Analysis of Web Search Engine Query Sessions

David Nettleton
Web Research Group
University Pompeu Fabra
Passeig de Circumval.lacio, 8
08003 Barcelona, Spain
david.nettleton@upf.edu

Liliana Calderón-Benavides
Web Research Group
University Pompeu Fabra
Passeig de Circumval.lacio, 8
08003 Barcelona, Spain
liliana.calderon@upf.edu

Ricardo Baeza-Yates[1]
Yahoo! Research
Ocata 1
08003 Barcelona, Spain
ricardo@baeza.cl

ABSTRACT

In this paper we process and analyze web search engine query and click data from the perspective of the query session (query + clicked results) conducted by the user. We initially state some hypotheses for possible user types and quality profiles for the user session, based on descriptive variables of the session. The query dataset is preprocessed and analyzed using some traditional statistical methods, and then processed by the Kohonen SOM clustering technique, which we use to produce a two level clustering. The clusters are interpreted in terms of the user type and quality profiles defined initially. Then we apply the C4.5 rule induction algorithm to predict the session quality and the user type, using two month's of click data for training, and testing on data captured during a third consecutive month. The objective of the work is to apply a systematic data mining process to click data, contrasting non-supervised (Kohonen) and supervised (C4.5) methods to cluster and model the data, in order to identify profiles and rules which relate to theoretical user behavior and user session "quality".

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications – *data mining*. H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *clustering, information filtering*. Online Information Services – *web-based services*. I.2.6 [**Artificial Intelligence**]: Learning – *concept learning, connectionism and neural nets, induction.*

General Terms

Algorithms, Measurement, Verification.

Keywords

Web mining, web queries and pages, query-sessions, clicks, clustering, rule induction, user types, quality profiles.

1. INTRODUCTION

Web search log data analysis is a complex data mining problem. This is not essentially due to the data itself, which is not intrinsically complex, and typically comprises of document and query frequencies, hold times, and so on. The complexity arises from the sheer diversity of URL's (documents) which can be found, and of the queries posed by users, many of which are unique. There is also the question of the data volume, which tends to be very large, and requires careful preprocessing and sampling. The analyst may also have the impression that there is a certain random aspect to the searches and corresponding results, given that we are considering a generalist search engine (TodoCL), as opposed to a specialized domain search engine (such as Medline) or a search engine contained within a specific website (for example, in a University campus homepage). Given this scenario, in order to extract meaning from the data, such as user behavior categories, we consider different key elements of the user's activity, such as: (i) the query posed by the user, (ii) the individual documents selected by the user, and (iii) the behavior of the user with respect to the documents presented by the search engine. Recent work, such as that of Ntoulas *et al* [10] has evaluated the predictability of page rank and other aspects in the web over different time periods. They found a significant change in the web over a period of 3 months, affecting page rankings. In [1], Baeza-Yates and Castillo traces the user's path through web site links, relating the user behavior to the connectivity of each site visited. Baeza-Yates *et al* [2] evaluates different schemes for modeling user behavior, including Markov Chains. In [9], Nettleton *et al* proposes different techniques for clustering of queries and their results. Also, Sugiyama *et al* [12] has evaluated constructing user profiles from past browsing behavior of the user. They required identified users and one day's browsing data. In the current paper, the users are anonymous, and we identify behavior in terms of "query sessions". A query session is defined as one query made by a user to the search engine, together with the results which were clicked on, and some descriptive variables about the user behavior (which results were clicked on, the time the pages are "held" by the user, etc.). Finally, Lee *et al* [6] have developed an approach for the automatic detection of user 'goals' in web search. They used a reduced set of 50 pre-selected queries from which ambiguous queries had been eliminated, to get a set of 30 queries. The results were promising, but were based in a very small set of queries that were biased to computer science and that were not too ambiguous.

Advantages of our approach: in this paper we propose a systematic data mining approach [7, 8] to query results click data, an area which is still relatively new in the web mining field. We also define a novel set of "quality" indicators for the query session, relate them to the clustered data, and create a predictive model using them as the output class. Our approach has the advantage of not requiring a history log of identifiable users, and defines profiles based on information relating to non-unique queries.

Structure of the Paper: in Section 1 we present the hypothetical user type and quality profiles which we propose to identify and predict in the data; in Section 2 we describe the data processing algorithms, Kohonen SOM and C4.5; in Section 3 we describe the data capture and preparation process; in Section 4 we describe the data analysis phase, and in Section 5 we present the data clustering work; finally, Section 6 describes the predictive modeling with C4.5 rule and tree induction, using the user type and quality profiles labels as predictive values. We end with some concluding remarks.

1.1 User Profiles

We can define as hypothesis, three main user search behavior categories defined by Broder [3], which can be validated from the data analysis. We have to add that this classification is very coarse, therefore the real data does not have to exactly fall into these categories. Broder's three categories are: **(i) Navigational:** this user type typically accounts for approx. 25% of all queries. The user is searching for a specific reference actually known by him, and once he finds it, he goes to that place and abandons the query session. For example, a user searches for "white house", finds the corresponding URL reference, and then goes to that reference and conducts no further searches**.** *This user would typically use a lower number of clicks and a minimum hold time* (the time the user takes to note the reference he is looking for). **(ii) Informational:** this type typically accounts for approx. 40% of all queries. The user is looking for information about a certain topic, visiting different Web pages before reaching a satisfactory result. For example, a user searches for "digital camera", finds several references, and checks the prices, specifications, and so on. This user would spend more time browsing (*higher document hold time*) and would make more document selections (*greater number of clicks*). **(iii) Transactional:** this type typically accounts for approx. 35% of all queries. The user wants to do something, such as download a program or a file (mp3, .exe), make a purchase (book, airplane ticket), make a bank transfer, and so on. This *user would make few document selections (clicks) but would have a higher hold time* (on the selected page). We can confirm the transactional nature by identifying the corresponding document page (for example, an on-line shopping web page for purchasing a book, a page for downloading a software program, etc.). In this paper, we are interested in applying a methodological data mining approach to the data, in order to identify profiles and rules, which are related to the three main user types defined by Broder [3], and the "session quality" profiles which we will now present in Section 1.2. Also, we wish to inter-relate the two visions of query-session and document, and identify useful features from the overall perspective by analyzing the resulting clusters and profiles. We propose that this is an effective approach for identifying characteristics in high dimensional datasets.

1.2 Quality of Query Sessions

In this section we define four hypothetical categories that indicate query session "quality", and which will be validated in Sections 5 and 6 from the data analysis. We define two categories to indicate a "high" quality query session, and two categories to indicate a "low" quality session. The quality of the query sessions can be affected on the one hand by the ability of the user, and on the other hand by the effectiveness of the search engine. The search engine is effective when it selects the best possible documents for a given user query. There are other related issues, such as response time and computational cost, although these aspects are out of the scope of the current paper. In the case of user queries, we have chosen some variables which describe the activity: number of search results clicked on; ranking of the search results chosen (clicked) by the user; duration of time for which the user holds a clicked document. From these variables, we can define some initial profiles which can be used to classify (or distinguish) the user sessions in terms of "quality". As a first example, we could say that a good quality session would be one where the user clicks on a few documents which have a high ranking (e.g. in the first five results shown), given that it is reasonable (though not definitive) to assume the ranking of the results is correct with respect to what the user is looking for and has expressed in the corresponding query. With reference to Table 1, this profile corresponds to "$high_1$". Contrastingly, if the user looks a long way down the list of results before clicking on a document, this would imply that the ranking of the results is not so good with respect to the query. Another profile for a good quality query session would be a high hold time, which implies that the user spends a longer time reading/visualizing the clicked document (profile "$high_2$" of Table 1).

Table 1. Hypothetical user query session quality profiles

Profile (quality of query session)	**$high_1$**	**$high_2$**	**low_1**	**low_2**
Average hold time of selected documents		high		low
Ranking of documents chosen	high		low/ medium	
Number of clicks	low		high	high

In the case of low hold times, we cannot assume low quality, because the user may be a "navigational" type, and therefore finds what he wants and leaves the current session. In the case of an "informational" or "transactional" user type, a lower hold time would indicate that the user has not found the content interesting. If we combine this with a high number of clicks, it would indicate that the user has found it necessary to check many results. This profile would correspond to "low_2" of Table 1. If the user selects many low ranking documents this would also identify that the ordering of the results does not correspond well with the query (profile "low_1" of Table 1). In order to distinguish the user

types in this way, we would need to analyze the content of the documents, which is outside the scope of this paper. Therefore, we will limit to quality profiles which can be detected without the need for document content analysis. Table 1 summarizes the key variable value combinations together with an indicator of query session quality. Later, we use these "profiles" to evaluate the query session quality in the clustering results, and as category label for a rule induction predictive model. The corresponding numerical ranges for the "low", "medium" and "high" categories were assigned by inspection of the distribution of each variable, together with consultation with the "domain" expert. The ranges for "low", "medium" and "high", respectively, for each of the variables of Table 1 are as follows: "average hold time for a given query", (0-40, 41-60, >60); "average number of clicks for a given query", (1-2, 3, >3). In the case of "average ranking of documents chosen for a given query", the corresponding labels have an inverse order, that is, "high", "medium" and "low, with corresponding ranges of (1-3, 4-5, >5). These ranges are also used for identifying the "Broder" user search behavior categories, as described previously in Section 1.1.

2. DATA PROCESSING ALGORITHMS

In this section, we briefly present the algorithm steps and the distance measure for the Kohonen SOM clustering algorithm, and the partition algorithm and criteria used by C4.5 rule/tree induction. They represent two techniques with a completely different approach: the SOM accumulates cases at each 'lattice node' starting with the complete dataset and progressively reducing the local areas (neighborhood) of update; on the other hand, C4.5 starts with a small training subset, testing it on the whole dataset, and progressively increases the size of the subset to include more cases, partitioning the cases based on the values of selected input variables. In general, the Kohonen SOM can be used as a first phase of data mining in order to achieve homogeneous clusters from high dimensional data. Then C4.5 can be used to create classifier models for each cluster created by the SOM. This is confirmed by the results we present in Section 6 of this paper, in which C4.5 produces higher accuracy on individual clusters, and lower accuracy given the whole dataset without clustering as input. Also, the Kohonen SOM presents a "machine learning" solution as an alternative to the traditional statistical approach of k-Means clustering, often used for clustering term-document data and queries. We could add that the neural network approach of the SOM is adequate for clustering complex datasets with noise and high dimensionality.

2.1 Kohonen SOM

The Kohonen SOM[5] is a set of processors which organize themselves in an autonomous manner, only requiring the original inputs and an algorithm to propagate changes in the net. The state of the net resides in the weights (coefficients) assigned to the interconnections between the units. It has two layers: layer one contains inputs nodes and layer two contains 'output' nodes. The modifiable weights interconnect the output nodes to the common input nodes, in an extensive manner.

Basic algorithm. The global objective is to move the weights towards the cluster centers via the updating of the weights by each input value.

Step 1: initialize the weight vectors. Can use random assignments or partially trained weights

Step 2: present inputs to the network.

Step 3: determine the weight vector that is closest to the input vector. Search over complete matrix to find the weight vector with the smallest Euclidean distance difference from the input vector. That is, find *i'*, *j'* such that

$$\| \mathbf{v} - \mathbf{w}_{i',j'} \| <= \| \mathbf{v} - \mathbf{w}_{i,j} \| \text{ for all i, j}$$

where $\boldsymbol{v}$ is the input vector and *i* and *j* range over all the nodes in the matrix.

Step 4: weight adaptation. The adaptation is only applied to weight vectors of nodes within a given neighborhood of the node chosen in Step 3. The neighborhood size is one of the setup parameters, and is gradually reduced during the training run. In this manner, node weights which are further away from the node chosen in Step 3 are modified less. A Gaussian function is then applied to the distance of each node weight vector from the chosen node. That is:

$$\mathbf{w}_{i,j}'' = \mathbf{w}_{i,j}' + \varepsilon \exp(\alpha \| \mathbf{v} - \mathbf{w}_{i,j}' \|^2) (\mathbf{v} - \mathbf{w}_{i,j}')$$

Where *v* is the input vector and the range of *i* and *j* range is limited to the neighborhood of the node *i'*, *j'* selected in Step 3. The other parameters, ε is the "step size" and α is a fixed coefficient assigned as the inverse of the neighborhood size.

2.2 C4.5 Decision Tree Algorithm

C4.5 [11] is an induction algorithm which generates rules from subsets (windows) of cases extracted from the complete training set, and evaluates their goodness using criteria based on the precision in classifying the cases. The main heuristics used (see below) are: (i) the information value which a rule provides (or tree branch) calculated by "info" and (ii) the global improvement that a rule/branch causes, calculated by "gain". C4.5 is based on the classic method of 'divide and conquer' [4].

Partition criteria used in generating a tree. An 'information gain' heuristic, in the form of an entropy function, is used in order to decide when to create partitions. It considers the information gain of an attribute *'a'* for a set of cases *T*, and is calculated as follows: if *'a'* is discrete, and $T_1,\ldots,T_s$ are the subsets of *T* consisting of cases with distinct value for attribute *'a'*, then the entropy function will be:

$$\text{info}(T) = -\sum_{i=1}^{\text{NClass}} \frac{\text{freq}(C_i, T)}{|T|} \times \log_2 \left\lfloor \frac{\text{freq}(C_i, T)}{|T|} \right\rfloor$$

and

$$\text{gain} = \text{info}(T) - \sum_{j=1}^{s} \frac{|T_j|}{|T|} \times \text{info}(T_j).$$

where T_j represents data subset *j* and "gain" measures the information obtained by partitioning *T* using attribute *'a'*. Therefore, the 'information gain' heuristic selects an attribute in order to maximize the information obtained.

3. DATA PREPARATION

In this section, we describe the original data used, which is organized in a relational data mart structure. We also describe the preprocessing realized to obtain the dataset from the point of view of the user query.

3.1 Data Mart

In order to conduct the different tests proposed in this paper, we used a set of web search logs, from the Chilean search engine, TodoCl.com, captured over a 92 day period from 20th April to 20th July 2004. The data contained in this log file was pre-processed and stored in a relational data base, which enabled us to carry out different analyses on the search behavior of the users of this search engine. We have a total of 65,282 queries and 122,184 documents available. Before proceeding, we first present some of the concepts used by Baeza-Yates *et al* in [2], necessary to understand the data structures used: (a) a "Query" is a set of one or more keywords that represent a user information need formulated to a search engine. (b) A "Query instance" is a single query submitted to a search engine in a defined point of time. (c) A "Query Session" consists of a sequence of "query instances" by a single user made within a small range of time. (d) A "Click" is a document selection that belongs to a query session. (e) A "document" is an "URL" Internet address reference. The data analyzed was captured by a team from the Center for Web Research (www.cwr.cl), Department of Computer Science, University of Chile [2]. The data mart we use consists of a series of relational tables which hold transactional and descriptive data about the queries made by the users and the documents clicked by the user from the search results presented to him. The "Click" table is the most disaggregated of the tables, and contains one line per click by the user. The URL (document) reference is included, together with the time and date of the click, the time the URL was held on screen by the user (hold time), and the ranking of the URL clicked in the list of URL's found. The "Query" table contains an index to the queries made by the users, including the query terms, number of terms and query frequency. Finally, the "Query Session" table links the "Query" table to the "Click" table, and aggregates the user sessions from the "Click" table.

"A priori" and "a posteriori" data. Often, in data mining, we consider the descriptive variables in terms of two groups: (i) "a priori", which are known before an event occurs (such as the launch of a search query) and (ii) "a posteriori", which are only known after the event has occurred. In the present study, we only have a few "a priori" variables, such as the number of terms and the terms themselves, to describe a user query. On the other hand, we have a significant number of relevant "a posteriori" variables, such as 'hold times for pages selected', 'ranking of pages selected', 'number of clicks', and so on. Therefore, we decided to use both "a priori" and "a posteriori" variables in the predictive model of section 6, but calculated exclusively for the given 'train and 'test' time periods. That is, the 'train' data used 'a posteriori' variables calculated exclusively from the first 2 months of data, and the 'test' data consists of variables calculated exclusively from the 3rd month of data. This is important for the predictive model of Section 6. On the other hand, the unsupervised clustering of Section 5 does not have to be restricted to 'a priori' variables or data, given that it represents the 'exploration phase'. Nevertheless, notice that we can always use historical "a posteriori" data for each query.

3.2 Data Preprocessing

Using the data mart described in Section 3.1 as a starting point, we preprocess to produce a "query" dataset, derived from tables "Query", "Query Session" and "Click". The resulting data structures are shown in Figure 1.

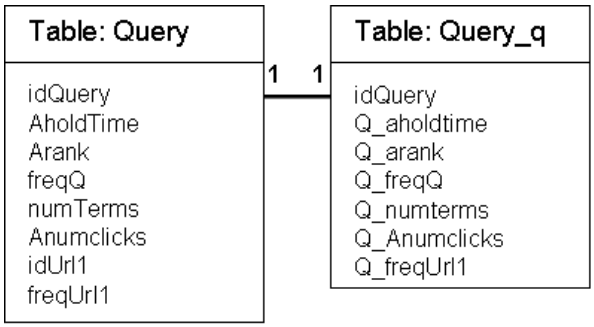

Figure 1. Dataset definition for queries, with associated tables of quantiles for selected variables

With reference to Figure 1, the "Query" table contains a series of statistics and aggregated values for the queries. "Aholdtime" is the average hold time for the URL's clicked which correspond to the query and "Arank" is the average ranking of those URL's. "FreqQ" is the number of times the query has been used (in the click data), and "numTerms" is the number of terms of the query. "Anumclicks" is the average number of clicks made corresponding to the given query in the click table. Finally, "idURL1" represents the URL whose frequency was greatest for the corresponding query (from the click data) and "freqURL1" is its corresponding frequency relative to the given query. These data variables have been selected to create a "profile" of the "query" in terms of the statistical data available, which will serve for posterior analysis of the characteristics of search behavior in terms of the queries.

Finally, with reference to Figure 1, the additional table contains the quantiles of selected variables which modeled the information from the Query table. The quantiles have been generated automatically using the SPSS program, and all the variables have been transformed into 10 quantiles, excepting "numterms" which has four quantiles (this number was chosen by a previous inspection of the distribution and number of values for numterm variable). The quantile versions of the variables were used as inputs to the Kohonen clustering algorithm.

Table 2. Query dataset: Pearson Correlation values for variable quantiles

	Quantiles avg. hold time	Quantiles avg. rank	Quantiles query frequency	Quantiles num. terms	Quantiles avg. num. clicks	Quantiles freq. of URL 1
Quantiles avg. hold time	1.000	**.399**	**.229**	-.061	**.706**	**.309**
Quantiles avg. rank	.399	1.000	.188	-.170	**.461**	.049
Quantiles query	.229	.188	1.000	**-.223**	.202	**.642**
Quantiles num. terms	-.061	-.170	-.223	1.000	-.050	-.173
Quantiles avg. num.	.706	.461	.202	-.050	1.000	.383
Quantiles freq. of URL 1	.309	.049	.642	-.173	.383	1.000

** Number of cases=11981. Significant results indicated in bold.

The quantiles ranges for the query data variables are as follows: "Q_aholdtime" 2(0), 3(1-7), 4(8-17), 5(18-29), 6(30-45), 7(46-48), 8(69-105), 9(106-188), 10(189-16303); "Q_arank" 1(1), 2(2), 3(3), 4(4), 6(5-6), 7(7), 8(8-9), 9(10-13), 10(14-119); "Q_freqQ" 3(2), 7(3), 8(4), 9(5-6), 10(7-284); "Q_numterms" 1(1), 2(2), 3(3), 4(4-12); "Q_Anumclicks" 2(1), 5(2), 7(3), 8(4), 9(5-6), 10(7-80). In the case of the variable "Q_aholdtime", we observe that the last quantile has captured some large non-representative values. In the case of the variable "Q_freqURL1" which was not used in the clustering, but was used for cross referencing across dataset, its quantiles were as follows: 2(1), 6(2), 8(3), 9(4), 10(5-166).

3.3 Data Sampling

Random case data was selected from the query dataset to create a new dataset of 1800 records. This dataset was used as input to the Kohonen SOM. The original dataset consisted of 11981 queries.

The queries selected must have a frequency greater than 1 (occur more than once) in the click data table. The requirement of frequency > 1 for queries and documents, avoids including "once off" or unique queries in the dataset, as these queries tend not to have any interrelations and create a great dispersion. Finally we filtered records whose hold time was greater than 900 seconds, given that this is a reasonable maximum for normal user sessions.

4. DATA 'PRE-ANALYSIS'

In this section we explain the initial analysis of the datasets, using correlation and graphical techniques. We also applied k-Means to perform an initial clustering to confirm the initial hypothesis of their being coherent clusters in the data. In this manner we can identify at this stage any errors in the data or problems due to preprocessing.

With reference to Table 2, we can observe a promising degree of correlation between key variable pairs for the complete query dataset comprising of 11981 different queries. In particular, we can indicate the following correlations: 0.706 for "average hold time" with respect to "average number of clicks"; 0.642 for "frequency of the query" with respect to "frequency of the URL which is recovered most often by this query"; 0.461 for "Average ranking of the documents clicked after running the query" with respect to "average number of clicks made after running the query". In Figure 2 we can see the sector diagrams generated for the quantiles of the key variables in the query dataset. We observe that in the case of "q. freq query" (quantiles of freq. of query), aprox. 55% of the values are in quantile 3, and the second largest proportion is that of quantile 7. For "Avg. number of clicks", quantile 5 has the largest proportion followed by quantile 2. For the correspondences of quantiles to original value ranges, refer to Section 3.2.

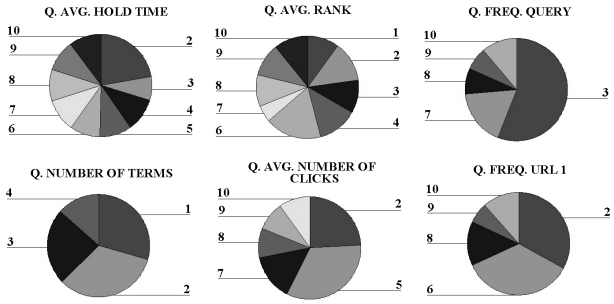

Figure 2. Sector diagrams for selected quantiles of key variables in the query dataset

5. DATA CLUSTERING

In this section, we explain the clustering process applied to the dataset using the Kohonen SOM technique, and the following analysis of the data groupings with respect to the user type and quality profiles defined in Section 1. The input variables to the query clustering were: Q_aholdtime, Q_arank, Q_freqQ, Q_numterms and Q_Anumclicks. See Section 3 for the descriptions of the variables.

Table 3. Kohonen clustering of Queries data: averages of input variables for 'level 1' cluster groups

Queries							Confidence	
Cluster Group*	Avg. number of terms	Avg. query freq.	Avg. hold time	Avg. ranking	Avg. number of clicks	Number of Queries	Avg. activ-ation	Stdev. activation
11	3.16	**2.63**	30.87	4.94	1.92	191	8.77	2.60
12	2.24	3.53	103.66	6.86	1.92	214	11.07	2.91
21	1.94	**6.84**	126.59	6.97	2.58	205	11.73	**3.40**
22	2.51	4.59	125.01	5.70	3.28	306	11.86	3.21
30	1.93	4.34	**128.78**	**9.86**	**6.88**	449	**14.18**	2.82
40	2.04	2.95	**4.16**	**4.42**	**1.00**	189	**6.44**	2.61
50	**3.45**	2.69	69.24	4.53	1.11	153	7.84	**2.35**
60	**1.53**	4.56	111.03	4.73	2.00	89	9.78	2.97

*8 level 1 clusters, 225 level 2 clusters assigned.

5.1 Clustering in Homogeneous Groups

The Kohonen SOM algorithm was used as the clustering method for each dataset, using only the quantile values of the selected variables as inputs. The Kohonen SOM was configured to produce an output lattice of 15 x 15, giving 225 clusters, for each dataset. The cluster quality was verified in a post-processing phase, by inspection of activation values and standard deviations. As recommended by Kohonen in [5], we trained the Kohonen net in two stages: (i) an ordering stage with a wider neighborhood value, a higher learning rate, and a smaller number of iterations; (ii) a convergence stage with a smaller neighborhood value, lower learning rate and greater number of iterations.

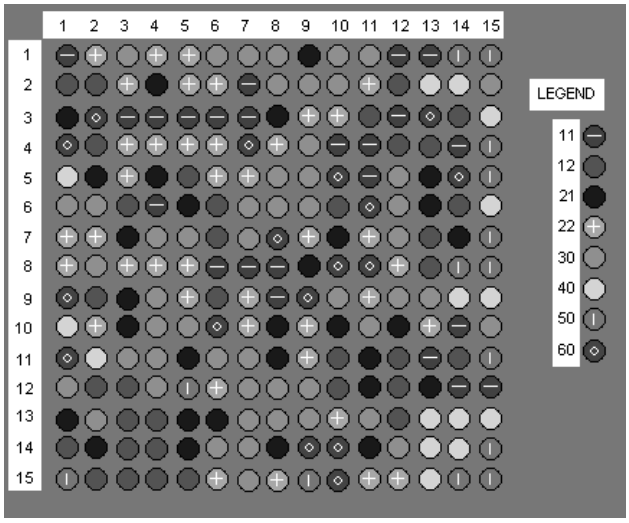

Figure 3. Kohonen SOM clustering for Queries

With reference to Figure 3, we can see the graphic output of the Kohonen SOM at the end of the (convergence stage) training run (5000 iterations). The original Kohonen SOM output was color coded, and we have assigned corresponding cluster id's (column 1 of Table 3) to represent the color scheme. For example, cluster 11 corresponds to dark purple, 12 to light purple, 21 to dark blue, and so on, to 60 which corresponds to red, following the sequence of the spectrum. In this manner we have maintained the information with respect to the sequential ordering of the cluster groups.

All the training runs were run to complete convergence, that is, when the Kohonen SOM no longer altered the clustering assignments and the patterns (see Figure 3) became fixed, which occurred within the 5000 iterations, for both datasets. In Figure 3 (Queries), we observe that the Kohonen SOM has created eight major cluster groups. We can now inspect the corresponding data in each of these cluster groups, in order to identify distinguishing characteristics in terms of the input variables.

Table 3 lists the level 1 cluster groupings generated by the Kohonen SOM 15x15 lattice for the query dataset. Each of the cluster groups consists of a number of lattice nodes (individual clusters), and these are later detailed in Section 5.2. In Table 3, we note that row 1 of the cluster data indicates that cluster group 11 has 191 corresponding queries, and its confidence (activation) value was on average 8.77 with a standard deviation of 2.60. The activation value refers to the neuron activation in the lattice and may be considered as a "quality" or "confidence" indicator for the corresponding cluster. With respect to the variable values, we observe that cluster group 11 has the minimum value (2.63) for "average query frequency" (shown in bold). Cluster group 30 has the maximum values for "average ranking of clicked results" (9.86), "average number of clicks" (6.88), and "average hold time" (128.78). Finally, cluster group 50 has the maximum value for "average number of terms" (3.45), the second lowest value for "average query frequency" (2.69) and the second lowest value for "average number of clicked results" (1.11).

We can say that these values are mutually coherent for cluster group 50, given that less frequent queries would tend to have a higher number of terms and as they are more specific, the user would click on less of the shown results.

Table 4. Average values for key variables used for clustering of query data (corresponding to cluster groups in Table 3), and one comparative variable not used in clustering (Freq of URL 1)

	Average values (for each cluster)						Confidence	Document
Cluster*	Hold time	Ranking	Freq. query	Number of terms	Number of clicks	Number of queries	Avg. activation	Freq. of URL 1
Level 2 Query clusters (for level 1 Cluster Group 12)								
6,7	18	**1.71**	2	2.07	2	14	6,88	1.29
13,4	**4.36**	7.64	2.21	**1.29**	**1.21**	14	6.63	1.21
14,6	8.8	5.5	2.5	1.8	1.4	10	7.45	**1.60**
12,2	31.22	**13.22**	2	2.22	2	9	11.54	**1.11**
5,15	**50.11**	5.78	2	**2.56**	2	9	10.08	1.33
Level 2 Query clusters (for level 1 Cluster Group 30)								
7,14	199.27	**8.82**	2	1.36	**6.59**	22	14.44	2.14
6,11	174.25	8.95	2	**3.15**	6.75	20	14.72	1.90
8,2	**90.71**	**16.18**	2	**1**	**10.88**	17	14.3	**1.71**
12,5	143.88	15.5	**9.44**	1.12	7.31	16	17.21	**4.50**
12,6	**343.12**	13.56	3.12	1.38	7	16	16.71	2.62
Level 2 Query clusters (for level 1 Cluster Group 40)								
13,13	0	2	**2**	**2**	1	31	4	**1.1**
14,13	0	2	2	**3**	1	21	4.48	**1.1**
15,13	0	**1.65**	2	**3**	1	20	4.2	1.4
13,14	0	**3.53**	2	2	1	19	4.8	1.11
1,5	0	1.82	4.76	**1.41**	1	17	7.95	**3.94**

*in descending order of number of queries assigned

5.2 Analysis of Clusters and Sub-Clustering

Once the Kohonen SOM had generated satisfactory clusters, we selected specific cluster groups by observation, which exhibited potentially interesting data value distributions. For each selected cluster group, we calculated statistics for the corresponding individual cluster nodes of the Kohonen lattice. The selected cluster groups for the query data set were 12, 30 and 40, which are detailed in Table 4.

In Table 4, columns 2 to 7 represent the same variables as those of Table 3. In column 1 we see the cluster id corresponding to the Kohonen lattice. For the corresponding cluster group, we show the first 5 clusters ordered by number of queries assigned. Therefore, in Table 4, query cluster group 12, we observe the summary statistics for individual query clusters (6,7; 13,4; 14,6; 12,2; 5,15).

Finally, in the last column of Table 4, "Freq of URL 1" is a variable which was not used as input to the query clustering, which represents the average quantile of the frequencies of the URL's which most coincided with the corresponding queries, in the given cluster. "Freq of URL 1" enables us to inter-relate the two different perspectives of user web search activity: queries made by the users and the documents they chose to click on. With reference to Table 4, query cluster group 12, we can see evidence of general characteristic values/tendencies, such as a low number of clicks (column 6) and medium hold times (column 2). We note that hold time values less than 1.0 have been rounded down to zero, thus the average values of zero for cluster group 40 in column 2. Individual clusters, such as (13,4) show specific characteristics such as low hold time (4.36), low number of terms in the query (1.29) and low number of clicks (1.21).

User Types: Now we make some interpretations of the level 1 query clustering results, in terms of the user categories presented in Section 1.

Navigational: the query-sessions grouped in level 1 query cluster 40 (see Tables 3 and 4) has a low hold time and a low number of clicks, which has a direct relation with Broder's proposal [3] with respect to the number of documents visited and time spent browsing as a consequence of a query of this type. One example of a "navigational" type query in cluster group 40 is "chilecompra" (chilepurchase) with corresponding URL "http://www.chilecompra.cl", average hold time of 0 and average number of clicks equal to 1.

Another example of a typical query in this cluster group is "venta de camisetas de futbol en chile" (sale of football shirts in chile) with corresponding URL: "http://www.tumejorcompra.tst.cl/-futbol.php", average hold time of 0 seconds and average number of clicks equal to 1.

Informational: in query cluster group 30 (see Tables 3 and 4), it can be clearly seen that clusters were generated which grouped the query-sessions whose number of clicks and hold time is high. One example of an informational type query in this cluster group is "cloroplasto" with principal corresponding URL "http://ciencias.ucv.cl/biologia/mod1/-b1m1a007.htm", average

hold time of 731 seconds and average number of clicks equal to 7. Another example is the query "software ingenieria structural" with principal corresponding URL "http://www.pilleux.cl/-mt771/", average hold time of 1062 seconds and average number of clicks equal to 8.

Transactional: in query cluster group 12 (see Tables 3 and 4) we can observe medium to high hold times and a low number of clicks, which coincides with our hypothesis for this type of users, although the characteristics are not as strong as for the "navigational" and "informational" user types. Given that we have the queries submitted to the search engine, and we have the documents (web pages) that the user selected from those retrieved by the search engine, we can confirm individual results as being transactional by visual inspection of the query and of the web page selected. For example, in cluster group 12, cluster 11,14 we have the following query: "compra y venta de autos" (purchase and sale of automobiles) with principal corresponding URL "http://autos.123.cl/registracion.asp", with a hold time of 580 seconds and average number of clicks equal to 3. In this case the transaction involves filling in a form.

Quality Profiles: We now interpret the clusters with reference to the session quality profiles presented in Section 1 (Table 1). *High1*: in all of cluster group 40 we can see a high clicked document ranking (low values) and a low number of clicks (all equal to 1), which corresponds to the hypothetical Profile 1 which indicates "high" quality. *High2:* cluster 30 has the highest average hold time (see Table 3), which is indicative of this quality type. In Table 5 we can see this characteristic confirmed by the corresponding level 2 clusters. *Low1:* cluster group 30 shows a low/medium clicked document ranking and a high number of clicks, which indicates a problem of low quality according to our definition. On the other hand, we also identified cluster group 30 as having profile *High2*, which is defined in terms of average hold time. This is not necessarily contradictory, given that the queries can show good quality in some aspects, and low quality in other aspects. We would have to investigate the individual level 2 clusters and samples of individual queries and their clicked documents, in order to confirm the problem areas. *Low2:* from the summary statistics of the query clustering, we have not clearly identified this profile among the clusters. We recall that profile *Low2* corresponds to a "low" quality profile, indicated by a low average hold time, together with a high number of clicks.

6. C4.5 TREE AND RULE INDUCTION

In this section we now use C4.5 to generate a decision tree/ruleset selecting from the whole query dataset (11981 queries). First we create a model with the user type label defined in Section 1.1 as the classifier category. Secondly, we use the quality label defined in Section 1.2 to create a second predictive model. In Section 6.1 we try training a model on the whole dataset, in order to identify individual rules of high precision which can be useful for user and session classification. The input variables were: "number of terms in query"; "query frequency in the historical data"; "average hold time for query", "average ranking of results selected for query", "average number of clicks for query", "frequency of the document/URL most retrieved by the query in the historical data", "average hour day for running of query (0 to 24)", "average day for running of query (1 to 7)". We note that the original variables have been used as input, not the quantile versions used as input to the Kohonen clustering. This was done to simplify the interpretation of the results in terms of the real data values. We used as training set the first two months click data, and as the test set we used the third consecutive month. See Section 3.1 for a description of the original data captured. All the statistical variables (averages, sums) were calculated exclusively for the corresponding time periods, in order to guarantee that only "a priori" information was used to train the models. The queries used in the train and test datasets were selected using the following criteria: the same query must occur in the train (months 1 and 2) and in the test data (month 3); frequency of query greater than 1 in the train data, and in the test data; frequency of the most frequent document corresponding to the query greater than 1, in the train dataset and in the test dataset. This selection is carried out in order to eliminate "unique" queries, and obtained a total of 1845 queries for the train and test datasets to predict "user type". In the case of the "quality label" model, we obtained a total of 1261 queries for the train and test datasets.

6.1 Rule and Tree Induction on All Dataset

In Figure 4 we see the resulting rules induced by C4.5 on the training dataset with the user categories as output. We observe that in order to classify 'nav' type users, C4.5 has used exclusively the 'hold time', whereas for 'tra' type users C4.5 has used 'hold time' and 'number of clicks'. Finally, in the case of 'inf' type users, C4.5 has used exclusively 'number of clicks'. This is coherent with our hypothetical definitions of these user types: 'nav' users have shorter hold times, 'inf' users have many clicks, and 'tra' users have greater hold times and fewer clicks. We could also say that the variables that best differentiate 'inf' users from 'nav' users are the hold time and the number of clicks, respectively.

Rule 1: Qholdtime ≤ 40
-> class nav [69.9%]

Rule 2: Qholdtime > 40
Qnumclicks ≤ 2
-> class tra [52.1%]

Rule 3: Qnumclicks > 2
-> class inf [56.4%]

Default class: nav

Figure 4. Rules induced by C4.5 on query data, using user type (nav, inf, tra) as the classifier label and % accuracy indicated for test dataset

The ruleset was evaluated on the test data (1845 items), which overall gave 686 errors (37.2%). The accuracy for the individual rules was as follows:

Rule	Used	Errors	Label
1	946	285 (30.1%)	nav
2	228	109 (47.8%)	tra
3	671	292 (43.5%)	inf

We observe that "nav" (navigational) is the easiest user type to predict, followed by "inf" (informational), whereas "tra" (transactional) seems to be more ambiguous and difficult to predict.

We also trained a model using the quality categories as the output label. The pruned decision tree generated by C4.5 for the quality categories is shown in Figure 5.

```
Qnumclicks <= 3 :
|  Qrank > 3 : high2 (171.0)
|  Qrank <= 3 :
|  |  Qnumclicks <= 2 : high1 (523.0)
|  |  Qnumclicks > 2 : high2 (25.0)
Qnumclicks > 3 :
|  Qholdtime <= 40 : low2 (108.0)
|  Qholdtime > 40 :
|  |  Qrank <= 3 : high2 (24.0)
|  |  Qrank > 3 : low1 (410.0)
```

Figure 5. Rules induced by C4.5 on query data, using session quality (high1, high2, low1, low2) as the classifier label

The tree of Figure 5 was tested on 1261 unseen cases, which gave an overall error of 44%. The classification matrix was as follows:

(a)	(b)	(c)	(d)	<< classified as	%correct by label
367	72	84	41	(a): class high1	65%
59	**73**	39	15	(b): class high2	39%
62	78	**207**	24	(c): class low1	56%
36	21	18	**65**	(d): class low2	37%

We observe that "high1" and "low1" are the easiest quality classes to predict, followed by "high2" and "low2" which gave significantly lower predictive accuracies. One possible cause of this could be the range assignments which we defined in Section 1.2, or due to ambiguities between the different classes.

6.2 Evaluation of Manually Defined Rules

As described in Sections 1.1 and 1.2, the rules for quality and user types were defined in terms of available descriptive variables, and the ranges were assigned by inspection of the distribution of each variable, together with consultation with the "domain" expert. For our data, the rule for INF type users corresponded to 36.4% of the total users in the training dataset, whereas the rule for TRA type users corresponded to 12.3% and the rule for NAV corresponded to 51.3%. The percentages in the test dataset varied in between 2 and 4%, with respect to those of the training dataset. If we "relax" the rule constraints, defining the low click threshold as 4 instead of 3, and the low holdtime threshold as 60 seconds instead of 40 (with respect to the ranges defined in Section 1.2), the percentage distributions change for the user types. In the case of INF users, the percentage in the train dataset goes down to 22.6%, that of TRA users goes up to 17.7%, and that of NAV users goes up to 59.7%. This shows a proportional increase for TRA and NAV of 5% and 8%, respectively, and a decrease for INF of 14%. Therefore, any change in the ranges has a significant effect on the classification of the user types. One approach would be to use the rules induced by C4.5 to calibrate the "hand-tunes" rules. Another approach would be to consider the rules in a parametric form and to learn the coefficients from the training data.

6.3 Summary Rule and Tree Induction

The overall precision on the whole dataset was not high for the rule induction model, although we did identify several rule "nuggets", which was our stated objective. One example of a good precision rule was "Qrank <= 3 and Qnumclicks <= 2 -> class high1" which had only a 31% error on the complete test dataset. It was also found that the user types were easier to predict than the quality classes. One further course of action would be to revise the ranges we assigned to the quality labels in Section 1.2, reassign the labels and rerun C4.5 on the data. Also we could try training on selected clusters from the Kohonen clustering, although we would have to retrain the clustering only on the first two months data, and test the rules on the third (unseen) month.

7. CONCLUSIONS

In this paper we have contrasted two different techniques, Kohonen SOM clustering and C4.5 rule/tree induction, for mining web query log data. This extends previous results done using other techniques such as k-means. We have detailed all the data mining steps, from initial data preparation, pre-analysis/inspection, transformation (quantiles, outliers), sampling, unsupervised and supervised learning algorithms, and analysis of results. In Section 1 of the paper we made some initial hypotheses about the user types and user session quality profiles, and in Sections 5 and 6 we have proceeded to analyze the results in order to identify characteristics which correspond to these types and profiles. The use of machine learning techniques to identify the user categories allows us to confirm the user type "mix" for specific data sets, and to define new user types. In this manner we can classify our users and query sessions in a way which helps us to quantify current user and search engine behavior, enabling us to adapt our system to it, and anticipate future needs.

A companion paper will use the same techniques from a document perspective, that is, the URLs clicked by the queries. In a forthcoming paper (Baeza-Yates, R., Calderon, L. and Gonzalez, C., *The Intention behind Web Queries,* SPIRE 2004, Glasgow, October 2004) we use SVM and Expectation Maximization to do unsupervised and supervised learning to explore similar issues, obtaining over 80% precision and recall for informational queries.

The next step is to use these results for particular applications such as improved ranking algorithms or different output interfaces for each query type.

Acknowledgements

This research was partially funded by Spanish MEC Grant TIN 2005-09201.

8. REFERENCES

[1] Baeza-Yates, R., Castillo, C. *Relating web structure and user search behavior* (extended poster). In *Proc. 10th World Wide Web Conference*, Hong Kong, China, May 2001.

[2] Baeza-Yates, R., Hurtado, C., Mendoza, M. and Dupret G. *Modeling user search behavior.* In *Proceedings of the Third Latin American Web Congress 2005*, p. 242 – 251. Buenos Aires, Argentina, Oct. 2005.

[3] Broder, A.Z. *A taxonomy of web search*. SIGIR Forum, 36(2):3-10, 2002.

[4] Hunt, E.B. *Artificial Intelligence*. Academic Press, New York, 1975.

[5] Kohonen, T. *Self organization and associative memory*. Berlin, Springer-Verlag, 1984.

[6] Lee, U., Liu, Z., Cho, J. *Automatic identification of user goals in web search.* In *Proc. 14th International World Wide Web Conference*, Chiba, Japan, May 2005.

[7] Nettleton, D.F. *El uso de tecnologia de minería de datos para la construcción y explotación del datawarehouse*. Novatica, Spain, pp. 52-55, 1999.

[8] Nettleton, D.F., Fandiño, V.L., Witty, M., Vilajosana, E. *The use of a data mining workbench for macro and micro economic modelling*. In *Proceedings of Data Mining 2000*, Cambridge University, U.K., July 5-7, pp. 25-34, 2000.

[9] Nettleton, D., Baeza-Yates, R. *Web Retrieval: techniques for the aggregation and selection of queries and answers, (in Spanish),* I Spanish Symposium on Fuzzy Logic and Soft Computing, Granada, Spain, Sept. 2005, 183-190.

[10] Ntoulas, A., Cho, J., Olston, C. *What's new on the web? The evolution of the web from a search engine perspective*. In *Proc. 13th International World Wide Web Conference*, New York, United States, May 2004.

[11] Quinlan, J.R. *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, Calif.. 1993.

[12] Sugiyama, K., Hatano, K., Yoshikawa, M. *Adaptive web search based on user profile constructed without any effort from users.* In *Proc. 13th International World Wide Web Conference*, New York, United States, May 2004.