Identification of Factors Predicting ClickThrough in Web Searching Using Neural Network Analysis

Ying Zhang
The Harold and Inge Marcus Department of Industrial and Manufacturing Engineering, College of Engineering, The Pennsylvania State University, University Park, PA 16802. E-mail: yzz114@psu.edu

Bernard J. Jansen
329F Information Sciences and Technology Building, College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16802. E-mail: jjansen@ist.psu.edu

Amanda Spink
Faculty of Information Technology, Queensland University of Technology Gardens Point Campus, 2 George St, GPO Box 2434, Brisbane QLD 4001 Australia. E-mail: ah.spink@qut.edu.au

In this research, we aim to identify factors that significantly affect the clickthrough of Web searchers. Our underlying goal is determine more efficient methods to optimize the clickthrough rate. We devise a clickthrough metric for measuring customer satisfaction of search engine results using the number of links visited, number of queries a user submits, and rank of clicked links. We use a neural network to detect the significant influence of searching characteristics on future user clickthrough. Our 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. We provide recommendations for leveraging these findings for improving the performance of search engine retrieval and result ranking, along with implications for search engine marketing.

Introduction

The usefulness of a search engine depends on the relevance of the results retrieved and ranked in response to user queries. While millions of Web pages may include a particular word or a phrase, some may be more relevant, popular, useful, or authoritative than others. Most search engines employ methods to rank the results to provide the *best*, *most useful*, or *most relevant* results first. How a search engine decides which pages are the best matches and in what order to show the results varies from one engine to another. The retrieval and ranking methods also change over time as Web use changes and new techniques evolve. Therefore, the evaluation of searching efficiency is a critical and ongoing research area.

To perform this evaluation, search engines record user-system interactions in a transaction log (a.k.a., search log or query log) for analysis. A search engine transaction log is an electronic record of the interactions that have occurred during a searching episode between a Web search engine and users searching for content on that Web search engine. Just as transaction logs have yielded comprehensive documentation of users' online behaviors, they have become important resources for system evaluation and studies of user searching behavior. The voluminous nature of such logs, however, means that companies interested in user behavior on the Web face enormous amounts of data to analyze to determine valuable metrics.

One of these commercial metrics is clickthrough rate (CTR), which is one measure of user satisfaction with the results retrieved by a search engine based on a query submitted by a user (Joachims, 2002; Joachims, Granka, Pan, Hembrooke, & Gay, 2005; Xue et al., 2004). Naturally, his may not always be the case. There are certainly times when higher clickthrough may indicate users not finding what they are looking for. Additionally, Dupret and Piwowarsk (2008) point out that search logs may be missing important data such as documents that the user has already seen. However, CTR is an important element reflecting the quality and effectiveness of commercial search engine and online advertising (Nettleton, Calderon, & Baeza-Yates, 2006), such as

Received August 4, 2008; revised October 11, 2008; accepted October 11, 2008

 DOI: 10.1002/asi.20993

sponsored search campaigns. Using the data recorded in the transaction log and knowing the number of results retrieved in response to a query, one can calculate the existing CTR.

Given the importance of clickthrough as a measure of user satisfaction and with clickthrough being the primary revenue generating mechanism for most search engines, it would be beneficial to develop more advanced inferential models that could predict future CTR of a given user based on current searching characteristics. Commercial search engine companies could then utilize user-system interactive data to improve the CTR by designing more efficient searching algorithms or advertising platforms, which could potentially improve revenue streams for online advertising. This is the primary motivation for focusing this research on clickthrough. Using methods that result in predictive models will aid search engines in serving relevant organic and sponsored results to users.

In this research, we identify and model the relationship between the data recorded in transaction logs (logon time, browser type, query length, etc) and future propensity of user clickthrough (i.e., how likely is the user to click on links in the results listing).

In the next sections, we first summarize concepts and previous work related to the use of Web transaction logs to investigate user behaviors. Then, the basic theories and training algorithms underlying our neural networks method are introduced. We used the multilayer perceptron neural network (MLPN), which is a backpropagation neural network. Afterwards, there is a discussion of the necessary data sets (training, testing, and evaluating) to build the corresponding neural networks, explore the constructed neural networks using our prepared data sets, and analyze the neutral network characteristics by varying parameters. Then, we present a sensitivity analysis of the input on clickthrough. Finally, the results and importance of the models utilized are highlighted before concluding with discussion of the findings.

Review of Literature

Web search engine transaction logs have become an important data collection method for studying information retrieval and searching. However, companies interested in Web user behavior face enormous amounts of data that they must analyze to gain worthwhile information. For example, Nielsen / NetRatings monitors the search behavior of approximately 500,000 people worldwide (Sullivan, 2006, 2008), and datasets of this size present significant challenges to analysts. There has been research in overall characteristics of Web users (Jansen & Spink, 2005; Park, Bae, & Lee, 2005; Wang, Berry, & Yang, 2003; Wolfram, 1999), as well as methods to analyze these logs effectively and efficiently (Almpanidis, Kotropoulos, & Pitas, 2007; Meghabghab & Kandel, 2004), along with several studies investigating other aspects of Web searching. (For a comprehensive review, see Markey, 2007a, 2007b.) Additionally, Chau, Fang, and Yang (2007) present results from the analysis of the search logs from Timway, a Chinese search engine, reporting that search topics and the mean number of queries per sessions are similar to usage of English search engines. Whittle, Eaglestone, Ford, Gillet, and Madden (2007) have explored new ways to mine value from search logs. Kellar, Hawkey, Inkpen, and Watters (2008) explore augmenting log analysis with other research methods. Machill, Beiler, and Zenker (2008) highlight the need for search engine research along culture and social lines.

Beitzel, Jensen, Chowdhury, Grossman, and Frieder (2004) reviewed a log of hundreds of millions of queries that constituted the total query traffic of a general purpose commercial Web search service. They 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. It is also relevant to the development of enhanced query disambiguation, routing, and caching algorithms.

Yates, Benavides, and Gonźalez (2006) presented a framework for the automatic identification of user interests, based on the analysis of query logs. The researchers found that supervised learning could identify user interests given certain established goals and categories. With unsupervised learning, one can validate the goals and categories used, refine them, and then select those most appropriate to the user's needs.

Fan, Pathak, and Wallace (2006) proposed a representation scheme of nonlinear ranking function and compared this new design to the vector space model. Fan et al. then tested the new representation scheme with the genetic programming-based discovery framework in a personalized search context using a Text Retrieval Conference (TREC) Web corpus.

This line of research is primarily descriptive of current actions, and researchers are beginning to use more robust methodologies to analyze the interactions between users and systems to predict future actions. One of the most challenging problems of building an efficient predictive model of Web search is that search engine transaction logs contain technically discrete time series data. Neural networks are good tools for identifying relationships between inputs and outputs from a set of examples; therefore, neural networks are good candidates for transaction log analysis. Due to the neural networks approximation properties as well as their inherent adaptation features, neural networks have wide application for modeling of nonlinear systems (Giles, Lawrence, & Tsoi, 2001). We were surprised to learn that only a few neural networks have been applied to the analysis of Web search engine logs.

Özmutlu, Spink, and Özmutlu (2004) provided the results from a comprehensive time-based Web study of US-based Excite and Norwegian-based Fast Web search logs, exploring variations in user searching related to the changes in time of the day. The researchers reported that the analysis of the datasets was very useful to Web search engines for reconstructing the search structure and reallocating the resources with respect to different periods.

In a follow-up of this research, Özmutlu, Seda, and Çavdur (2005) analyzed contextual information in search engine AQ3

DOI: 10.1002/asi

query logs. The study proposed a topic identification algorithm using artificial neural networks. A sample from an Excite data log was selected to train the neural networks, and then the neural network was used to identify topic changes in the data log. The researchers reported that topic shifts were estimated correctly, with a 77.8% precision in the overall database. Özmutlu, Çavdur, Spink, and Özmutlu (2005) have shown that one can train neural networks using multiple search logs. Özmutlu, Çavdur, and Özmutlu (2008) conducted a cross-validation of an artificial neural network application to automatically identify topic changes in Web search engine user sessions by using data logs of different Web search engines for training and testing the neural network.

However, these works were focused primarily on classifying past behaviors or query topics. These studies did not provide an efficient model to identify user-system interaction that could predict future user behaviors reliably, especially user clickthrough. An exception is Zhang, Jansen, and Spink (Forthcoming) who explore time-series analysis to predict clickthrough. A primary metric in search engine evaluation for both organic and sponsored links, CTR for a search engine is a critical measure of both system performance and revenue generation. Jansen, Brown, and Resnick (2007) have conducted laboratory investigations of factors influencing clickthrough. Ravid, Bar-Ilan, Baruchson-Arbib, and Rafaeli (2007) explore the relationship between search engine queries and the access pages on Web sites.

To address this gap in current research, we construct a neural network to study user-system interaction and to provide an efficient mechanism to predict user clickthrough. We explore two primary neural networks, applying each network method to a training data set to compare the fitting results of each approach. We follow this by conducting a sensitivity analysis of the input neurons based on the better-fitted neural network method, which is the MLPN, to determine which types of data represented in the transaction log are predictive of users' clickthrough.

Research Question

Specifically, we ask, which user-search engine interaction factors are correlated with future clickthrough?

> From a practical point of view, lots of information included in the transaction log may or may not impact the user's clickthrough. Therefore, we want to find the potential factors that will predict increased or decreased clickthrough of a user so that the search engine companies can determine more efficient methods to optimize the CTR.

Methodology

Neural networks are powerful data modeling tools that are able to capture and represent complex relationships between input and output. Neural networks are complex, nonlinear, distributed systems, and, consequently, they have broad application. Many remarkable properties of neural networks result from their origins as biological information processing cells. Neural networks are especially useful for open loop and closed loop feedback control, which make these especially useful for our application with search log data. Log data is not normal; therefore, the standard statistical methods such as regression may not be effective.

In the open loop application, neural networks serve as classification, pattern recognition, or function approximation. To perform any of these functions, however, one must train the neural networks, and a widely used training technique for neural networks is backpropagation error algorithm. This training technique involves a forward pass to compute responses corresponding to the input patterns followed by a backward pass to adjust the synaptic weights. Both passes are repeated until the actual responses of the network match the desired ones (Kampolis, Karangelos, & Giannakoglou, 2004). Feedforward networks are memory-less in the sense that their response to an input is independent of the previous network state.

Unlike open loop neural networks, closed loop neural networks are dynamic systems. When a new input pattern is presented, neuron outputs will be computed. Because of the feedback paths, the inputs to each neuron are modified, which leads the network to enter a new state. Consequently, different network architectures require different learning algorithms. For this project, we use open loop feedforward neural networks because transaction log analysis conforms most closely to pattern recognition.

Since the purpose of this research is to explore the behaviors of online users and to discover which information shown in the transaction log influences and predicts the future clickthrough, we designed two primary open loop neural networks, and after tuning the networks, analyzed the weights of each input element. Knowing how specific types of information impact clickthrough will allow commercial search engine companies to leverage the user-system interactive data to design more efficient searching algorithms to increase clickthrough. After evaluating the two types of neural networks, we use MLPN because it was the better performing network.

MLPN

In our study, a MLPN is a network with multiple layers using back propagation algorithm to tune the weights. Generally, backpropagation algorithm in the multilayer feed forward network is enough to perform the system identification and has had a wide application in different areas. Therefore, in this section, we will introduce its basic structure and provide the pseudo-code used to train the network for the transaction log data.

Basic structure. MLPNs often have one or more hidden layers followed by an output layer of linear neurons. Multiple layers of neurons with nonlinear transfer functions (i.e., sigmoid nonlinearity function) allow the network to learn nonlinear as well as linear relationships between input and

DOI: 10.1002/asi

output vectors. Generally, such networks are trained more efficiently with standardized data. In this research, we use normalized input and target data as the training and testing sample, and we use sigmoid transfer function as the activation function to constrain the output from hidden layers of the network within the range from 0 to 1.

There is no clear way of determining how many hidden neurons and layers are necessary to form a decision region that is sufficiently complex to satisfy the demands of a given problem. Thus, parameters required are best determined based on experimentation. For the current project, we designed a neural network with a flexible number of layers to filter out nonlinear relationships as much as possible. After building the network structure, we also designed a learning algorithm to fit the desired output. Figure 1 shows the detailed structure of an MLPN.

Training algorithm. Training the data set for MLPN comprises two parts: forwarding the network and backpropagating. Forwarding the network means that all outputs are computed using sigmoid thresholds of the inner product of the corresponding weight and input vectors. Backpropagating the network entails transmitting errors backwards through the network by apportioning them to each unit according to the portion of the error for which the units are responsible. In this research, we use the DELTA backpropagation method to train the neural network.

Because numerous textbooks and papers have illustrated the basic algorithms of backpropagating the network (c.f., Haykin, 1999), here we simply list the notation of variables and algorithms used to train the network.

Variable Notation:

$\vec{x}_j^{\,l}$: Input vector for unit j in layer l. The input from unit i in layer l to unit j in layer $l+1$ could be denoted by x_{ji}^{l+1}.

$\vec{w}_j^{\,l}$: Weight vector for unit j in layer l. The weight between unit i in layer l and unit j in layer $l+1$ could be denoted by w_{ji}^{l+1}. In addition, we use $w_{ji}^{adj^l}(t)$ to represent the adjusted weight at the t^{th} iteration.

$\vec{z}_j^{\,l}$: Weighted sum of the inputs for unit j in layer l.

$\vec{o}_j^{\,l}$: Output vector for unit j in layer l.

$\vec{t}_j$: Target vector for unit j in the output layer.

η: Learning rate, in this study, $\eta = 0.25$.

n_l: Number of units in layer l.

Bias: The bias for threshold function in each layer.

α: Momentum, which means the proportion of previous adjusted weight needed to adjust the current weight for the whole neural network. To increase the learning rate without leading to oscillations, Rumelhart, Hinton, and Williams (1986) suggested a modification to generalized delta to include a momentum term. In our study, $\alpha = 0.9$.

δ_j^l: First partial derivative of sum square error w.r.t the input of each unit, $\delta_j^l = \frac{\partial E}{\partial z_j^l} = -(t_j^l - o_j^l)(1 - o_j^l)o_j^l$.

Gain: Proportion of δ needed to tune the neural network.

Data Set:

We have three data sets with which to construct the neural network:

1. Training sample: $\langle \vec{x}_j^{\,l}, \vec{t}_j^{\,l} \rangle$.
2. Testing sample: $\langle \vec{x}_j^{\,\prime l}, \vec{t}_j^{\,\prime l} \rangle$.
3. Evaluating sample: $\langle \vec{x}_j^{\,\prime\prime l}, \vec{t}_j^{\,\prime\prime l} \rangle$.

Training, Testing, and Evaluating Algorithm:

1. Normalize the input and target value into the range of lower and upper limit (i.e., 0.1 and 0.9).
2. Generate a feedforward network through all the layers (see Figure 1):
   a. Input the instance $\vec{x}_j^{\,l}$, and calculate the weighted sum of inputs and weights $\vec{z}_j^{\,l} = \vec{w}_j^{\,l} \cdot \vec{x}_j^{\,l}$.
   b. Put the weighted sum into the sigmoid activation function, and get the output from each layer: $\vec{o}_j^{\,l} = \frac{1}{1+e^{gain \times z_j^l}}$. Regard the output of layer l as the input of layer $l+1$.
3. Initialize all the weights to small random values (e.g., between -0.5 and 0.5).
4. Use DELTA backpropagation method to train the network backwards using calculated error between target and output in each layer until the termination condition is met.

   For each training sample $\langle \vec{x}_j^{\,l}, \vec{t}_j^{\,l} \rangle$ that could be randomly picked through the training data set:
   a. Based on the feedforward neural network constructed in step 2, calculate the error of the output

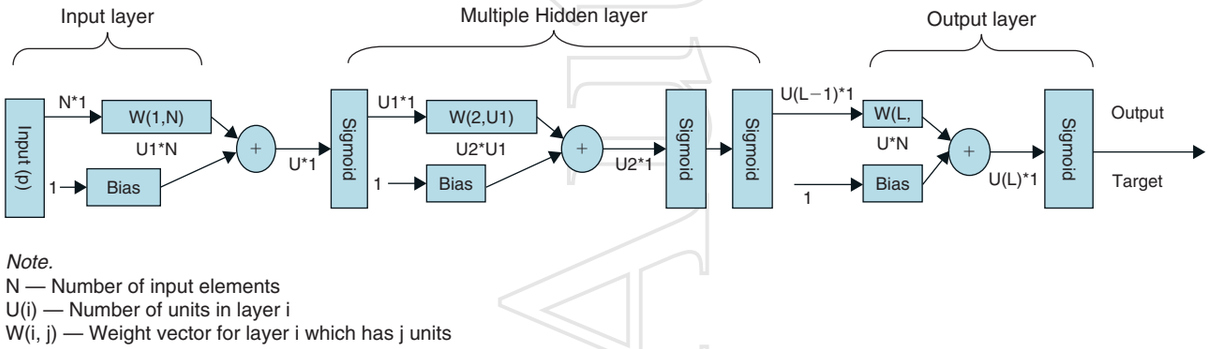

FIG. 1. MLPN structure.

DOI: 10.1002/asi

layer units $\delta^l_j = gain \times (t^l_j - o^l_j)(1 - o^l_j)o^l_j$. At the same time, we calculate the train error of the whole neural network as $\sum_j gain \times (t^l_j - o^l_j)(1 - o^l_j)o^l_j$.

b. Calculate the error of the units in each hidden layer $\delta^l_j = gain \times o^l_j(1 - o^l_j) \sum w^l_{ji}\delta^{l+1}_i$.

c. Update neural network weight w^l_{ji} for each unit i at each layer l as follows: $w^l_{ji}(n) = w^l_{ji}(n) + \eta\delta^l_j o^{l-1}_i + \alpha w^{adj^l}_{ji}(n-1)$, here $w^l_{ji}(n)$ means the weight connecting neuron i and neuron j at the n^{th} iteration, and $w^{adj^l}_{ji}(n)$ means the adjusted weight at the $(n-1)^{th}$ iteration.

For each testing sample $\langle \vec{x}'^l_j, \vec{t}'^l_j \rangle$,

(1) Use the constructed network and testing sample to calculate the output error for the entire testing sample:

$$\text{testing error} = \sum_j gain \times (t'^l_j - o'^l_j)(1 - o'^l_j)o'^l_j.$$

(2) Use the minimum testing error variable to restore the minimum testing error found at each iteration.

Termination condition:

If testing error is less than β* minimum testing error (i.e., $\beta = 1.2$), terminate the training process.

5. For each evaluating sample $\langle \vec{x}''^l_j, \vec{t}''^l_j \rangle$, evaluate the network.

MLPN Compared to RBFN

We also explored another neural network for transaction log analysis. The Radial Basis Function Neural Network (RBFN) is an alternative to highly nonlinearity-in-the-parameters neural network (Park & Sandberg, 1991), which means the determinants of neural centers have high nonlinearity. Traditionally, the RBFN method has been used for strict interpolation in multidimensional space. The original RBFN method requires that there be as many radial basis function centers as data points.

We continuously trained both neural networks until the termination condition was satisfied, namely, the current iteration's error for testing data set was greater than 1.2 times the previous iteration's error. For the MLPN method, the parameters of the number of hidden layers and hidden neurons required were selected based on the experiments. After testing the network several times, we chose two hidden layers with four and six hidden neurons, respectively. In this study, one iteration means training the network using 3,000 pieces of training data, which equals the number of epochs (10 in this study) times the number of records in the training data set (300 in this study).

Figures 2 and 3 show that the training error for the MLPN starts at about 0.9 and is close to 0.2 after iteration 29, while the training error for the RBFN starts at about 0.15 and shrinks almost to 0.05 after iteration 17. This phenomenon is explainable according to the training characteristics of the MLPN and the RBFN. The MLPN uses differentiable and continuous activation functions within hidden layers to screen out the nonlinear behaviors and to tune weights, while the RBFN uses linear output layer to tune the weights after ruling out all the nonlinear behaviors using clustered centers.

Therefore, the RBFN deals with less random and irregular nonlinear data than the MLPN does. For this reason,

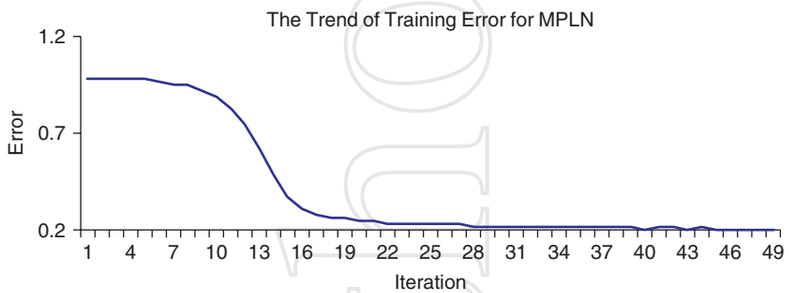

FIG. 2. Training Error for MLPN.

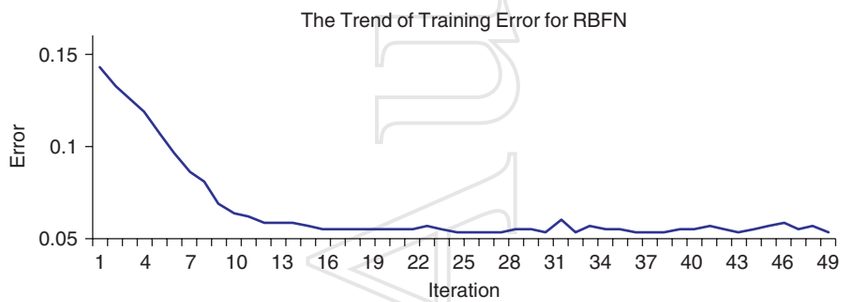

FIG. 3. Training Error for RBFN.

DOI: 10.1002/asi

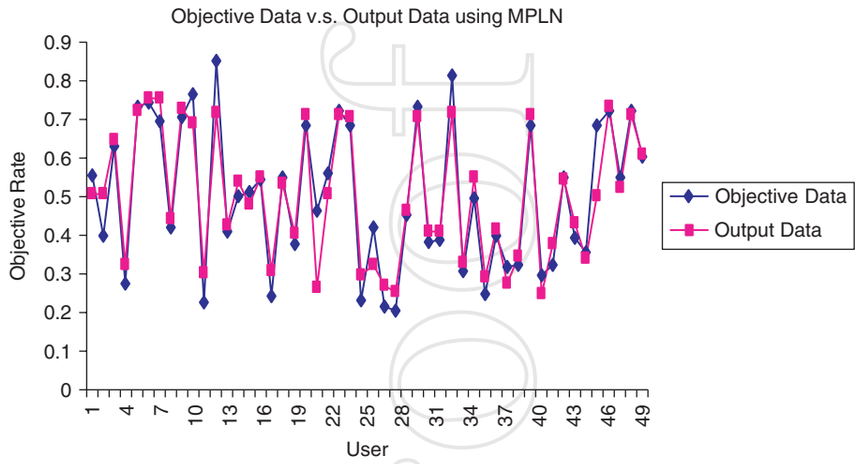

FIG. 4. Training Error for MLPN.

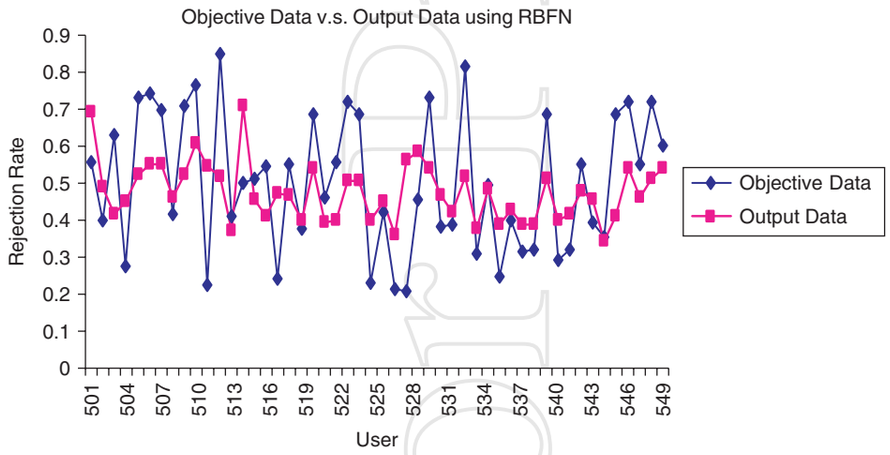

FIG. 5. Training Error for RBFN.

the RBFN could begin to train the network with a lower training error and terminate the iteration earlier. Additionally, between iteration 7 and iteration 17, the training error for the MLPN drops dramatically, while the error does not change much for the other parts. As for the RBFN, the training error maintains the same slope.

Although the training error of the RBFN is much smaller than that of the MLPN, we could not say that the RBFN performs better than the MLPN because each calculates errors based on different input data sets. The error for the RBFN is based on the output coming out of the hidden layer, which has been screened of some nonlinear behaviors, thereby creating data that is more aggregated. We use the evaluation data set from the transaction log to test the fitting of the curve between the output data and objectives to see which neural network worked better using the transaction log.

Figures 4 and 5 show the fitting curves for the MLPN and the RBFN using the same evaluating data set. We can see that the MLPN performs much better than the RBFN in the fitting curves. In other words, the RBFN hidden layer cannot filter out the nonlinear behaviors as well as the MLPN hidden layer does. From a practical point of view, different users will have different searching styles, which is possibly the primary cause of high nonlinearity in the data set.

Because the MLPN behaves much better than the RBFN does, in the rest of this study, we focus only on the sensitivity analysis of the input neurons based on the MLPN.

Data Analysis

AQ4

Data Analysis

In this study, we used a Dogpile (www.dogpile.com) search engine transaction log. Owned by Info space, Dogpile is a market leader in the meta-search engine business, incorporating into its search results the listing from other search engines, including results from the four leading Web search indices (i.e., Ask Jeeves, Google, MSN, and Yahoo!). When accepting a submitted query, Dogpile simultaneously sends the query to multiple Web search engines,

DOI: 10.1002/asi

TABLE 1. Fields in the transaction log.

Field	Description
Record number	A unique identifier for the record. A record is a single tuple in the database. A record is the log of an interaction between the user and the search engine. An interaction is one of the following actions: submit a query, click on a link, or view a results page.
IP address	The Internet protocol (IP) address of the computer on which the user was logged on during the searching session.
Cookie	Parcels of text sent by a server to a Web browser and then sent back unchanged by the browser each time the browser accesses that server. Cookies are used for authenticating, tracking, and maintaining specific information about users, such as site preferences and the contents of their electronic shopping carts.
Time	The time when an interaction was recorded by the search engine server.
Query	The terms of the queries that the user typed into the search engine text box when searching.
Vertical	There are five types of verticals (Web, Audio, Image, Video, News) representing different content collections. They are represented by tabs on the search engine interface and provide a convenience for the users to find different information in different formats.
Sponsored	One of two possible types of links retrieved and presented on the search engine results page (SERP). Sponsored links appear because a company, organization, or individual purchased the keywords that the users used in the search query. If a user clicked a sponsored link, then this field will show 1. Otherwise, the field shows 0.
Organic	The other type of link retrieved and presented on the SERP. These links are retrieved by search engine using its proprietary matching algorithm. If the user clicked an organic link, then this field will show 1. Otherwise, the field shows 0.
Browser	The type of browser used by the users.
Location	The place/country where a user used the search engine as determined by the IP address.

TABLE 2. Additional calculated fields in the transaction log.

Field	Description
User intent	There are three categories of user intent that we calculated, which are *informational*, *transactional*, and *navigational* that reflected the type of user desired content. For this process, we select a sample of records containing not only the query but also other attributes, such as the order of the query in the session, query length, result page, and vertical, and then manually classified the queries in one of three categories, which is derived from work in Rose and Levinson (2004) using an algorithm developed by Jansen, Booth, and Spink (2008).
Query length	The number of terms contained in a particular query.
Results page	A number representing the search engine results page (SERP) viewed (blank is first page, 1 is second page, etc.) during a given interaction.
Reformulation pattern	There are nine categories of query reformulation. We used the algorithm outlined in Jansen, Zhang, and Spink (2007) to classify the queries.

collects the results from each Web search engine, removes duplicates results, and aggregates the remaining results into a combined ranked listing using a proprietary algorithm. Dogpile has tabbed indexes for federated searching of *Web*, *Images*, *Audio*, and *Video* content. Dogpile also offers query reformulation assistance with query suggestions listed in an "Are You Looking for?" section of the interface.

The Dogpile transaction log contains 4,193,956 records from May 15th, 2006. Table 1 shows the fields included in this log.

We also calculated four additional attributes for each record, presented in Table 2.

We define our terminology similar to that used in other Web transaction log studies (Park et al., 2005).

- *Term:* a series of characters separated by white space or other separator
- *Query:* string of terms submitted by a searcher in a given instance
- *Query length:* the number of terms in the query. (Note: this includes traditional stop words
- *Session:* series of queries submitted by a user during one interaction with the Web search engine
- *Sessions Duration*: the period from the time of the first interactions and the time of the last interaction by a searcher interacting with a search engine

To begin the clickthrough analysis, we perform some basic indexing and calculation based on the records in the log and then select several potential inputs for the neural networks. Because clickthrough is based on each user, we group the records according to each unique IP (Internet protocol) address and cookie to determine a single user. Based on the records for each single user, we use the database SQL selection method to retrieve the information necessary to generate numerically formatted training, testing, and evaluating samples. Table 3 shows arranged factors used to train the neural network.

Typically, CTR is simply the number of clicked links divided by the total number of links for an individual query. However, we could not use this formula for two reasons. One, we did not have the total number of links presented to the user

DOI: 10.1002/asi

TABLE 3. Additional calculated data for training the network.

Field	Description
Number of records	The number of records representing a single user.
Average query length	Average query length typed by a single user.
Occurrence of user intent	We calculate the total number of users for each user intent (*informational*, *transactional*, and *navigational*) and then calculate the occurrence by dividing the number of records for each user intent by the total number of users.
Occurrence of browser	We calculate the total number of users for each browser (Firefox, Mozilla, MSIE, etc) and then calculate the occurrence by dividing the number of records for each browser by the total number of users.
Occurrence of vertical type	We calculate the total number of users for each vertical type (Web, Audio, News, Images, Video) and then calculate the occurrence by dividing the number of records for each vertical type by the total number of users.
Mean number of clicked organic results	We calculate the total number of records representing opened organic links and then calculate the occurrence by dividing the number of records for opened organic links by the total number of records for each user.
Average number of clicked non-first pages	We calculate the total number of records representing opened non-first pages and then calculate the occurrence by dividing number of the record for opened non-first pages by the total number of records for each user.
Average rank	The average rank of the links opened by each user.
Reformulation rate	The number of times when the user changed the queries.
Log-in time	The log in time for each user as recorded by the first interaction of the user on the search engine.
Log-out time	The log out time for each user as recorded by the last interaction of the user on the search engine.
Session duration	The time frame spent by each user (i.e., equals *log out time – log in time*.
Rejection rate	= alpha ∗ (reformulation rate / number of record) + beta ∗ (non-clicked number / number of record) + gamma ∗ (average rank), where alpha, beta and gamma are decimal fractional factors between 0 and 1. In this study, alpha = 1, beta = 0.5, gamma = 0.01.
Clickthrough	= 1 – Rejection Rate

in response to a query. Two, we were interested in session-level data (i.e., the collection of all queries submitted by a user during a session). Therefore, we used a more sophisticated formula to calculate the user's likelihood of clicking a link by first calculating a rejection rate (i.e., the propensity of a user to not click on the results). Then, we calculated clickthrough by taking the inverse of the rejection rate.

Data and Factors

We grouped the records according to each unique IP address and cookie to determine a single session. For the Dogpile data set, we had information on hundreds of thousands of users. However, this huge data set is not necessary to train the neutral networks because the principle of training is about how to use insufficient data to get necessary relationships between inputs and outputs. If we use all the records to do the training process, the construction of neutral networks will be meaningless.

Therefore, a smaller randomly selected data subset is appropriate to determine the characteristics and performance of the neutral networks while training the transaction log data set. Considering the computational time and efficiency of the network, we used 550 randomly selected user sessions as the training, testing, and evaluating data. We used the first 300 sessions as the training sample, and the next 200 sessions as the testing data, and the final 50 sessions as the evaluation sample.

We then used a larger data set to complete the final analysis, dividing this set into 30 groups totaling 16,383 interactions, 29 groups, with each group containing 550 users' session data, and one group containing 433 interactions. Because the last group was incomplete, we did not use it to tune the neutral networks, giving us 15,950 records for the analysis. AQ5

We selected nine types of information as the input to the neutral networks, with clickthrough will be regarded as the desired output (see Table 4).

Results and Implications

To get an understanding of the entire log, we first used Matlab connected to a SQL server to perform a temporal analysis of the dataset. We had to ensure that each time unit we analyzed had equal time buckets so that the length of the time slot would not affect statistical data. We divided the 4,193,956 records from Dogpile into 1,080 equidistance groups. The basic statistical analyses for this data set follow.

For the Dogpile daily transaction log, data is based on a 24-hour daily transaction log (from 00:00 to 24:00). Figure 6 shows that the number of records within each group. As one can see, the population flow goes up during the daytime and drops during the night. If we regard the extreme data that occurred at about the 70th and 270th buckets as aberrant or resulting from abnormal behaviors, such as some people continuously and maliciously logging in and out of a search engine within a short period, then we get the same results as other studies. According to the time-based analysis on Excite and Fast search engine logs studied by Özmutlu et al. (2004), the decrease in the queries per session indicates that Web search engine users might spend less effort on retrieving their

DOI: 10.1002/asi

TABLE 4. Inputs of the neural network.

Factor #	Factor name	Description
Factor 1	Number of records	The number of records from a single user, which is a count of the number of interactions between the user and the search engine.
Factor 2	Sum of rank	The total rank of links opened by each user. This was a measure of both the number of links opened and how far into the results listing the user went.
Factor 3	Mean number of organic Links clicked	We calculate the total number of records representing the opened organic links and then calculate the rate by dividing the number of opened organic links by the total number of records for each user.
Factor 4	Mean query length	Average query length measures in terms submitted by the user.
Factor 5	Type of browser	We calculate the total number of users for each browser (Firefox, Mozilla, MSIE, etc.) and then calculate the rate by dividing the number of records for each user intent by the total number of users.
Factor 6	Rate of vertical type	We calculate the total number of users for each vertical type (Web, Audio, News, Images, and Video) and then calculate the rate by dividing number of the records for each vertical type by the total number of users.
Factor 7	User intent rate	We first calculate the total number of users for each user intent (*informational*, *transactional*, and *navigational*) and then calculate the rate by dividing the number of records for each user intent by the total number of users.
Factor 8	Log in time	The log in time for each user.
Factor 9	Time range	The time frame spent by each user = log out time − log in time.

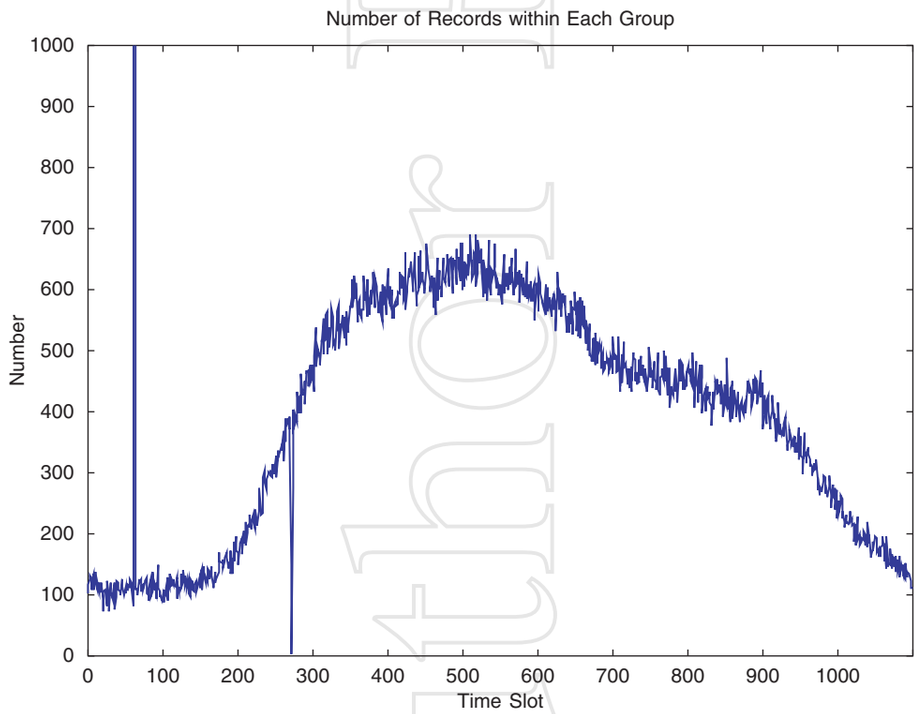

FIG. 6. Number of records within each time slot for daily data set (Population Flow).

information needs later in the day. Given that our results were similar to Özmutlu et al. (2004), this reassured us that our data had external validity.

We can also analyze the popularity of different browsers by calculating the number of browsers used within each time slot. From Figure 7, we can see that most people prefer to use Internet Explorer (IE) browser compared with Firefox, Mozilla, and other browsers. Furthermore, the rate of use for Firefox, Mozilla and other browsers is rarely affected by time.

Similarly, Figure 8 shows our analysis of the types of verticals searched by the users. People seem to prefer to use content in the Web vertical rather than images, video, and audios. Again, this was unaffected by time.

DOI: 10.1002/asi

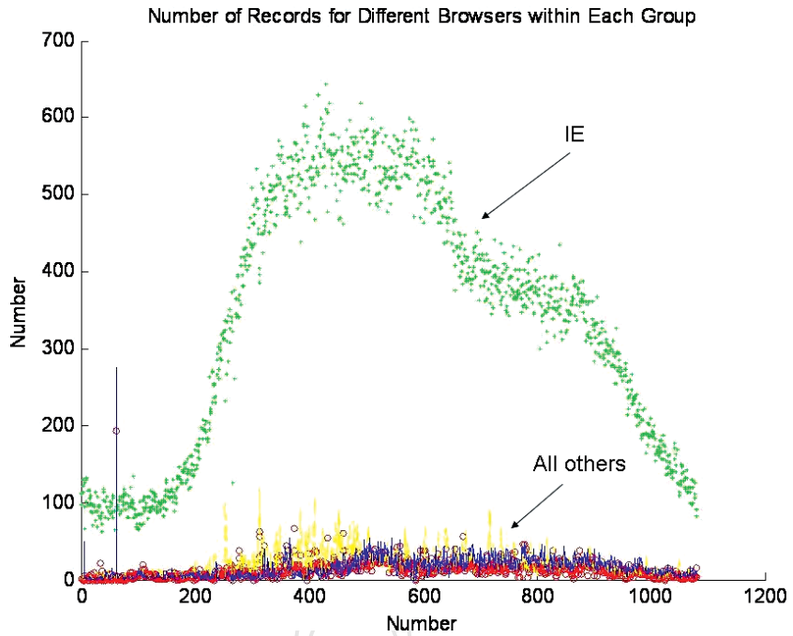

FIG. 7. Number of records for different types of browsers within each time slot for daily data set (blue, Firefox; green, MSIE; red, Mozilla; yellow, other browsers).

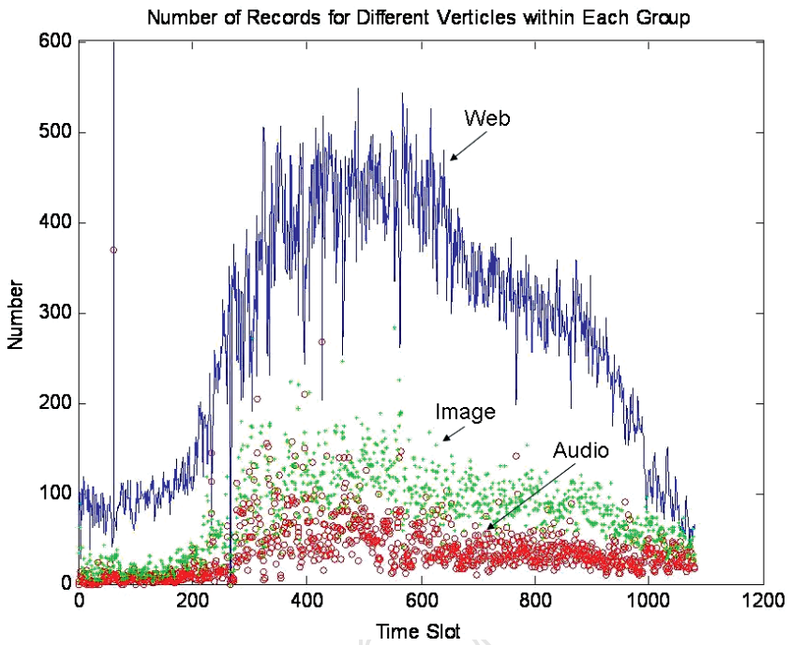

FIG. 8. Number of records for different types of vertical within each time slot for daily data set (blue, Web; green, image; red, audio).

Figure 9 shows which type of information searchers are looking for during the different periods. Most of the searching is *informational*. Pages containing *transactional* and *navigational* content take represent relatively small proportion of searches.

From this basis analysis, we get an idea of how transaction logs analysis picture the behaviors of the online users over time. We know that users spend less effort on retrieving their information needs later in the day. Most people prefer using the IE browser and search for *informational* content using the Web vertical. These descriptive tidbits, however, cannot explain the potential interactions between online users and search engines. Moreover, the results cannot generate an efficient method to explore the potential for improving

DOI: 10.1002/asi

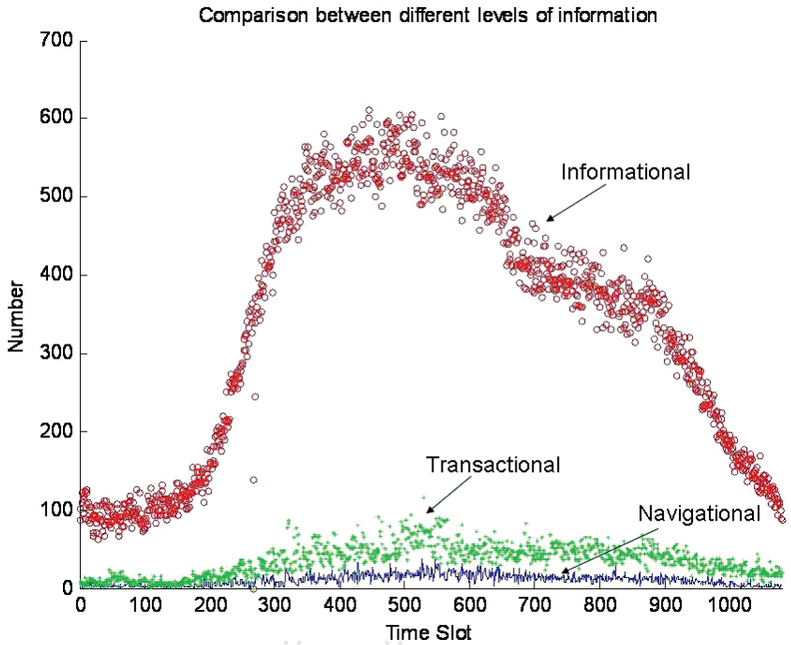

FIG. 9. Number of records for different types of information level within each time slot for daily data set (blue, navigational; green, transactional; red, informational).

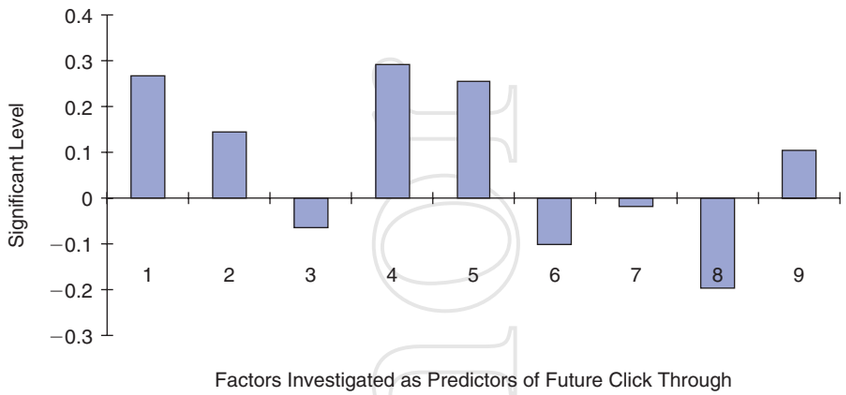

FIG. 10. Final analysis of factors affecting future clickthrough.

the efficiency and accuracy of search engines by predicting clickthrough.

To address such shortcomings, we use a more complicated quantitative analysis method to perform system identification and to discover what information influences the clickthrough. We now report the results of the clickthrough analysis using neural network analysis. Again, we employed Matlab connected an SQL server.

Figure 10 is a graphic of the analysis results, providing a picture of the relative impact on clickthrough for each of the input factors.

Overall, we can group the nine factors into three general classifications.

First, there are five factors that have a significant and positive effect on future clickthrough (i.e., correlate with higher clickthrough). These are factors 1 (*Number of Interactions*), 2 (*Sum of Rank*), 4 (*Average Query Length*), 5 (*Browser Type Rate*), and 9 (*Time Range*).

Second, there are three factors that have a negative effect on future clickthrough (i.e., correlate with lower clickthrough). These factors are 3 (*Mean Number of Organic Links*), 6 (*Vertical Type Rate*), and 8 (*Time of First Query*).

Finally, there is one factor that does not have significant impact on future clickthrough. This factor is 7 (*User Intent Type*). The type of user intent does not influence clickthrough. Table 5 provides a clear idea of how each input impacts clickthrough.

From a practical point of view, the more that a user reformulates the initial query, the clickthrough will increase, although there may be individual queries where the user

DOI: 10.1002/asi

TABLE 5. Sensitivity analysis of the clickthrough.

Factor #	Factor name	Effect on clickthrough
Input 1	Number of records	Has a positive effect on clickthrough. The clickthrough will increase as the number of user interactions with the system increase. This makes sense given that a user who submitted multiple queries or views multiple SERPs will have the opportunity to click on more links.
Input 2	Sum of rank	Has a positive effect on clickthrough. If a user clicks on links further down in the results listing, that user is more likely to click on more links.
Input 3	Mean number of organic links clicked	Has a negative, although slight, effect on clickthrough. Basically, if a user clicks on a lot of clicks now they will click on fewer links in the future.
Input 4	Mean query length	Has a positive effect on clickthrough. The clickthrough will increase the longer the user's query.
Input 5	Type of browser	Has a positive effect on clickthrough. The clickthrough will be higher if the user uses IE relative to other browsers.
Input 6	Rate of vertical type	Has a negative, although slight, effect on clickthrough. If the user searches in a vertical other than Web, the clickthrough will decrease.
Input 7	User intent rate	Does not have much impact on clickthrough. So, regardless of whether the user intent is *informational*, *navigational*, or *transactional*, the clickthrough is generally unchanged.
Input 8	Log in time	Has a negative effect on clickthrough. The clickthrough will increase with the users who start earlier in the day and decrease with those users who logon later in the day.
Input 9	Time range	Has a positive effect on clickthrough. The clickthrough will increase as the duration of a user's stay increases.

clicks on no links. These sessions are probably indicative of more exploratory searching tasks where the user does not have a definitive need and is using the search engine results to help in reformulating the queries.

The next two factors (sum of rank clicked and mean number of organic links clicked) are generally inline with what one would expect. As a user clicks on links further down in the results listings, clickthrough will increase. This is probably due to the query not adequately representing the user's need. There is a negative effect with mean number of organic clicks, but the effect was slight. If a user clicks a lot of links on given query, it would seem that the query probably represents the need and the user has a lot of possible relevant results. Therefore, future clickthrough will decrease.

For mean query length, longer queries are correlated with higher rates of clickthrough. Longer queries are often associated with more specific information needs. Therefore, these users may be receptive to viewing more results. Examples are users that are in the purchase phase of the buying funnel and are keying in on particular products or prices.

We found it interestingly that users using IE browsers may accept more results than those who use other browsers. It would take other methods than log analysis to determine why this is so. Given the limited number of inputs concerning users available from transaction log data, insight from behavioral characteristics, such as browser choices, could be a beneficial approach for gaining additional insight concerning users.

Users who searched in the Web vertical had higher clickthrough than users searching in Audio, Images, or Video. Though, this may be due to the large number of users who searched in the Web vertical relative to the others.

There was no change in clickthrough based on user intent (i.e., the clickthrough was practically the same whether the users were looking for *informational*, *navigational*, or *transactional* content).

Users who searched early in the day had higher clickthrough rates than those who searched later in the day. Why this is so would require further research; however, it is an interesting result with implications for online marketing. Ads appearing earlier are potentially more valuable than ads appear later in the day.

The longer the users searched, the higher the clickthrough. This would make sense as users who stay longer submit more queries, therefore having more opportunity to click on more links.

From a review of all nine factors, how would we describe a user that has the most likely potential for high clickthrough? This user would logon early in the day using the IE browser. The user would submit a query greater than average in length, modify this query several times, and interact with the search engine for a longer than average duration. Most of the searching would be in the Web vertical.

According to these interesting findings, search engines could take steps to increase potential clickthrough. Because there is a greater probability of increased clickthrough earlier in the day, those buying and selling Web advertisements may recognize this as prime online marketing time. Users still use search engines primarily as information systems, so product advertisers can target these users and not just the noted commercial shoppers. Finally, by detecting users with potentially high clickthrough rates, search engines would service these users a tailored SERP.

Conclusion and Future Work

In this article, we focused on how neural networks can be useful in examining Web search engines' transaction logs to develop predictive factors of future clickthrough rates. In some sense, this study is a first step in using neural networks in the analysis of user-system interactions for Web search

DOI: 10.1002/asi

data. This research explores the online behaviors of users so that commercial search engine companies can utilize the user-system interaction data contained in transaction logs to improve clickthrough by designing more efficient retrieval and ranking algorithms.

For the extended data analysis based on the neural networks methodologies, we designed two neural networks (RBFN and MLPN), the characteristics and qualities of which are compared by screening out the nonlinear behaviors with reasonable explanations. The results show that the neural networks perform well in handling large data sets, especially for data sets having huge unpredictable elements and abnormal behaviors. The MLPN, which proved to be more efficient for system identification using the transaction log, was used to detect the significant factors affecting the final response, namely clickthrough. This research is a first step in the field in which the search engine transaction log analysis uses neural networks to analyze search data.

Naturally, there are limitations in the present study. First, we did not consider different training algorithms for both the MLPN and the RBFN and did not analyze the best conditions by changing the initial parameters of the networks. Consequently, we cannot say the MLPN will always perform better than the RBFN in any situation. However, it did perform better with this data. Second, in the extended part of this study part, the clickthrough as well as the selected influencing factors is based only on the data in the transaction log from a single search engine. Therefore, the final results retrieved from our analysis may not represent the patterns shown by users using other types of search engines. However, prior work has shown that searching characteristics are fairly common among different search engines (Jansen & Spink, 2005); therefore, we would expect these results to be applicable to other search engines.

With that said, future work should combine the results from different search engine transaction logs to conduct a comprehensive study based on multiple types of resources. With a wider range of interaction data, especially the number of results retrieved in response to a query, one could explicitly use this approach to predict future clickthrough rates for given queries. Using neural networks on large sample sizes is also an area for future study. An investigation of other variables as input (e.g., other temporal factors) could be a fruitful area of study.

AQ1

References

Almpanidis, G., Kotropoulos, C., & Pitas, I. (2007). Combining text and link analysis for focused crawling—An application for vertical search engines. Information Systems, 32(6), 886–908.

Beitzel, S.M., Jensen, E.C., Chowdhury, A., Grossman, D., & Frieder, O. (2004, 25–29 July). In Hourly analysis of a very large topically categorized Web query log (pp. 321–328). Paper presented at the 27th annual international conference on Research and development in information retrieval, Sheffield, U.K.

Chau, M., Fang, X., & Yang, C.C. (2007). Web searching in Chinese: A study of a search engine in Hong Kong. Journal of the American Society for Information Science and Technology, 58(7), 1044–1054.

Dupret, G., & Piwowarski, B. (2008). In T.-S. Chua & M.-K. Leong (Eds.), A user browsing model to predict search engine click data from past observations (pp. 331–338). Paper presented at the 31st annual international conference on Research and development in information retrieval, Singapore, Singapore.

Fan, W., Pathak, P., & Wallace, L. (2006). Nonlinear ranking function representations in genetic programming-based ranking discovery for personalized search. Decision Support Systems, 42(3), 1338–1349.

Giles, C.L., Lawrence, S., & Tsoi, A.C. (2001). Noisy time series prediction using a recurrent neural network and grammatical inference. Machine Learning, 44(1–2), 161–183.

Haykin, S. (1999). Neural networks—A comprehensive foundation (2nd ed.). New York: Prentice Hall.

Jansen, B.J., Booth, D., & Spink, A. (2008). Determining the informational, navigational, and transactional intent of Web queries. Information Processing & Management, 44(3), 1251–1266.

Jansen, B.J., Brown, A., & Resnick, M. (2007). Factors relating to the decision to click-on a sponsored link. Decision Support Systems, 44(1), 46–59.

Jansen, B.J., & Spink, A. (2005). How are we searching the World Wide Web? A comparison of nine search engine transaction logs. Information Processing & Management, 42(1), 248–263.

Jansen, B.J., Zhang, M., & Spink, A. (2007). Patterns and transitions of query reformulation during Web searching. International Journal of Web Information Systems, 3(4), 328–340.

Joachims, T. (2002). In Optimizing search engines using clickthrough data (pp. 133–142). Paper presented at the 8th ACM SIGKDD international conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada.

Joachims, T., Granka, L., Pan, B., Hembrooke, H., & Gay, G. (2005, 15–19 August). In Accurately interpreting clickthrough data as implicit feedback (pp. 154–161). Paper presented at the 28th annual international ACM SIGIR conference on Research and development in information retrieval, Salvador, Brazil.

Kampolis, I.C., Karangelos, E.I., & Giannakoglou, K.C. (2004). Gradient-assisted radial basis function networks: Theory and applications. Applied Mathematical Modelling, 28(2), 197–209.

Kellar, M., Hawkey, K., Inkpen, K.M., & Watters, C. (2008). Challenges of capturing natural Web-based user behaviors. International Journal of Human-Computer Interaction, 24(4), 385–409.

Machill, M., Beiler, M., & Zenker, M. (2008). Search-engine research: A European-American overview and systematization of an interdisciplinary and international research field. Media Culture & Society, 30(591–608).

Markey, K. (2007a). Twenty-five years of end-user searching, part 1: Research findings Journal of the American Society for Information Science and Technology, 58(8), 1071–1081.

Markey, K. (2007b). Twenty-five years of end-user searching, part 2: Future research directions. Journal of the American Society for Information Science and Technology, 58(8), 1123–1130.

Meghabghab, G., & Kandel, A. (2004). Stochastic simulations of Web search engines: Rbf versus second-order regression models. Information Sciences, 159(1–2), 1–28.

Nettleton, D.F., Calderon, L., & Baeza-Yates, R. (2006). Analysis of Web search engine query and click data from two perspectives: Query session and document, 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006) Philadelphia, PA.

Özmutlu, S., & Cavdur, F. (2005). Neural network applications for automatic new topic identification. Online Information Review, 29(1), 34–53.

AQ2

Özmutlu, H.C., Çavdur, F., & Özmutlu, S. (2008). Cross-validation of neural network applications for automatic new topic identification. Journal of the American Society for Information Science and Technology, 59(3), 339–362.

Özmutlu, H.C., Çavdur, F., Spink, A., & Özmutlu, S. (2005, 31 October–3 November). In Cross validation of neural network applications for automatic new topic identification (pp. 1–10). Paper presented at the

DOI: 10.1002/asi

Association for the American Society of Information Science and Technology (ASIST 2005), Charlotte, NC.

Özmutlu, S., Spink, A., & Özmutlu, H.C. (2004). A day in the life of Web searching: An exploratory study. Information Processing and Management, 40, 319–345.

Park, J., & Sandberg, I. (1991). Universal approximation using radial-basis function networks. Neural Computation, 3(2), 264–257.

Park, S., Bae, H., & Lee, J. (2005). End user searching: A Web log analysis of NAVER, a Korean Web search engine. Library & Information Science Research, 27(2), 203–221.

Ravid, G., Bar-Ilan, J., Baruchson-Arbib, S., & Rafaeli, S. (2007). Popularity and findability through log analysis of search terms and queries: The case of a multilingual public service website. Journal of Information Science, 33(5), 567–583.

Rose, D.E., & Levinson, D. (2004, 17–22 May). In Understanding user goals in Web search (pp. 13–19). Paper presented at the World Wide Web Conference (WWW 2004), New York.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart & J.L. McClelland (Eds.), Parallel distributed processing (pp. 318–362). Cambridge, MA: MIT Press.

Sullivan, D. (2006). Major search engines and directories. Retrieved January 1, 2006, from http://searchenginewatch.com/links/article.php/2156221

Sullivan, D. (2008, February 23). Nielsen / NetRatings search engine ratings. Retrieved 10 March, 2008, from http://www.searchenginewatch.com/reports/netratings.html

Wang, P., Berry, M., & Yang, Y. (2003). Mining longitudinal Web queries: Trends and patterns. Journal of the American Society for Information Science and Technology, 54(8), 743–758.

Whittle, M., Eaglestone, B., Ford, N., Gillet, V.J., & Madden, A. (2007). Data mining of search engine logs. Journal of the American Society for Information Science and Technology, 58(14), 2382–2400.

Wolfram, D. (1999). Term co-occurrence in Internet search engine queries: An analysis of the Excite data set. Canadian Journal of Information and Library Science, 24(2/3), 12–33.

Xue, G.-R., Zeng, H.-J., Chen, Z., Yu, Y., Ma, W.-Y., Xi, W., et al. (2004, 8–13 November). In Optimizing Web search using Web click-through data (pp. 118–126). Paper presented at the Thirteenth ACM conference on Information and knowledge management, Washington, DC.

Yates, R.B., Benavides, L.C., & Gonźalez, C. (2006). The intention behind Web queries. In F. Crestani, P. Ferragina & M. Sanderson (Eds.), Lecture notes in computer science: String processing and information retrieval (spire 2006) (Vol. 4209/2006, pp. 98–109). Glasgow, Scotland: Springer Berlin / Heidelberg.

Zhang, Y., Jansen, B.J., & Spink, A. (Forthcoming). Time series analysis of a Web search engine transaction log. Information Processing & Management.

Author Queries

AQ1 Regarding the Ref. list: (a) Please provide editors for proceedings, if any. (b) Please make sure the links work and are pointing to the page to which you're referring. (c) Zhang- please use "in press" instead of "forthcoming" if it has been accepted for publication. If so, then change it in the text citation as well.

AQ2 This is not cited in the text.

AQ3 This is not recorded in the Ref. list.

AQ4 Please provide an intro paragraph for section 5 or simply delete the "data analysis" as a subheading (which seems rather redundant anyway).

AQ5 Does this mean 30 groups, 29 groups, and one group? If so, please verify if the change to series commas is correct.