors for building the time series regression models for the active users. We found all 20 to be significant predictors in different combinations depending on the characteristics of the user for predicting the rank of the results clicked. Table 9 contains the number of times each predictor was fitted in a user model out of a maximum of 17,066 times.

Table 9: *Distribution of the Predictor Variables*

Predictor	Number	Percentage
Duplicate	11,492	67.34
LAGS(ItemRank,1)	10,472	61.36
Qlength	6,596	38.65
Informational	6,409	37.55
Short_elapsed_time	4,369	25.60
DIFF(ranklag1,1)	3,281	19.23
Moderate_elapsed_time	3,060	17.93
Navigational	2,975	17.43
DIFF(avgranklag2and3,1)	1,904	11.16
Afternoon	1,887	11.06
Morning	1,683	9.86
New	1,649	9.66
New_session	1,598	9.36
Weekend	1,513	8.87
Specialization	1,445	8.46
Reformulation	1,394	8.17
DIFF(avgranklag4and5,1)	1,377	8.07
Gen_with_reform	1,173	6.87
Night	1,037	6.08
Generalization	1,020	5.98

Table 9 also lists the percentage distribution for each of those variables. The most widely used predictors are *duplicate* which was used in roughly 67 percent of the user models and *Ranklag1* which was fitted for about 61 percent of the user models. This implies that *duplicate* and *Ranklag1* were major factors which had a significant impact on user browsing behavior. *Informational* and *query length* were used to fit slightly more than one-third of the user models to predict the rank clicked. The least used predictors were *gen_with_reform* (generalization of a query by reformulation), *generalization*, and *night* which were used in about 6 to 7 percent of the models.

Table 10 shows the average values of the coefficients of the independent variables which were used to characterize the model for the aggregate set of users in this

analysis and Figure 14 shows the plot of the average coefficient value for each of the predictor variables.

Table 10: *Mean values of the coefficients of the predictor variables*

Predictor	Mean
Afternoon	.17753
DIFF(avgranklag2and3,1)	-.04440
DIFF(avgranklag4and5,1)	-.01104
DIFF(ranklag1,1)	-.10093
Duplicate	2.04246
Gen_with_reform	2.09850
Generalization	-1.61486
Informational	1.00184
LAGS(ItemRank,1)	.42453
Moderate_elapsed_time	-.38721
Morning	.71814
Navigational	-1.22724
New	-.71152
New_session	-.75847
Night	1.30838
Qlength	.36579
Reformulation	.35805
Short_elapsed_time	-.17042
Specialization	1.24979
Weekend	.65520

Although each predictor will have a different coefficient value for each user based upon the unique time series model that was fitted to his transaction log history, analyzing the average values over the aggregate users helps to identify the general impact of each of the predictors. Out of the 20 predictors, 11 of them have positive coefficients, while 9 have negative coefficients.

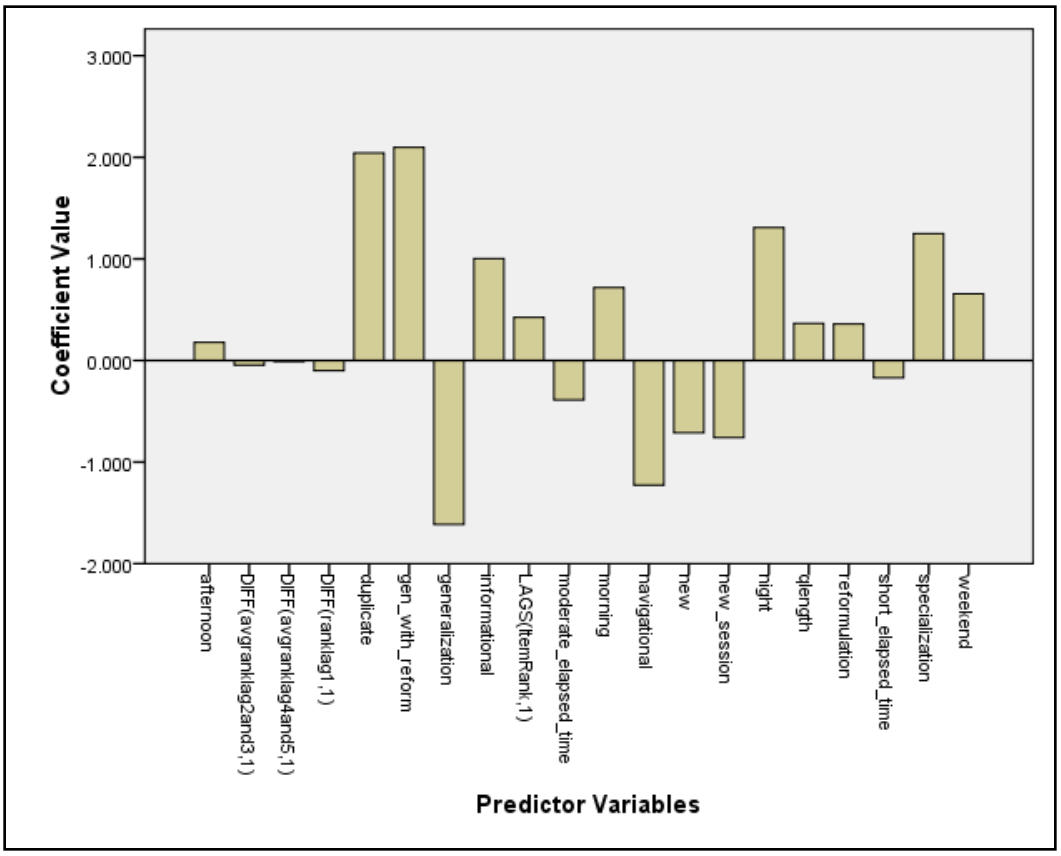


Figure 14: Analysis of Factors Affecting One-step-ahead Prediction of ItemRank Click

The time series equation that is used to predict the rank of the result that will be clicked contains one or more of these predictors and the sign of the coefficients affects the resultant rank clicked in a positive or negative manner. It is important to be clear in the understanding about the value of the predicted rank click. In our analysis, lower value of the rank corresponds to the top ranked results in the results page. For example, rank 1 refers to the first result that is listed on the SERP, rank 2 refers to the result that is listed in the second position of the SERP and so on. Thus, a positive coefficient increases the predicted value of the rank that will be clicked while a

negative coefficient decreases the predicted value (which corresponds to clicking top ranked results).

The average values of the coefficients are generally in-line with what one would expect. *New* has a negative coefficient which causes a decrease in the predicted rank click. This should be expected because a user issuing a new query would be more likely to click on the top-ranked results. *Duplicate* and *Reformulation* have positive coefficients and increase the value of the predicted rank click. This makes sense because a user who has just issued the previous query again or reformulated the previous query presumably might not have been satisfied with results from the previous search and looks to click on lower-placed results, which have a higher rank value. *Navigational* query decreases the predicted rank and *Informational* query increases the predicted rank. This finding agrees with Jansen et al [17] who found that searchers with *navigational* queries clicked on the higher-placed (which have a lower rank value) results than did searchers with *informational* and *transactional* needs. *Query Length* is found to increase the predicted rank that would be clicked, and this agrees with the finding by Zhang et al [19] whose results show that users who enter the shortest queries are more likely to click on the top most ranked results. Searching during the *Weekend* is shown to increase the rank that would be clicked compared to *Weekday* which is the reference for the models. The first differences of the *Ranklag1*, *avgranklag2and3*, and *avgranklag4and5* have low negative coefficients which

suggest that this behavior changes rapidly from user to user depending on their own personal transaction log history.

To determine the relative importance of the significant predictors, we have to look at the values of the standardized coefficients. Table 11 lists the predictors with their average standardized coefficient values and Figure 15 plots the significance level of each of the predictor variables.

Table 11: *Mean values of the standardized coefficients of the predictor variables*

Predictor	**Mean**
Afternoon	.08498
DIFF(avgranklag2and3,1)	-.03003
DIFF(avgranklag4and5,1)	-.02145
DIFF(ranklag1,1)	-.10893
Duplicate	.30526
Gen_with_reform	.09562
Generalization	.09718
Informational	.32713
LAGS(ItemRank,1)	.47610
Moderate_elapsed_time	-.08251
Morning	.10156
Navigational	.13105
New	.00956
New_session	-.09796
Night	.11653
Qlength	.38608
Reformulation	.07879
Short_elapsed_time	-.17254
Specialization	.13515
Weekend	.12337

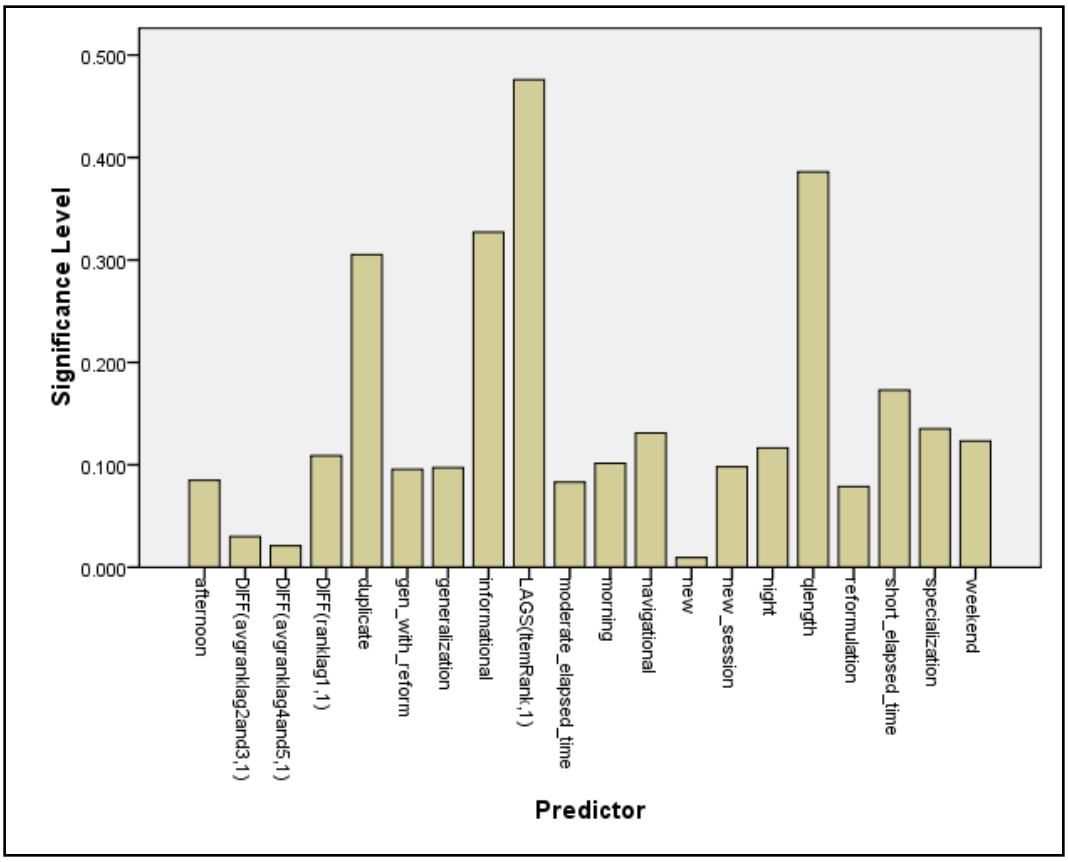


Figure 15: Significance Level of the Predictor Variables

We can see from Figure 15 that the most significant predictor of *ItemRank* is *RankLag1*, which is the value of *ItemRank* clicked one-step behind. This is to be expected because a good example of a time series is one that can be predicted from lagged values of itself. *Query Length* is also found to be highly significant and the fact that it has an average positive coefficient implies that users searching for longer queries tend to click on lower-placed results. Excluding the above two predictors, all the other predictors are indicator variables which take on binary values of 0 or 1 depending on whether the corresponding event occurred or not. Thus, their significance can be directly compared with each other because it strictly depends only

on the values of its coefficients. Among them, *informational* and *duplicate* queries are found to cause the most significant impact on the rank clicked by the user. All the others have similar levels of significance with a slight variance except for *new* which is shown to cause the least amount of change in predicting the rank click. This could be because of the fact that a *new* query always occurs with the start of a new session and part of its impact might have been absorbed by the *new_session* variable.

Naturally, there are certain limitations present in this study. Since our analysis involves developing prediction equations for the individual user, it is vital that the search engine is able to identify the user by other means rather than solely depending on the IP address in order to build the transaction log history. However, the search engines nowadays have the means of identifying the user by having them sign in to their personalized pages, among other methods such as desktop search bars. There is currently no way to find out if the search behavior of a user changes when using a different search engine. But ideally, it should not matter because the model obtained is applicable only for the search engine from which it was developed.

Some of the independent variables used in this research were obtained from the additional calculated fields listed in Table 2, which were computed using algorithms from previous works and do not have an accuracy of a 100 percent either due to errors in the query terms or because of multiple underlying user intents [17]. This may have resulted in reduced accuracy of the user models reflected in the adjusted R^2 values.

Nevertheless, our time series models obtained for entire set of active users show an average adjusted R^2 value of 57 percent. This does not imply that the model on an average is able to accurately predict 57 percent of the rank clicks but explains 57 percent of the variance that is exhibited by *ItemRank*. The accuracy is determined by the standard error of the estimate that is used to calculate the 95 percent confidence intervals for the predicted *ItemRank*. Our results show that search engine user behavior can be modeled using time series analysis where every user can be characterized by a unique set of equations that can help to predict his or her future actions. In particular, we can formulate individual equations for every user in order to predict the rank the user is likely to click one-step ahead. The aggregate analysis has helped to identify the significant predictors for *ItemRank* and the users can potentially be clustered into similar groups by the independent variables present in their time series model. Search engines can either use the individual user models to predict the future user actions or they can cluster users based on the similarity of the individual user models or use a common model to predict their actions.

Chapter 6

Conclusion and Future Research

In this research, we have demonstrated how the methodology of time series analysis using time series regression can be applied to Web search transaction logs to model the user Web searching behavior. In particular, this study is a pioneering effort in using time series regressive techniques to model the individual user behavior and develop a unique equation to predict the rank that will be clicked one-step ahead for each user. We explored two different techniques of applying time series analysis, the ARIMA method and the time series regression method and opted in favor of time series regression due to its advantages over the former for our research. We calculated important temporal factors and identified relevant independent variables that were used to fit the time series models for every active user in our dataset. We ran through a sample analysis for a single user and provided a framework for using time series analysis for developing a predictive model for the individual user. The aggregate analysis helped in recognizing similar behavior among users and in identifying the significant predictors of the rank of the result that would be clicked one-step ahead.

In future studies, we would like to study the possibility of predicting some of the independent variables that were used to predict the future rank clicks. Since it was possible to predict the rank that would be clicked one-step ahead by using its previous lagged values along with categorical variables like user intent and query modification pattern, it could be possible to predict the modification pattern from the its own

lagged values, rank clicks and other categorical variables using time series analysis techniques. We would also like to perform a K-means cluster analysis which is a tool designed to assign cases to a fixed number of groups (clusters) whose characteristics are not yet known but are based on a set of specified variables. It could be very useful when there are a large number (thousands) of cases, and therefore in our case should be able to provide some interesting user clusters based on the different predictor variables.

References

1. *World Wide Web Size*. 2009; Available from: www.worldwidewebsize.com.
2. Jansen, B., A. Spink, and I. Taksa, *Handbook of Research on Web Log Analysis (forthcoming)*. Handbook of Research on Web Log Analysis. 2008, Hershey, PA, USA: IGI Global.
3. Agichtein, E., et al., *Learning user interaction models for predicting web search result preferences*, in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 2006, ACM: Seattle, Washington, USA. p. 3-10.
4. Jansen, B., *Search log analysis: What it is, what's been done, how to do it.* Library & Information Science Research, 2006. **28**(3): p. 407-432.
5. Zhang, Y., B.J. Jansen, and A. Spink, *Identification of Factors Predicting ClickThrough in Web Searching Using Neural Network Analysis.* ASIS&T, 2008.
6. Broder, A., *A taxonomy of web search.* SIGIR Forum, 2002. **36**(2): p. 3-10.
7. Jansen, B.J., D. Booth, and A. Spink, *Predicting query reformulation during web searching*, in *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*. 2009, ACM: Boston, MA, USA. p. 3907-3912.
8. *Top five search engines*. 2009; Available from: http://www.seoconsultants.com/search-engines/.
9. Jansen, B. and A. Spink, *How are we searching the World Wide Web? A comparison of nine search engine transaction logs.* Information Processing & Management, 2006. **42**(1): p. 248-263.
10. Box, G., G. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting & Control (3rd Edition)*. 1994: {Prentice Hall}.
11. Belkin, N.J., *Interaction with Texts: Information Retrieval as Information-Seeking Behavior*, in *Information retrieval '93. Von der Modellierung zur Anwendung*. 1993, Universitaetsverlag Konstanz: Konstanz, Germany. p. 55-66.
12. Nancy C. M. Ross, D.W., *End user searching on the Internet: An analysis of term pair topics submitted to the Excite search engine.* Journal of the American Society for Information Science, 2000. **51**(10): p. 949-958.
13. Wen, J., J. Nie, and H. Zhang, *Query clustering using user logs.* ACM Trans. Inf. Syst., 2002. **20**(1): p. 59-81.
14. Silverstein, C., et al., *Analysis of a very large web search engine query log.* SIGIR Forum, 1999. **33**(1): p. 6-12.

15. Jansen, B.J., et al., *Defining a session on Web search engines: Research Articles.* J. Am. Soc. Inf. Sci. Technol., 2007. **58**(6): p. 862-871.
16. Beitzel, S.M., et al., *Hourly analysis of a very large topically categorized web query log*, in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. 2004, ACM: Sheffield, United Kingdom. p. 321-328.
17. Jansen, B.J., D.L. Booth, and A. Spink, *Determining the informational, navigational, and transactional intent of Web queries.* Inf. Process. Manage., 2008. **44**(3): p. 1251-1266.
18. Ozmutlu, S., A. Spink, and H.C. Ozmutlu, *A day in the life of Web searching: an exploratory study.* Information Processing and Management, 2004. **40**(2): p. 319-345.
19. Zhang, Y., B.J. Jansen, and A. Spink, *Time series analysis of a Web search engine transaction log.* Information Processing & Management, 2008. **In Press, Corrected Proof**.
20. Liu, N., et al. *Web Query Prediction by Unifying Model*. in *Data Mining Workshops, 2008. ICDMW '08. IEEE International Conference on*. 2008.
21. Piwowarski, B. and H. Zaragoza, *Predictive user click models based on click-through history*, in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. 2007, ACM: Lisbon, Portugal. p. 175-182.
22. Dupret, G.E. and B. Piwowarski, *A user browsing model to predict search engine click data from past observations*, in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 2008, ACM: Singapore, Singapore. p. 331-338.
23. Craswell, N., et al., *An experimental comparison of click position-bias models*, in *Proceedings of the international conference on Web search and web data mining*. 2008, ACM: Palo Alto, California, USA. p. 87-94.
24. Aris, A., et al., *Representing Unevenly-Spaced Time Series Data for Visualization and Interactive Exploration*, in *Human-Computer Interaction - INTERACT 2005*. 2005. p. 835-846.
25. Darlington, R.B. *A regression approach to time series analysis*. 1996; Available from: http://www.psych.cornell.edu/Darlington/series/series0.htm.
26. Nau, R.F. *Fitting Time Series Regression Models*. Available from: http://www.duke.edu/~rnau/411l696.htm.
27. Montgomery, D., *Design and Analysis of Experiments*. 2004: Wiley.

28. Jansen, B.J., D.L. Booth, and A. Spink, *Patterns of Query Modification during Web Searching (Forthcoming).* Journal of the American Society for Information Science and Technology, 2009.
29. Jansen, B.J., M. Zhang, and A. Spink, *Patterns and transitions of query reformulation during web searching.* International Journal of Web Information Systems, 2007. **3**(4): p. 328 - 340.
30. Park, S., Ho, and Jin, *End user searching: A Web log analysis of NAVER, a Korean Web search engine.* Library & Information Science Research, 2005. **27**(2): p. 203-221.
31. Kutner, M., C. Nachtsheim, and J. Neter, *Applied Linear Regression Models*. 2004: McGraw-Hill/Irwin.
32. Field, A., *Discovering Statistics Using SPSS (Introducing Statistical Methods series)*. 2009: Sage Publications Ltd.

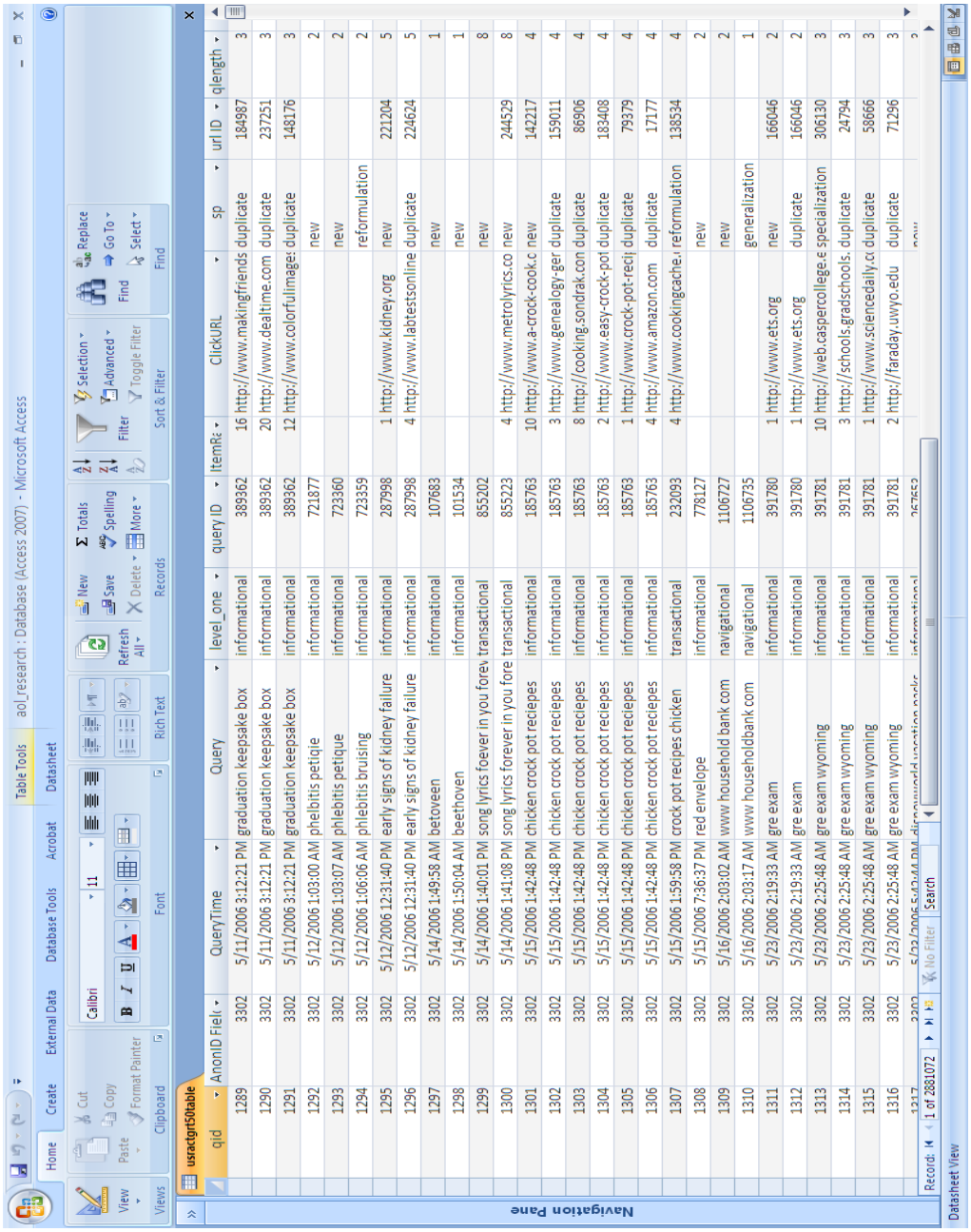
Appendix A

Data Collection:

Figure 16: Snapshot of a sample of data collected by AOL transaction log stored as an ASCII file

Appendix B

Data Preparation:



qid	AnonID	QueryTime	Query	level_one	query ID	ItemRa	ClickURL	sp	url ID	qlength
1289	3302	5/11/2006 3:12:21 PM	graduation keepsake box	informational	389362	16	http://www.makingfriends	duplicate	184987	3
1290	3302	5/11/2006 3:12:21 PM	graduation keepsake box	informational	389362	20	http://www.dealtime.com	duplicate	237251	3
1291	3302	5/11/2006 3:12:21 PM	graduation keepsake box	informational	389362	12	http://www.colorfulimages	duplicate	148176	3
1292	3302	5/12/2006 1:03:00 AM	phelbitis petiqie	informational	721877			new		2
1293	3302	5/12/2006 1:03:07 AM	phlebitis petique	informational	723360			new		2
1294	3302	5/12/2006 1:06:06 AM	phlebitis bruising	informational	723359			reformulation		2
1295	3302	5/12/2006 12:31:40 PM	early signs of kidney failure	informational	287998	1	http://www.kidney.org	new	221204	5
1296	3302	5/12/2006 12:31:40 PM	early signs of kidney failure	informational	287998	4	http://www.labtestsonline	duplicate	224624	5
1297	3302	5/14/2006 1:49:58 AM	betoveen	informational	107683			new		1
1298	3302	5/14/2006 1:50:04 AM	beethoven	informational	101534			new		1
1299	3302	5/14/2006 1:40:01 PM	song lyrics foever in you forev	transactional	855202			new		8
1300	3302	5/14/2006 1:41:08 PM	song lyrics forever in you fore	transactional	855223	4	http://www.metrolyrics.co	new	244529	8
1301	3302	5/15/2006 1:42:48 PM	chicken crock pot reciepes	informational	185763	10	http://www.a-crock-cook.c	new	142217	4
1302	3302	5/15/2006 1:42:48 PM	chicken crock pot reciepes	informational	185763	3	http://www.genealogy-ger	duplicate	159011	4
1303	3302	5/15/2006 1:42:48 PM	chicken crock pot reciepes	informational	185763	8	http://cooking.sondrak.con	duplicate	86906	4
1304	3302	5/15/2006 1:42:48 PM	chicken crock pot reciepes	informational	185763	2	http://www.easy-crock-pot	duplicate	183408	4
1305	3302	5/15/2006 1:42:48 PM	chicken crock pot reciepes	informational	185763	1	http://www.crock-pot-reci	duplicate	79379	4
1306	3302	5/15/2006 1:42:48 PM	chicken crock pot reciepes	informational	185763	4	http://www.amazon.com	duplicate	17177	4
1307	3302	5/15/2006 1:59:58 PM	crock pot recipes chicken	transactional	232093	4	http://www.cookingcache.r	reformulation	138534	4
1308	3302	5/15/2006 7:36:37 PM	red envelope	informational	778127			new		2
1309	3302	5/16/2006 2:03:02 AM	www household bank com	navigational	1106727			new		2
1310	3302	5/16/2006 2:03:17 AM	www householdbank com	navigational	1106735			generalization		1
1311	3302	5/23/2006 2:19:33 AM	gre exam	informational	391780	1	http://www.ets.org	new	166046	2
1312	3302	5/23/2006 2:19:33 AM	gre exam	informational	391780	1	http://www.ets.org	duplicate	166046	2
1313	3302	5/23/2006 2:25:48 AM	gre exam wyoming	informational	391781	10	http://web.caspercollege.e	specialization	306130	3
1314	3302	5/23/2006 2:25:48 AM	gre exam wyoming	informational	391781	3	http://schools.gradschools.	duplicate	24794	3
1315	3302	5/23/2006 2:25:48 AM	gre exam wyoming	informational	391781	1	http://www.sciencedaily.cc	duplicate	58666	3
1316	3302	5/23/2006 2:25:48 AM	gre exam wyoming	informational	391781	2	http://faraday.uwyo.edu	duplicate	71296	3

Figure 17: After cleaning and importing the transaction log data into a database and calculating additional attributes

Appendix C

Data Analysis:

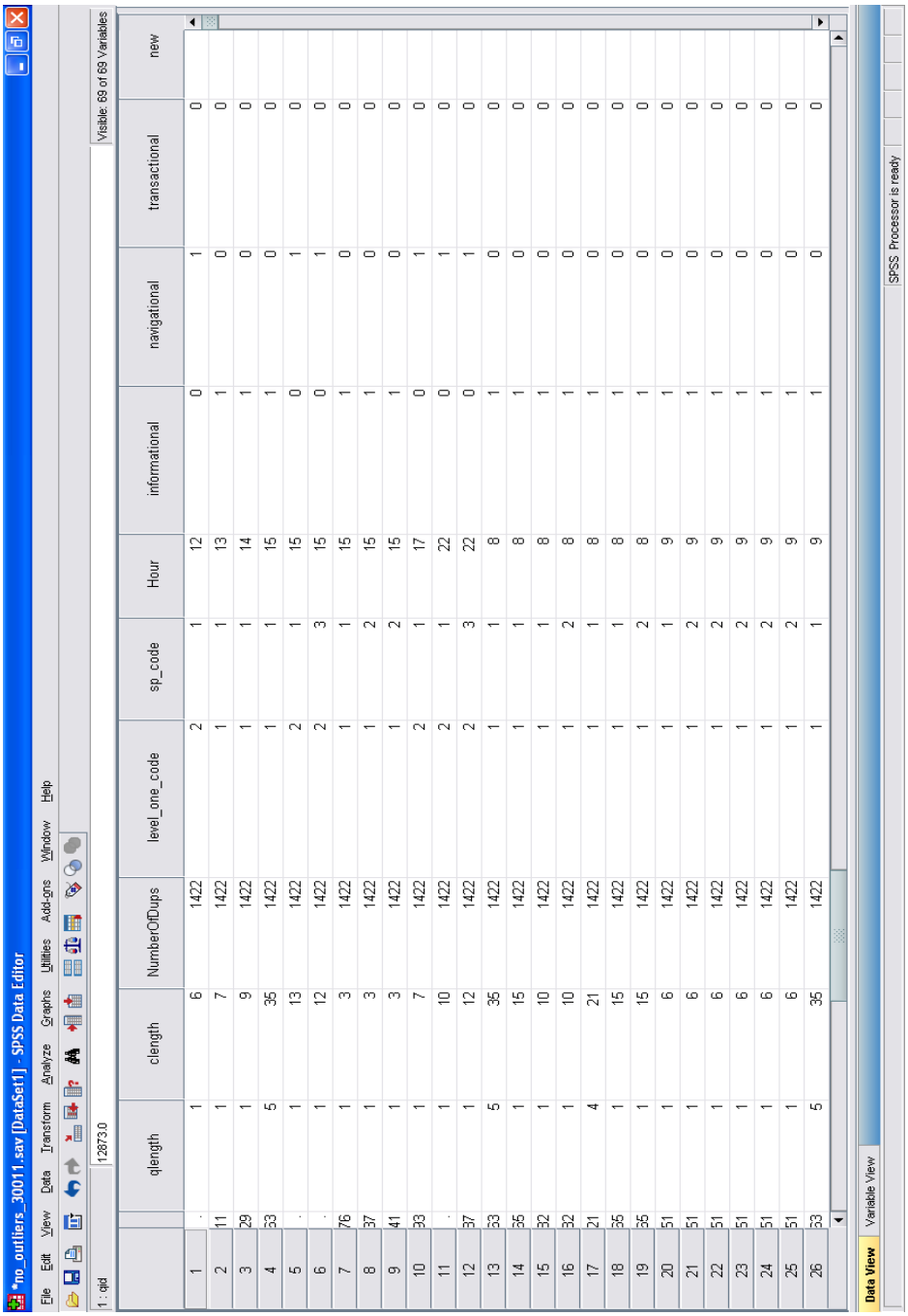


Figure 18: After importing the cleaned and prepared database to SPSS for determining the predictor variables and coding the time series analysis

Appendix **D**

Derivation of the Estimation of the parameters in Linear Regression Models

Consider the multiple linear regression model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon$$

There are k regressors,

B_i , i=1,.....k is the coefficient of the ith regressor,

B_0 is the value of the intercept,

ε is the error term assumed to have zero mean and $\{\varepsilon_i\}$ are uncorrelated random variables.

The data for regression in terms of observations is given below.

y	x_1	x_2	$\cdots$	x_k
y_1	x_{11}	x_{12}	$\cdots$	x_{1k}
y_2	x_{21}	x_{22}	$\cdots$	x_{2k}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_n	x_{n1}	x_{n2}	$\cdots$	x_{nk}

The model equation in terms of the observations is given by

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_k x_{ik} + \varepsilon_i$$

The above equation in terms of the observations in matrix notation is given by

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}$$

where

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

In general, **y** is an (n x 1) vector of the observations, **X** is an (n x p) matrix of the levels of the independent variables, **β** is a (p x 1) vector of the regression coefficients, and **ε** is an (n x 1) vector of random errors.

We wish to find the vector of least squares estimators $\widehat{\boldsymbol{\beta}}$, that minimizes

$$L = \sum_{i=1}^{n} \epsilon_i^2 = \boldsymbol{\epsilon}'\boldsymbol{\epsilon} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

L may be expressed as

$$\begin{aligned} L &= \mathbf{y}'\mathbf{y} - \boldsymbol{\beta}'\mathbf{X}'\mathbf{y} - \mathbf{y}'\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \\ &= \mathbf{y}'\mathbf{y} - 2\boldsymbol{\beta}'\mathbf{X}'\mathbf{y} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \end{aligned}$$

The least squares estimators must satisfy

$$\left.\frac{\partial L}{\partial \boldsymbol{\beta}}\right|_{\hat{\beta}} = -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{0}$$

which simplifies to

$$\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{y}$$

To solve the equation, multiply both sides by the inverse of $\mathbf{X}'\mathbf{X}$.

Thus, the least square estimator of **β** is

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$