Web Query Prediction by Unifying Model

Ning Liu\textsuperscript{1} Jun Yan\textsuperscript{1} Shuicheng Yan\textsuperscript{2} Weiguo Fan\textsuperscript{3} Zheng Chen\textsuperscript{1}

\textsuperscript{1}Microsoft Research Asia, Beijing, P.R.China
{ningl, junyan, zhengc}@microsoft.com
\textsuperscript{2}National University of Singapore
eleyans@nus.edu.sg
\textsuperscript{3}Virginia Tech, Blacksburg
VA 24061, USA
wfan@vt.com

Abstract

Recently, many commercial products, such as Google Trends and Yahoo! Buzz, are released to monitor the past search engine query frequency trend. However, little research has been devoted for predicting the upcoming query trend, which is of great importance in providing guidelines for future business planning. In this paper, a unified solution is presented for such a purpose. Besides the classical time series model, we propose to integrate the Cosine Signal Hidden Periodicities Model to capture periodic information of query time series. Motivated by the fact that these models cannot capture the external accidental event factors which could significantly influence the query frequency, the query correlation model is also modified and integrated for predicting the upcoming query trend. Finally linear regression is utilized for model unification. Experiments based on 15,511,531 queries from a commercial search engine query log ranging within 283 days well validate the effectiveness of our proposed unified algorithm.

1. Introduction

With the rapid growth of World Wide Web, search engines nowadays are significantly impacting people’s daily life [9, 2]. Since the search queries’ frequency can effectively reflect the hotspots on the Web [14], predicting the frequency trend of user searches is becoming more and more desirable and useful. As some examples, if the queries’ future frequency trend can be predicted, the Web content providers will benefit greatly from emphasizing and optimizing the hottest portion of what they deliver so as to attract more traffic; and for online advertisers, bidding for the keywords to be hot is crucial for increase of the click-through rates of advertisements and sales.

With the prevailing of Web search engines like Google, Yahoo and Microsoft Live Search, the query log collected by these search engines have been utilized in various ways [3, 15, 10]. The queries submitted by the end users directly reflect the users’ search intentions, and hence are effective to reveal what are being or have been hot on the Web. Numerous commercial products, e.g. Google Trends (http://www.google.com/trends) and Yahoo! Buzz (http://buzz.yahoo.com/), have emerged in the past few years. Users can easily observe which queries have been hot in the past from these products. However, to the extent of our knowledge, information provided by most of existing products is restrained to the historical query frequency in search engine, and they cannot predict the frequency trend of queries on the Web in the future. The market demand for the latter is also great, and the prediction of the upcoming hotness can provide important.

Though some classical time series models [11, 5], such as Autoregressive (AR) model [8], Autoregressive Moving Average (ARMA) model [12] and Auto Regressive Integrated Moving Average (ARIMA) model [11], can predict the frequency change of the time series data, there exist big challenges for predicting the query frequency trend by these models. Firstly, many queries often show periodic characteristics. For example, the query “cinema” gets hot in every weekend since it is time for entertainment for many people. However, most of the classical prediction models such as AR and ARMA did not take this periodic information into consideration. In addition, many query frequencies might be significantly influenced by external accidental factors. For example, the query “collect donations” always gets hot right after an earthquake happens. It is hard for these classical time series models to make accurate prediction for such queries. However, we could predict that the series data of “collect donations” is going to be hot if we observe the unusual peak for query “earthquake”.

In this work, we propose to unify both periodicity and accidental factors with classical AR time series
model for predicting the upcoming query frequency, i.e. query hotness on the Web. Firstly, the periodicity of the temporal query data is explicitly fitted by Cosine Signal Hidden Periodicities Model, which directly brings the advantages over classical AR model on periodic data. Secondly, we modify the query correlation model to calculate the temporal correlation between related queries such that the influences coming from external accidental factors can be modeled. Finally, the prediction performance is boosted by unifying the results of these three different models by linear regression. The divide and conquer strategy is used for model unification by applying different regression models for predicting different kinds of queries.

Experimental results on 15,511,531 queries from the MSN search engine query log ranging within 283 days demonstrate the effectiveness of our proposed unified algorithm in predicting the query frequency trend on the Web. The statistical t-test shows that the improvement of our proposed unified model is significant in contrast to the classical AR, ARMA and ARIMA model. The main contribution of this paper is summarized as follows. (1) We propose to unify AR time series model, Cosine Signal Hidden Periodicities Model and modified temporal query correlation model by linear regression. This provides a general way for predicting the trend of queries. (2) We experimentally validated the effectiveness of our proposed unified model on more than 15 million real world queries.

2. Problem formulation

In this work, we use the conventional time series representation for temporal query data, namely discontinuous frequency function [6]. A query time series is represented as a sequence of integers, each of which stands for the number of times the query issued by all users at that time span. The frequency function of a query time series $Q$ over $M$ time units is a sequence of length $M$, denoted as,

$$Q = \{q_1, q_2, ..., q_M\},$$

where $q_i$ represents the aggregate used number of query $Q$ by users in the $i^{th}$ time span, and $M$ is the total length of the time series. A time span could be an hour, a day, a week or a month. In all the case studies and experiments of this paper, we consider one day as the time span. To predict the future query frequency trend, we define the query frequency prediction problem as forecasting a number of next step frequencies based on the historical values of the time series.

**Definition 1.** Given the first $N$ elements of the time series $Q$, the problem of $(M-N)$-step prediction is defined as

$$\{\hat{q}_{N+1}, \hat{q}_{N+2}, ..., \hat{q}_M\} = f(q_1, q_2, ..., q_N),$$

where $f$ is the prediction function used for describing the relationship between the first $N$ elements and the last $M-N$ elements of $Q$.

Let $\{\hat{q}_{N+1}, \hat{q}_{N+2}, ..., \hat{q}_M\}$ stands for the predicted results of query time series $Q$ by prediction function $f$ and $\{q_{N+1}, q_{N+2}, ..., q_M\}$ stands for the true frequency of query time series $Q$ during the same time period, the objective of the query trend prediction problem is to minimizing the error between the frequency prediction $\{\hat{q}_{N+1}, \hat{q}_{N+2}, ..., \hat{q}_M\}$ and $\{q_{N+1}, q_{N+2}, ..., q_M\}$ by learning a proper prediction function $f$, i.e.

$$f^* = \arg\min_f ||f(q_1, q_2, ..., q_N) - \{q_{N+1}, q_{N+2}, ..., q_{N+M}\}||,$$

where $|| \cdot ||$ stands for the L-2 norm by considering the two sequences as two vectors. In this work, our task is to consider different models which can capture the temporal information in query time series, i.e. the general time series information, the periodic information as well as the accidental information of queries respectively. Finally we unify these three prediction functions to the final prediction function $f$.

3. Models for query trend prediction

3.1. Time series model

Numerous time series models can be used for predicting the trend of temporal data. Among them, one of the most simple and classical one is the Autoregressive (AR) model. In this work, we select it as the first component of our proposed unified algorithm. An AR model of order $p$, denoted as AR($p$), is,

$$q_i = c + \sum_{i=1}^{p} \varphi_i q_{i-n} + \varepsilon_i,$$

where $c$ is a constant, $\varphi_1, ..., \varphi_p$ are the model parameters, and $\varepsilon_i$ is the noise term. An AR model could be treated as an infinite impulse response filter.

The parameters of the AR model are usually estimated by Yule-Walker equations [13] or least square regression [7]. In this work, we implement the latter one. A standard “windowing” transformation is used to transfer a time series into a set of instances for regression analysis. Given a time series

$$Q = \{q_1, q_2, ..., q_N\},$$

an instance for regression analysis is defined as

$$y_i = (q_i, q_{i+1}, ..., q_{i+n})'.$$

Thus the AR parameters could be figured out by solving the following equation,
where

\[ \Phi Y = 0, \]

\[ \Phi = (\phi_1, \phi_2, ..., \phi_p, -1), \]

\[ Y = (y_1, y_2, ..., y_{N-1}). \]

As described above, the time series prediction problem could be transformed into a regression problem, and hence the complexity of this model is the same as linear regression. Through estimating the parameters by least square regression, the \( q_t \) can be directly estimated by \( q_{t-i}, i = 1, 2, \ldots, p. \)

3.2. Periodicity model

There often exists periodicity property for time series in real world tasks, especially for query time series. For example, the query “Christmas” has the obvious period of one year. In the field of Digital Signal Processing (DSP), the Cosine function is traditionally adopted to approach periodic time series as

\[ q_i = \sum_{j=1}^{k} A_j \cos(\omega_j t + \phi_j) + \xi_i, \quad (1) \]

where positive real number \( A_j \) is the Amplitude of Angular Frequency \( \omega_j \), \( \phi_j \) is the Phase of \( \omega_j \). We call equation (1) as Cosine Signal Hidden Periodicity (CSHP) model from which we can get to know the periodicities of \( q_i \) as

\[ T_j = \frac{2\pi}{\omega_j}, \quad j = 1, 2, \ldots, k. \]

Then the frequency spectrum of the model is given by

\[ S_N(\lambda) = \sum_{i=1}^{N} q_i e^{-i\omega_i} \lambda \in [-\pi, \pi], \quad (2) \]

and based on this we have the lemma 1 showed below [4]. \( \lambda \in [-\pi, \pi] \) is the parameter.

**Lemma 1.** if \( \exists k \) and \( \lambda_j^* \) such that \( S_N(\lambda_j^*) \geq S_N(\lambda) \)

\[ \lambda \in [\lambda_j^* - \frac{\pi}{2\sqrt{N}}, \lambda_j^* + \frac{\pi}{2\sqrt{N}}] \]

then the CSHP Model (1) has \( k \) periodicities, and the parameters of this model are estimated as

\[ \omega_j = \lambda_j^*, \quad T_j = \frac{2\pi}{\omega_j} = \frac{2\pi}{\lambda_j^*}, \]

\[ A_j = \frac{\sum_{i=1}^{N} q_i e^{-i\lambda_j^*}}{\omega_j}, \quad \phi_j = \arg(\omega_j). \]

We name all the queries which have periodicity as the periodic queries. In this paper, all the periodic queries are detected based on lemma 1.

Inspired by Lemma 1, we develop an approach, called Periodicity Detection Procedure (PDP), as listed in Table 1 to detect the periodicity of the query time series. As for the details, we first centralize the time series. And then transform it by equation (2). Finally we detect the peak points of the transformed time series, i.e. the periodicity of the query time series according to lemma 1.

**Table 1. The Periodicity detection procedure (PDP)**

| INPUT | Query time series \( Q = \{q_1, q_2, ..., q_N\} \) |
| OUTPUT | The periodicity \( T \) of \( Q \) if it is a periodic query |
| STEP 1. | Compute the mean of \( Q \): \( \bar{Q} = \frac{1}{N} \sum_{i=1}^{N} q_i \) |
| STEP 2. | Centralize \( Q \) to a zero mean time series \( X \): \( x_i = q_i - \bar{Q}, \quad t = 1, 2, ..., N \) |
| STEP 3. | Calculate \( S_N(\lambda) \) by equation (2) and judge whether \( S_N(\lambda) \) has peaks by applying Lemma 1 |
| STEP 4. | If \( S_N(\lambda) \) has peaks, output the periodicity \( T_j \) corresponds to each peak. Otherwise, \( Q \) is not a periodic query |

Based on the detected periodicities, the CSHP model can be easily leveraged for time series trend prediction. We first detect period of a query time series by PDP, then the prediction is conducted by \( \hat{Q} = \{\hat{q}_{n+1}, \hat{q}_{n+2}, ..., \hat{q}_{n+T}\} = \{\hat{q}_{n+1-T}, \hat{q}_{n+2-T}, ..., \hat{q}_{n+T}\} \) according to the period property. The routines of prediction with CSHP are illustrated in Table 2. The complexity of this procedure is comparable with linear regression [4].

**Table 2. Procedure for predicting with CSHP model**

| INPUT | A periodic time series \( \hat{Q} = \{\hat{q}_1, \hat{q}_2, ..., \hat{q}_N\} \) |
| OUTPUT | Prediction \( \hat{Q} \) |
| STEP 1. | Get the periodicity of the CSHP Model on \( Q \) by PDP introduced in Table 1, where \( q_i = \hat{Q} + \sum_{j=1}^{k} A_j \cos(\omega_j t + \phi_j) + \xi_i \) |
| STEP 2. | Suppose the strongest periodicity of \( Q \) is \( T_j \). Get the prediction \( \hat{Q} = \{\hat{q}_{n+1}, \hat{q}_{n+2}, ..., \hat{q}_{n+T}\} \) by the corresponding time period in its nearest period, i.e., \( \hat{Q} = \{\hat{q}_{n+1}, \hat{q}_{n+2}, ..., \hat{q}_{n+T}\} \) |

3.3. Query correlation model

The measure of temporal similarity between query time series was first introduced by Chien et al. in [6]. For the time series related to a given query \( Q \), a normalization step is firstly conducted for each time series. Let \( SUM_i \) be the total number of queries (not
necessarily distinct) at the $i^{th}$ time unit, $Q$ is normalized as,
\[
\hat{Q} = \{q_1, q_2, \ldots, q_N\},
\]
where $q_i = q_i / \text{SUM}_i$.

The temporal similarity between query time series is defined by considering $q_i$ of each query as a random variable. The correlation coefficient between two time series $\hat{Q}$ and $\hat{Q}'$ is defined as,
\[
sim(\hat{Q}, \hat{Q}') = \frac{1}{M} \sum \frac{\hat{q}_i - \mu(\hat{Q})}{\sigma(\hat{Q})} \frac{\hat{q}'_i - \mu(\hat{Q}')}{\sigma(\hat{Q}')} \tag{3}
\]
where $\mu(\hat{Q})$ is the average frequency of the normalized time series $\hat{Q}$ and $\sigma(\hat{Q})$ is the standard deviation of $\hat{Q}$. The similarity always lies within $[-1, 1]$, where 1 indicates an exact positive linear relationship, -1 indicates the opposite, and 0 indicates full independence. The effectiveness of this similarity measure in seeking correlated queries was well validated in [6].

Different from many previous query correlation models, we modify Chien’s work to involve the time lag in the computation for prediction purpose. To the given query $\hat{Q}$, the correlation between it and another query $\hat{Q}'$ with time lag $l$ is calculated by,
\[
sim(\hat{Q}, \hat{Q}') = \frac{1}{M} \sum \frac{\hat{q}_i - \mu(\hat{Q})}{\sigma(\hat{Q})} \frac{\hat{q}'_{i+l} - \mu(\hat{Q}')}{\sigma(\hat{Q}')} \tag{3}
\]

Through this way, if query $\hat{Q}$ is strongly related to $\hat{Q}'$ with lag $l$, we can predict that $\hat{Q}$ will be hot after time $l$ if we observe the hotspot of $\hat{Q}'$ in current time. The optimal time lag for $\hat{Q}'$ is defined by,
\[
l' = \arg \max_l, \sim(\hat{Q}, \hat{Q}') , \text{ s.t.} l \geq 1 \tag{4}
\]

To simplify the computation, we arbitrarily assume $l = 1, 7,$ and 30, i.e. a day, a week, and a month, to select the time lag with the largest correlation value in all experiments of this paper. Based on the detected correlated queries, we derive a correlation model for query prediction by utilizing the information from all the correlated queries. Given a similarity threshold, let $W^1, W^2, \ldots, W^c$ be the $c$ correlated queries of $Q$ with time lags $l^1, l^2, \ldots, l^c$, and
\[
W' = \{w_1', w_2', \ldots, w_N'\}.
\]

Firstly, the same “windowing” transformation is applied for data preprocessing. Suppose the predicted results are $\hat{Q}' = \{q_{N+1}', q_{N+2}', \ldots, q_M'\}$, the modified correlation model for prediction is calculated by,
\[
\hat{q}_{N+t} = \sum_{l=1}^{c} \delta_{\delta_{N+t}, l} \cdot \frac{\sim(\hat{Q}, \hat{Q}')}{\delta_{\delta_{N+t-l}, t} w_{N+t-l}'}, t = 1, 2, \ldots, M - N \tag{5}
\]
where $\delta_{\delta_{N+t}, l} = 0$ if $N + t - l' \leq N$ and $\delta_{\delta_{N+t}, l} = 1$ otherwise. The $\sim(\hat{Q}, \hat{Q}')$ is used for weighting the importance of the query $\hat{Q}'$ in predicting $\hat{Q}$.

We summarize the procedure for predicting the query trend through modified correlation model in Table 3. The complexity of this step depends on the computation of the signal correlation, i.e. Eqn (3). When it comes to implementation issues, a number of statistical approximate strategies have been proposed in [6] to accelerate the computation of correlations among large amount of query data. The details which are ignored in this article please refer to [6].

<table>
<thead>
<tr>
<th>Table 3. Query trend predicting with modified correlation model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INPUT</strong> Time series $Q = {q_1, q_2, \ldots, q_N}$</td>
</tr>
<tr>
<td><strong>OUTPUT</strong> Prediction $\hat{Q}$</td>
</tr>
<tr>
<td><strong>STEP 1</strong> Normalize $Q$ and find its related queries $W^1, W^2, \ldots, W^c$ with corresponding time lag through Eqn (3) and (4)</td>
</tr>
<tr>
<td><strong>STEP 2</strong> Predict the query trend through Eqn (5) for $\hat{Q}' = {q_{N+1}', q_{N+2}', \ldots, q_M'}$</td>
</tr>
</tbody>
</table>

4. Experiments

In this Section, we evaluate the effectiveness of the proposed unified model for query frequency trend prediction on the real query data from the MSN search engine query log. We had access to a collection of 15,511,531 queries along with their daily aggregate clicks from October 2006 through August 2007, 283 days in total. The algorithmic performance in improving query frequency prediction is compared in detail with some classical time series prediction models like AR, ARMA and ARIMA.

4.1. Experimental configuration

For evaluating the frequency prediction accuracy, we need first estimate the model parameters for different prediction models and the parameters related with experimental configuration:

- Train/test ratio to split the whole data set. The training data should be sufficiently large to ensure the accuracy of model parameters, and the length of the test series cannot be too long to be unpredictable. In our experiments, the data of the first 240 days are used as model training data, and the remaining 43 days are used as ground truths for testing.
- Parameters for classical time series models AR, ARMA and ARIMA. These parameters are used as configuration for baseline algorithms, and include
the degree of differencing and the moving average order. We use the Akaike Information Criterion (AIC) [1] to determine the appropriate values for these parameters. As an example, for the AR model, which is the component model of our proposed unified model, $p$ is set as 10 empirically according to the Akaike Information Criterion (AIC) [1].

- A limitation of this unified model is that it cannot directly predict the trend of the new queries, which has not appeared in the query log, due to the lack of historical information. In our developed system, we firstly find the similar queries of the new query by substring match, i.e. edit distance. And then the trend of the new query is predicted by the similar queries.

For the results evaluation, we adopt the Root Mean Square Error (RMSE) [2] as the measurement to evaluate the accuracy of the frequency prediction results. The definition of RMSE is given as

$$RMSE(\hat{Q}, Q) = \sqrt{\frac{1}{n} \sum_{i=N}^{n} (\hat{q}_i - q_i)^2},$$

where $n = M - N$, $\hat{Q} = \{\hat{q}_{N+1}, \hat{q}_{N+2}, \ldots, \hat{q}_M\}$ is the normalized predicted time series and $Q = \{q_{N+1}, q_{N+2}, \ldots, q_M\}$ is the normalized corresponding ground truth. For the overall evaluation, we define the Average RMSE (ARMSE) over all queries as,

$$ARMSE = \frac{1}{|Q|} \sum_{Q} RMSE(\hat{Q}, Q),$$

where $|Q|$ is the number of queries we take into consideration in the summation. All the models in this work are implemented in C# language and all these experiments are run on a workstation with dual-core 2.8G AMD processor and 10G memories.

4.2. Experimental results

In this Section, we demonstrate the experimental results of the proposed unified model over our query collection (with more than 15 million queries) collected from MSN search engine query log. Evaluated by the AR model, we show in Figure 1 the results of our proposed unified model in contrast to the classical time series models, which are AR, ARMA and ARIMA. The baseline models did not explicitly consider query periodicity and correlation over the collected queries. To eliminate the effect of different scale of queries, we normalize the queries in Euclidian space before evaluation. For example, for the predicted query $\hat{Q} = \{\hat{q}_{N+1}, \hat{q}_{N+2}, \ldots, \hat{q}_M\}$, we let $\hat{q}_{N+1} = q_{N+1}/\|\hat{Q}\|$, where $\|\hat{Q}\|$ is the L-2 norm of $\hat{Q}$. The same normalization strategy is applied to the ground truth series and thus the RMSE of all queries is measured at the same scale.

The baseline algorithms AR, ARMA and ARIMA are three of the most commonly used time series models for temporal data prediction. From Figure 1 we see that the unified model can perform much better than the three commonly used classical models, and the RMSE can be reduced 29%, 27% and 24% respectively. To verify that the improvement of the unified model is statistically significant in contrast to the three baseline models, we conduct the T-test using results on all queries but general queries. We did not consider general queries in the T-test since the results of our unified model are the same as the results of AR model for these queries. The T-test results are given in Table 4 which shows that the improvements of our proposed unified model are statistically significant from the baseline models.

![Figure 1. ARMSE of the proposed unified model in contrast to different classical time series models](image)

![Table 4. T-test of the unified model in contrast to classical time series models](table)

<table>
<thead>
<tr>
<th>Unified model v.s.</th>
<th>AR</th>
<th>ARMA</th>
<th>ARIMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-test</td>
<td>0.000767</td>
<td>0.000986</td>
<td>0.00349</td>
</tr>
</tbody>
</table>

Though we have experimentally proved that the unified model can outperform the classical AR, ARMA and ARIMA model in predicting the next 43 days’ query frequency based on the 283 days’ training data, many of the real world applications do not require such a long term prediction. In this experiment, we aim to answer the question that whether the unified model can perform better in terms of shorter term prediction. In Figure 2, we show the ARMSE over all our collected queries in predicting the query trend from 1 future day to 43 future days. The x-axis is the number of days we will predict and the y-axis is the ARMSE.

From Figure 2 we can observe that both our proposed unified model and the classical time series model can predict the query trend more accurate for short term prediction than for long term prediction. Among the three baseline models, the ARIMA model can perform better than the other two for the leading 20 days in prediction. The ARMA model is a little bit better than the AR model at the leading 10 days. However, the three baseline models perform very close...
for the very short term prediction (only predict two days),
the ARIMA model is a little bit better than the
proposed unified model. Except for this case, the
unified model is much better than the three classical
baseline models. One interesting observation is that
there has three jump points in the curve of the unified
model. The first jump occurs around 7 days. The
second jump occurs around 15 days and the last jump
occurs around 30 day. Though we currently cannot
theoretically analyze this phenomenon, we observe that
many queries has the periodicity of a week (7 days),
half a month (15 days) and a month (30 days). We
believe there has strong correlation between the three
jump points in Figure 5 and the periodic queries.

Figure 2. The prediction error of different models
in predicting queries for different number of days.

5. Conclusion and future work

Predict the query trend and unveiling the hotspots
on the Web has great business potentials. Most existing
approaches can only analyze the past query trends. In
this work, we predict the upcoming hotness of queries,
which can take additional benefits to real world
applications, by incorporating traditional AR time
series model with periodicity model and our proposed
correlation model with possible time lags. The
periodicity model is superior over traditional ones
when the time series shows strong periodic
characteristic. The correlation model further enhances
the prediction capability by modeling the correlations
between the examined query and other related queries.
To unify the classical time series model, periodic
model and correlation model, we propose to classify
the queries into general queries, periodic queries and
accidental queries such that the three models can be
unified for different category of queries respectively.
Experiments on real query data collected from MSN
search engine query log of 283 days well demonstrate
the capability of our proposed model in predicting the
hotspots on the Web.

In the future, we plan to further study the semantic
query clustering strategies so that we can not only
predict the hot queries but also the hot semantic topics,
which are clusters of queries. In addition, advanced
model unification strategies will be explored to further
improve the proposed unified model.

6. References

identification. IEEE Transactions on Automatic Control, 19,
6, 716-723.
Groundwater Modeling: Simulation of Flow and Advective
clustering of a search engine query log. In Proceedings of the
Sixth ACM SIGKDD international Conference on
Knowledge Discovery and Data Mining, 407-416.
Series Analysis, Forecasting and Control. Prentice Hall.
Introduction to Time Series and Forecasting, Springer Texts
in Statistics, 2nd ed.
between search engine queries using temporal correlation. In
Proceedings of the 14th international Conference on World
of the first-order autoregressive parameter. Journal of the
American Statistical Association, 74, 140-146.
American Life Project report
Analysis of Geographic Queries in a Search Engine Log, In
Proceedings of the LocWeb
University Press.
Autocovariance of an ARMA Model: An Exact Form
Trends, Journal of Webology, Volume 1, Number 2,
December
engine query logs for query recommendation. In Proceedings
of the 15th international Conference on World Wide Web
1039-1040.