On Extracting Session Data from Activity Logs

David Mehrzadi   Dror G. Feitelson

School of Computer Science and Engineering
The Hebrew University, 91904 Jerusalem, Israel

Abstract

Activity logs from large-scale systems facilitate the study of user behavior, which can be used to improve and tune the user experience. However, the available data often lacks important elements such as the identification of user sessions. Previous work typically compensated for this by setting a threshold of around 30 minutes, and assuming that breaks in activity longer than the threshold reflect breaks between sessions. We show that using such a global threshold introduces artifacts that may affect the analysis, because there is a high probability that long sessions are not identified correctly. As an alternative, we suggest that a suitable individual threshold be found for each user, based on that user's activity pattern. Applying this approach to a large dataset from the AOL search engine leads to a distribution of session durations that is free of artifacts like those that appear when using a global threshold.

Categories and Subject Descriptors I.6.5 [*SIMULATION AND MODELING*]: Model Development; H.1.2 [*MODELS AND PRINCIPLES*]: User/Machine Systems—Human factors; H.4.3 [*INFORMATION SYSTEMS APPLICATIONS*]: Communications Applications—Information browsers

General Terms Experimentation, Human Factors, Measurement

Keywords Session, Activity log, User behavior

1. Introduction

The notion of a *session* is central in describing user interactions with computer systems. Intuitively, a session comprises the sequence of activities in which the user engages. It starts when the user sits down at the terminal or connects to the system in any way, and ends when he leaves. The session may end for a variety of reasons. One possibility is that the user has achieved what he came to do. Another is that he gave up in frustration. Understanding such dynamics is one of the reasons that data about sessions is important. We give specific examples in Section 2.

Regrettably, data about user sessions is often hard to come by. Most large interactive systems (such as web-based systems) do in fact log all user activities. However, they typically do not log the beginnings and ends of sessions, possibly because this data is not explicitly available. We therefore have to resort to analyzing the existing information in order to infer session boundaries.

The most commonly used approach for identifying session boundaries relies on global thresholds and common sense. The notion of a session implies continuous activity. Therefore any long breaks in activity may be taken as evidence that the session has ended. The question is then reduced to one of providing a quantitative metric for "long". The commonly used answer is to compare breaks with a certain global threshold, and label any breaks that surpass the threshold as session breaks. Work on how to select a suitable threshold is surveyed below in Section 3.

Somewhat surprisingly, little if any research has been conducted regarding the implications of this approach. In particular, an important question is the degree to which the resulting data about user sessions is sensitive to the selected threshold. Our results in Fig. 2 unfortunately indicate that such sensitivities exist. Specifically, we find that the distribution of derived session durations reflects the threshold used to derive the sessions. This is highly undesirable, because it implies that any research using the data about the sessions is also tainted and its results may depend on the precise threshold that was used.

In order to provide a more reliable alternative, we perform a user-based study. The results indicate that the breaks of each user typically have a bimodal distribution with a natural threshold, but that these thresholds do not necessarily coincide (Section 4). We therefore suggest that individual thresholds be used rather than a single global threshold. This approach leads to the results described in Sections 5 and 6. In particular, the derived distribution of session durations is shown to be free of artifacts that can be directly attributed to the methodology used in extracting the session data.

Our work is done in the context of web search engine logs, and in particular the extensive log that was released by AOL in 2006, which is one of the largest datasets that is available for research [19]. This log provides timestamps only for the queries submitted by users, and not for clicks on results. The considerations and results may be different in other systems or if more data is available, specifically clicks data [23].

2. Sessions and Their Importance

As noted above, our definition of a session is the sequence of activities performed by a user "in one sitting". This is equivalent to the period of time from logging into a system to logging out again. However, in many interactive systems users are not required to log in, and even if they are, they may not bother to log out. Login and logout events are therefore not always available, and cannot be used to characterize sessions in the general case.

It should be noted that other definitions of the term "session" have been used in the literature. Gayo-Avello provides an extensive survey in the context of query logs from web search engines [9] (which is also the context in which most of our work is carried out). In particular, some researchers use the term session to refer to the sequence of activities performed to satisfy an information need, regardless of how long it takes and whether there are any breaks in

SYSTOR 2012 June 4–6, Haifa, Israel

context	threshold	reference
general web surfing	10–15 min	[10]
	30 min	[20]
	2 hr	[17]
web search	5 min	[24]
	30 min	[6, 7, 13]
	1 hr	[23]
e-commerce	30 min	[16]
parallel supercomputers	20 min	[21, 27]

Table 1. *Examples of previous work using a global threshold to define sessions.*

the middle [2, 13]. We stress that we do not use this definition, but rather the definition involving a sequence of actions done consecutively one after the other. However, in Section 6 we compare the two definitions and show they produce largely consistent results.

Once data about user sessions is available, this can be used to describe the dynamics of system use. For example, Arlitt provides a detailed description of how users interacted with the web site of the 1998 Soccer World Cup tournament in France [1]. This includes the number of pages requested by users in each session, the amount of data transferred in a session, and the distribution of think times between requests in a session. Similarly, Zilber et al. analyzed the workload on several large-scale parallel supercomputers, and used a classification of sessions to identify several user types based on their preferred time of activity and the characteristics of the jobs they submit [27]. Such data is important when characterizing the locality of workloads, namely the regularity that may be expected by adaptive systems that attempt to adjust to their workloads [8].

The next step can be to use such data in order to create a generative workload model. Such models mimic the behavior of users in order to create realistic workloads, including effects such as diversity and feedback. For example, Costa et al. have created a model of how users interact with sites providing streaming media [5]. This can be used to evaluate different content distribution schemes under realistic conditions. Shmueli and Feitelson have created a user-based model of the workload on parallel supercomputers [21]. Including feedback and session terminations in the model enabled the evaluation of innovative schedulers that attempt to prioritize users who are most probably engaged in an interactive session [22]. This has the dual effect of delaying less urgent work to non-prime time, and freeing up resources at prime time thus enabling an increase in system utilization and throughput.

The goal of generative workload models is to enable the design and evaluation of better systems. In the context of e-commerce, counting user sessions provides a more meaningful metric for a site's potential than just counting unique users. Moreover, characterization of user behavior in sessions facilitates the classification of users and the identification of those that are more likely to buy [15]. Based on such considerations, Cherkasova and Gupta suggest an admission control mechanism based on the observation that it is more important to guarantee the completion of sessions that have already been admitted [4]. Naturally this depends on the ability to identify sessions, both in logged data and in real time.

3. Finding Sessions Using Global Thresholds

As noted above, the dominant methodology to extract session data from activity logs is to postulate a certain threshold value, and assume that breaks in activity longer than this threshold represent sessions breaks. An extensive survey is given by Gayo-Avello [9], and examples from various contexts are shown in Table 1. In addition to being used in log-based research, global thresholds are also used by on-line utilities. For example, Google analytics uses a timeout of 30 minutes to define sessions.

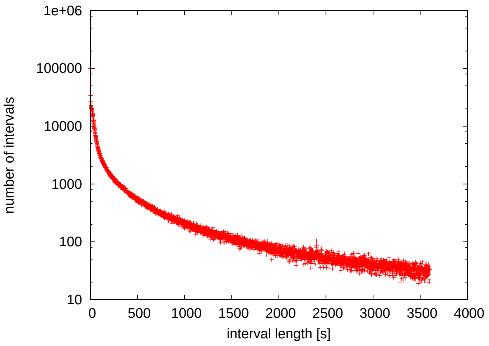

Figure 1. *distribution of intervals between successive queries from the AOL 2006 log.*

Several approaches have been proposed for setting suitable threshold values. One simple approach is to consider the distribution of intervals between successive actions, for all users of the system. This typically shows that in most cases the interval is short, but that long intervals are also possible. The threshold is then set so as to include the bulk of the distribution, that is the large number of relatively short intervals that are observed.

As an example, the distribution of intervals observed in the AOL search log is given in Fig. 1. Note that the Y axis is logarithmic; thus there is a very large number of intervals that are very short, up to a few minutes. In the range of 20-30 minutes (around 1000 to 2000 seconds) the slope of the plot is much reduced, which would make it appear as a suitable range for the threshold. But there is no unique spot that would appear to be a natural threshold value. Wang et al. suggest to compare the time between a click and the next query to the average of such times [25]. However the average may be tainted by very long intervals that occur if a user was not active for a long time.

An alternative approach is to set the threshold based on its effect on the number of sessions that will be identified [10]. If the threshold is set too low, short sessions are broken into individual actions (e.g. queries in the case of web search data). As the threshold is increased, the number of sessions drops and then stabilizes. He and Göker suggest that the threshold should be set at the point where this happens [10]. A more quantitative mathematical model for this was recently derived by Huynh and Miller [12].

Another option is not to commit to a single threshold, but rather to use a whole set of possible thresholds. This allows one to study the effect of the threshold value on the ensuing analysis. This approach was used by Arlitt in his analysis of the WC'98 site [1].

Our concern with using a global threshold is not with the method used to select the threshold value, and in fact it may be the case that the exact value has little effect on results [3]. However, we are concerned with the effect of applying the same threshold to all users and all sessions. Examples of the results of doing so with the AOL 2006 search log are shown in Fig. 2, using three different threshold values: 10 minutes, 20 minutes, and 1 hour. The following observations can be made:

- The distribution of resulting session durations has a pronounced break at the point of the threshold used to create it, with a sharp reduction in the number of sessions longer than the threshold value (indicated by arrows in the figure). This is probably due to artificial truncation of longer sessions due to using a threshold that is too low. Note that, especially for low thresholds, this means that most probably *all* non-trivial sessions are not iden-

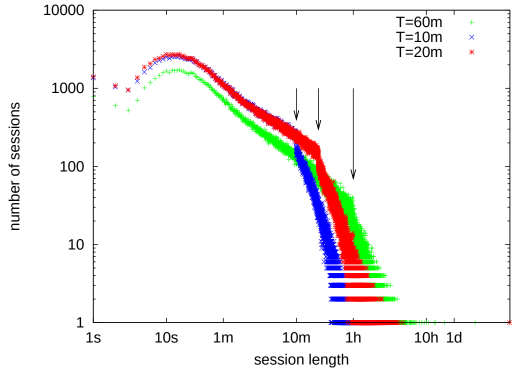

Figure 2. *Histograms of session durations obtained when using thresholds of 10, 20, and 60 minutes.*

tified correctly.

- The body of the distribution for thresholds of 10 and 20 minutes is essentially the same.
- The body of the distribution using a threshold of 60 minutes is uniformly lower by about a third. This is attributed to short sessions being united into longer ones due to using a higher threshold. This could mean that long sessions are not identified correctly — at least some of them may be bogus.

The conclusion is that it is not practical to find a suitable global threshold: any chosen value will be too short for some sessions with relatively long breaks, but too long for other cases where it will erroneously concatenate sessions that should actually remain separated. In either case, the data about the longer sessions (which may be the more important ones) will be erroneous.

4. Finding Sessions Using Per-User Thresholds

The claim that a single threshold may not be appropriate for all users is substantiated by the data shown in Fig. 3. The data is again from the AOL search log. The figure includes texture plots of the distribution of intervals between queries for select users from the log. The X axis of these plots is the interval length, in a logarithmic scale. The Y values are randomized so as to displace the points relative to each other and expose their density. Assuming that intervals within a session are short and intervals between sessions are long, we want to identify a gap in the distribution that may be used to distinguish between the short intervals and the longer ones. The users in the figure are arranged so that in the central column a value of 30 minutes appears to be a reasonable threshold between short and long intervals. In the left-hand column, it appears that a lower threshold may be better, whereas in the right-hand column a higher threshold value seems preferable. This motivates the idea of adjusting the thresholds and using a customized value for each individual user.

The idea of using user-specific thresholds has been proposed before by Murray et al. [18] (and essentially the same idea was suggested even earlier by Ware et al. in the context of file-access patterns [26]). Given data about a single user's intervals between queries, they suggested the following algorithm to find a threshold to distinguish between short and long intervals. First, sort the intervals from short to long. Then, for each one, calculate the quotient of the interval divided by the standard deviation of all those that appeared before it in the list, that is all those that are shorter. The interval for which this quotient is maximized is thus identified as being substantially longer than all previous ones, and the threshold should be set between this interval and the preceding one.

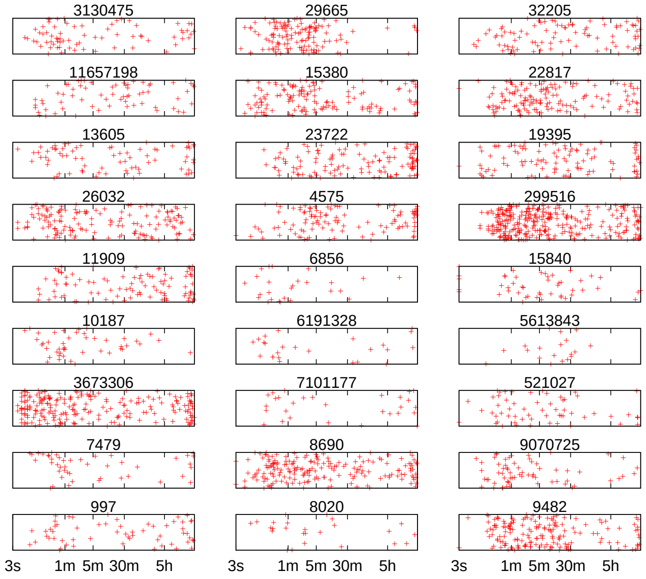

Figure 3. *Texture plots of the distribution of intervals between queries for select users from the AOL 2006 log.*

While this algorithm seemed to work well for the examples checked, it may be susceptible to the following problems:

- The maximum may occur at values that are not reasonable, e.g. a couple of minutes (too short) or several hours (too long).
- The algorithm is explicitly based on the assumption that the distribution is bimodal, with sort intervals that have little variability and long ones that also have little variability. Under these conditions adding the first long interval will indeed maximize the set's variance. But if the long intervals are themselves varied (including breaks ranging from a few hours to days or even weeks) there may be several high values, and the maximum may not necessarily identify the first large interval. In particular, this indicates that the algorithm is not suitable for analyzing long logs like the 3-month AOL search log.

In order to avoid these potential pitfalls we suggest the following simple algorithm, which codifies common sense and domain knowledge rather than relying on more abstract statistical devices. The distribution of intervals has many short intervals that represent gaps between actions within the same session. But the long intervals (between sessions) may be very different from each other, so the resulting distribution is not necessarily bimodal. In order to actually group the long intervals together and create a second mode we use binning on a logarithmic scale. The goal of the algorithm is then to find the best global minimum between the two modes that is also "reasonable". This is done as follows:

- In order to obtain a general view of the distribution of intervals, we compute a histogram using logarithmically-sized bins, with bin boundaries at powers of 2. These are somewhat coarse in order to avoid excessive detail and the danger of local minima.
- In order to guarantee that the resulting threshold is reasonable, we only consider the bins ranging from 512 seconds (slightly less than 10 minutes) to 8192 seconds (around $2\frac{1}{4}$ hours).
- Each candidate bin is given a score, based on how much lower it is than the maxima on its two sides. Empty bins get a bonus.
- The threshold is placed in the highest-scoring bin.
- In case of a tie, the bin closest to 1200 seconds (20 minutes) is selected.

```
// input: t[i] is timestamp of i'th query

// find intervals
for i=2..N
    d[i-1] = t[i] - t[i-1]

// create histogram
for i=1..N-1
    bin = 1
    lim = 32
    while (d[i]>lim)
        bin++
        lim *= 2
    hist[bin]++

// assign scores in desired range
for bin=5..9
    maxLeft = max( hist[2..bin-1] )
    maxRight = max( hist[bin+1..12] )
    score = 0
    if (hist[bin] <= 2/3*maxLeft) score++
    if (hist[bin] <= 1/2*maxLeft) score++
    if (hist[bin] <= 1/3*maxLeft) score++
    if (hist[bin] <= 1/6*maxLeft) score++
    if (hist[bin] <= 2/3*maxRight) score++
    if (hist[bin] <= 1/2*maxRight) score++
    if (hist[bin] <= 1/3*maxRight) score++
    if (hist[bin] <= 1/6*maxRight) score++
    if (hist[bin] == 0) score = 5
    s[bin-4] = score

// find maximal score
lim = threshold = 512
max = s[1]
for bin=2..5
    lim *= 2
    if (s[bin]>max)
        max = s[bin]
        threshold = lim
if ((threshold==512) && (s[1]==s[2]))
    threshold = 1024
```

Figure 4. *Pseudocode of algorithm for setting the threshold.*

Detailed pseudo-code of the algorithm is given in Fig. 4. Examples of the histograms of intervals for different users are shown in Fig. 5. As can easily be seen, many of the histograms have minima that can serve as natural thresholds between short and long intervals. Thus the continuous distribution of Fig. 1 is actually the sum of multiple distributions that each may have a natural threshold, but at a somewhat different location.

5. Direct Evaluation

Regrettably, we do not have any ground-truth regarding real session breaks in the AOL log. We therefore need to employ indirect methods in order to evaluate the quality of the session breaks generated by using per-user individual thresholds according to our algorithm. We use two such methods. First, we look at the produced results in isolation. Then we subject them to a subjective test in comparison to human judgment.

The main output of our algorithm is an identification of session breaks for the activity of each user in the log. To obtain a global view of these results, we plot the distribution of session lengths for all the users in aggregate. This is shown in Fig. 6. The graph immediately indicates that the artificial breaks caused by using

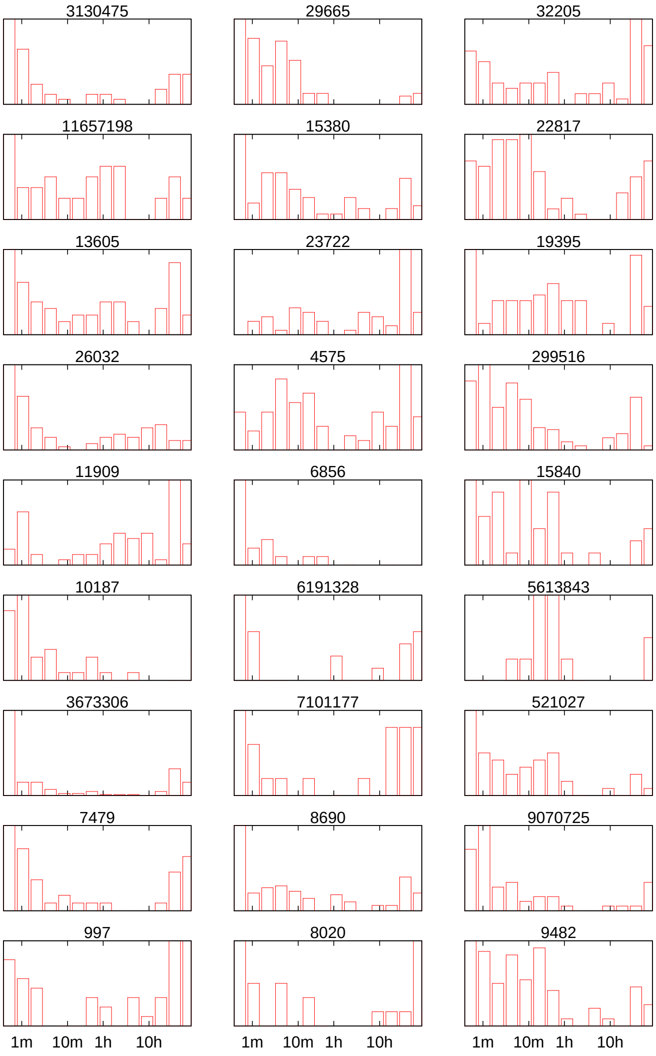

Figure 5. *Histograms of intervals between queries corresponding to the data shown in Fig. 3.*

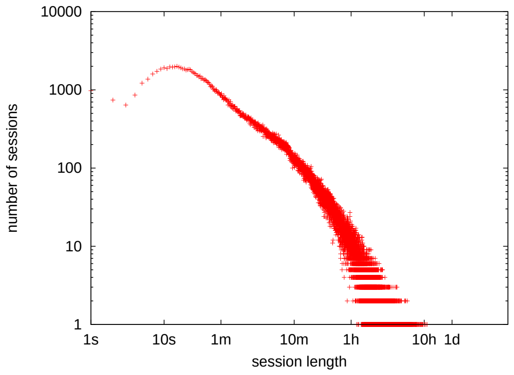

Figure 6. *Histogram of session durations obtained when using individual bounds for different users using our algorithm. Compare with Fig. 2.*

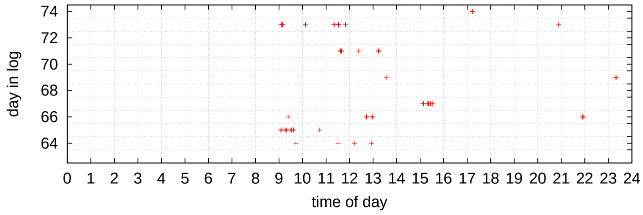

Figure 7. *Excerpt of activity of a certain user in the AOL search log.*

global thresholds, as shown in Fig. 2, have been eliminated. Thus the resulting session data does not directly reflect a parameter of the methodology, as happens when using a global threshold.

It is also interesting to compare the resulting distribution with those generated for different thresholds. This shows that the number of short sessions, in the range of up to about 10 minutes, is uniformly lower by about a quarter than for the distributions with low thresholds of 10 or 20 minutes; it is slightly higher than for the distribution created with a threshold of 60 minutes. On the other hand, more long sessions are identified.

To perform a more detailed evaluation we consider the methodology of Murray et al. [18]. This involved using human judgment to label a sample of the original data, and decide which sequences of actions appear to constitute a session. Based on this human labeling, they measured the precision and recall of both global threshold algorithms and their own user-based threshold algorithm. Interestingly, they found that each type of algorithm is better for a different metric. The global threshold algorithms had excellent recall, and identified practically all the session breaks labeled by the human judges. However, they also had many false positives, leading to lower precision scores. Their user-based algorithm, on the other hand, had excellent precision and practically no false positives. However, it had much lower recall, and missed many session boundaries identified by the human judges.

In order to apply this methodology, we first selected 50 random users with the caveat that we eliminated users who had less than 20 queries in the whole log, in order to ensure that we had enough activity to work with. Using displays of user activity like that shown in Fig. 7, we marked all intervals that appeared to be session breaks. For example, our human judge decided that the 4 queries on day 64 in the figure constitute a single session. However, the gap between the first and second queries on day 66 was marked as a session break. Analyzing these markings, it turns out that human markings are highly consistent with using a per-user threshold. In 38 cases (76%) there was a clean separation, and all marked intervals were longer than all unmarked intervals. In the rest, at most 4.3% of the unmarked intervals were longer than the minimal marked interval, and in most cases the percentage was much lower.

Comparing the markings of the human judge with the thresholds produced by the algorithm led to results similar to those of the global-threshold algorithms considered by Murray et al. In particular, of 4992 intervals considered, there was consensus that 1334 constitute session breaks and 3384 are intervals within a session. There were only 4 gaps judged to be breaks by the human but not by the algorithm, leading to very high recall. However, the algorithm also identified 270 intervals as session breaks that were not considered breaks by the human, leading to only 83% precision. Thus our algorithm appears to exhibit the opposite behavior to that of Murray et al.

The meaning of these results is that the algorithm sometimes sets the threshold lower than the human. Indeed, when analyzing the thresholds produced, we find that the human judge was much more consistent, always using thresholds in the range of 1–2 hours. The algorithm, in contradistinction, used all the allowed threshold values, ranging from 512 seconds to 8192 seconds. Choosing the lower threshold values then identified breaks that were not labeled as such by the human. However, we have no way to really know which classification is better.

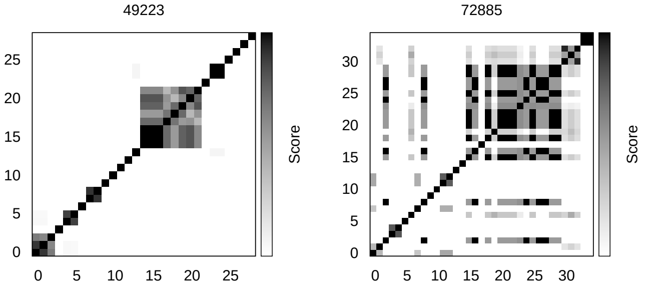

Figure 8. *Example heatmaps showing similarity between successive queries.*

6. Comparison with Semantic Boundaries

The evaluation presented in the previous section is based only on timing data. This risks a circular argument, where breaks in activity are used both to find session boundaries and to evaluate whether the found boundaries are reasonable. An alternative is to compare the session boundaries we found with topical breaks in the sequence of queries. We expect that distinct sessions will typically include queries on different subjects.

The problem is that related queries, those that are expected to belong to the same session, need not be identical to each other. In fact, one of the reasons for long sessions may be that the user did not find what he was looking for, and tried again and again with various modifications of the original query. We therefore need a method to quantify the degree of similarity between queries. For this we use common n-grams.

An n-gram is a sequence of n consecutive letters from a search term. For example, if a user is looking for a parking garage, the queries “park” and “parking” are different. But if we look at 4-grams, then “parking” is replaced by the set {“park”, “arki”, “rkin”, “king”}, which has one 4-gram in common with the shorter query. Such similarity is then quantified using the Jackard distance, i.e. the ratio of the shared n-grams to the total distinct n-grams in both queries.

We visualize the relationship between successive queries using heatmaps, as exemplified in Fig. 8. Both the X axis and the Y axis denote the serial numbers of a user’s queries. The i, j location then represents the similarity between the ith query and the jth query, where white means no common n-grams and black means that all n-grams appeared in both queries. These heatmaps are symmetrical, so either the top or bottom triangle could be used instead of drawing the full square. By design, the diagonal is all black (each query is identical to itself). Large squares (possibly with various levels of gray) on the diagonal denote sequences of related queries. An example is the large square comprising queries 15 through 22 by user 49223, which are all related to efforts to find a parking garage in Manhattan. Off-diagonal elements denote situations where the user returns to search again for something that he has searched for before. For user 72885, the repeated theme was the game of lacrosse.

Given such heatmaps, we can add lines between queries that our algorithm decides are in different sessions. This combines the

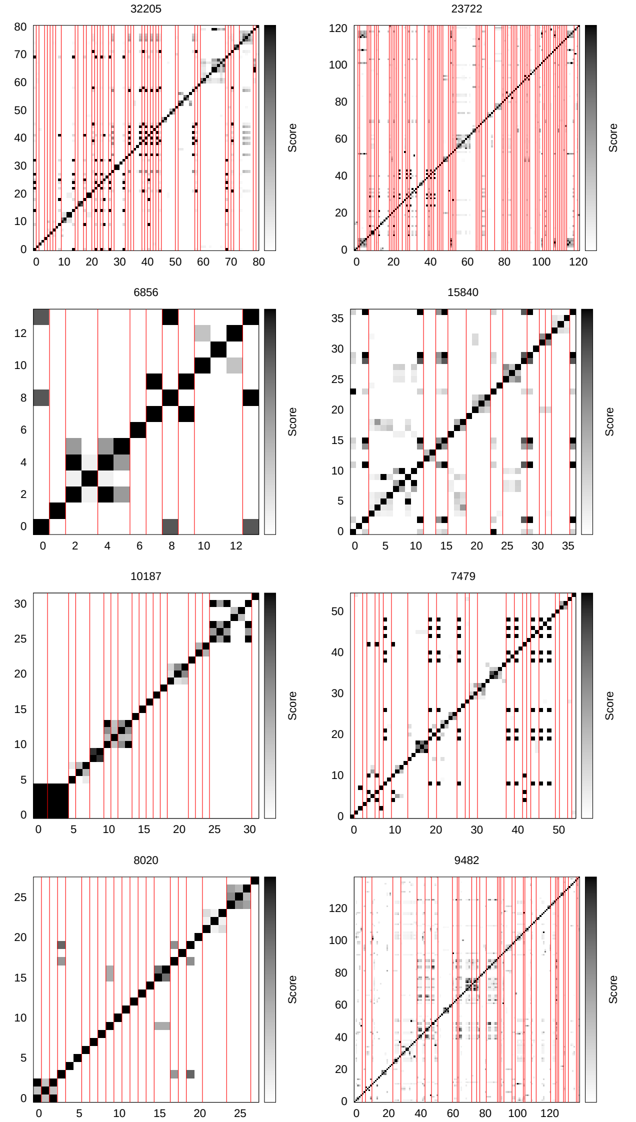

Figure 9. *Combining the query similarity heatmaps with session breaks as found by our algorithm.*

session breaks we found with the query similarity data. Examples of the obtained results are shown in Fig. 9.

As we can see, in many cases the session boundaries indeed correspond to the large squares on the diagonal of the heatmaps. But there are also situations where a session boundary dissects a square, or where several small squares are not divided by a session boundary. These do not necessarily indicate a problem. In the first, the user continues to look for similar things in the next session. In the second, the user does several unrelated things in sequence.

In addition, other artifacts may influence this analysis. One is that some users have a certain field of interest, and all their queries therefore tend to include a common n-gram even if they are actually unrelated. Examples we have observed are “pictures”, “lyrics”, and “golf”. Another is that some of the users may be bots, that is automated agents who issue queries repeatedly in certain intervals.

Despite these reservations, the overall agreement between session boundaries and topic changes is reasonably high. Only about 7% of sequences of related queries were cut by a session boundary. However, most sequences were of length 1 (meaning that the query was unrelated to those that came before or after it). If we only consider sequences of length 2 and above, then 29% of them were cut by a session boundary.

Conversely, 79% of the sessions had only one topic in them. Of the 21% that had more than one topic, 13% had two topics, and the other 8% had more than two topics. However, this may include cases where the user actually continued to search for the same thing, but used completely different search terms.

Again, these numbers do not necessarily imply a problem. For example, Jones and Klinkner analyze the queries of users conducting web search, and conclude that search tasks may be interleaved and have a hierarchical structure [14]. Nevertheless, work on detecting topic shifts [11] may add useful information beyond the use of thresholds, and it would be interesting to see how the two can be combined into a single procedure for identifying sessions.

7. Conclusions

Finding session boundaries in activity logs is a hard problem. The commonly used approach is to define a global threshold on inter-activity intervals, and label any intervals longer than this threshold as session breaks. However, this approach seems to be inappropriate. First, the global distribution of inter-activity intervals is typically smooth, with no natural threshold value. Worse, using a global threshold may lead to artifacts that directly reflect the chosen threshold value, like the breaks in the session duration distribution shown in Fig. 2. As a result it is nearly guaranteed that long sessions will not be identified correctly.

As an alternative, we suggest using domain knowledge and intuition to set per-user thresholds rather than a single global threshold. This is based on the observation that the distributions of intervals for individual users often do display a structure including a natural threshold between short and long intervals. However, these individual thresholds are different for different users, explaining the lack of such structure in the global distribution. Using this approach, we show that artifacts like those created by a global threshold are eliminated.

It should be noted that the best thresholds may be different for different contexts. For example, in web servers, sites that provide services like calculating monetary exchange rates may expect very short sessions, whereas social sites like Facebook and socio-entertainment sites like YouTube may expect very long sessions — including longer breaks [12]. Our approach for finding empirical per-user thresholds naturally accommodates such diversity.

The biggest problem with finding session boundaries is the lack of adequate data. One important aspect of data quality is that timestamps often denote only the beginning of a set of activities. For example, in the AOL search log we used, only query timestamps are provided. Having data about the timestamps relating to clicks on search results would enable us to better characterize the continuity of user activity, as a long sequence of clicks may cause a long interval between queries that does not really represent a break in activity. Another aspect of data quality is the total lack of labeled data that can be used for evaluation and learning. Moreover, it is not clear that intuitive human judgment is a good substitute for such information. It is therefore imperative that more and better data be collected about user behavior in different domains of activity.

Acknowledgment

This work was supported by a grant from the Israeli Internet Association.

References

[1] M. Arlitt, "*Characterizing web user sessions*". *Performance Evaluation Rev.* **28(2)**, pp. 50–56, Sep 2000.

[2] D. J. Brenes and D. Gayo-Avello, "*Stratified analysis of AOL query log*". *Information Sciences* **179(12)**, pp. 1844–1858, May 2009.

[3] N. N. Buzikashvili and B. J. Jansen, "*Limits of the web log analysis artifacts*". In *Workshop on Logging Traces of Web Activity: The Mechanics of Data Collection*, May 2006.

[4] L. Cherkasova and P. Phaal, "*Session-based admission control: A mechanism for peak load management of commercial web sites*". *IEEE Trans. Comput.* **51(6)**, pp. 669–685, Jun 2002.

[5] C. Costa, C. Ramos, Ítalo Cunha, and J. M. Almeida, "*GENIUS: A generator of interactive user media sessions*". In 7th *Workshop on Workload Characterization*, pp. 29–36, Oct 2004.

[6] D. Donato, F. Bonchi, T. Chi, and Y. Maarek, "*Do you want to take notes? identifying research missions in yahoo! search pad*". In 19th *Intl. World Wide Web Conf.*, pp. 321–330, Apr 2010.

[7] D. Downey, S. Dumais, and E. Horvitz, "*Models of searching and browsing: Languages, studies, and applications*". In 20th *Intl. Joint Conf. Artificial Intelligence*, pp. 1465–1472, Jan 2007.

[8] D. G. Feitelson, "*Locality of sampling and diversity in parallel system workloads*". In 21st *Intl. Conf. Supercomputing*, pp. 53–63, Jun 2007.

[9] D. Gayo-Avello, "*A survey on session detection methods in query logs and a proposal for future evaluation*". *Information Sciences* **179(12)**, pp. 1822–1843, May 2009.

[10] D. He and A. Göker, "*Detecting session boundaries from web user logs*". In 22nd *BCS-IRSG Ann. Colloq. Information Retrieval Research*, pp. 57–66, 2000.

[11] D. He, A. Göker, and D. J. Harper, "*Combining evidence for automatic web session identification*". *Inf. Process. & Management* **38(5)**, pp. 727–742, Sep 2002.

[12] T. Huynh and J. Miller, "*Empirical observations on the session timeout threshold*". *Inf. Process. & Management* **45(5)**, pp. 513–528, Sep 2009.

[13] B. J. Jansen, A. Spink, C. Blakely, and S. Koshman, "*Defining a session on web search engines*". *J. Am. Soc. Inf. Sci. & Tech.* **58(6)**, pp. 862–871, Apr 2007.

[14] R. Jones and K. L. Klinkner, "*Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs*". In 17th *ACM Conf. Inf. & Knowledge Management*, pp. 699–708, Oct 2008.

[15] D. A. Menascé, V. A. F. Almeida, R. Fonseca, and M. A. Mendes, "*A methodology for workload characterization of e-commerce sites*". In 1st *ACM Conf. Electronic Commerce*, pp. 119–128, Nov 1999.

[16] D. A. Menascé, V. A. F. Almeida, R. Riedi, F. Ribeiro, R. Fonseca, and W. Meira Jr., "*A hierarchical and multiscale approach to analyze E-business workloads*". *Performance Evaluation* **54(1)**, pp. 33–57, Sep 2003.

[17] A. L. Montgomery and C. Faloutsos, "*Identifying web browsing trends and patterns*". *Computer* **34(7)**, pp. 94–95, Jul 2001.

[18] G. C. Murray, J. Lin, and A. Chowdhury, "*Identification of user sessions with hierarchical agglomerative clustering*". In *Proc. Am. Soc. Inf. Sci. & Tech.*, vol. 43, 2006.

[19] G. Pass, A. Chowdhury, and C. Torgeson, "*A picture of search*". In 1st *Intl. Conf. Scalable Information Syst.*, Jun 2006.

[20] B. Schroeder, A. Wierman, and M. Harchol-Balter, "*Open versus closed: A cautionary tale*". In 3rd *Networked Systems Design & Implementation*, pp. 239–252, May 2006.

[21] E. Shmueli and D. G. Feitelson, "*Uncovering the effect of system performance on user behavior from traces of parallel systems*". In 15th *Modeling, Anal. & Simulation of Comput. & Telecomm. Syst.*, pp. 274–280, Oct 2007.

[22] E. Shmueli and D. G. Feitelson, "*On simulation and design of parallel-systems schedulers: Are we doing the right thing?*" *IEEE Trans. Parallel & Distributed Syst.* **20(7)**, pp. 983–996, Jul 2009.

[23] E. Shriver and M. Hansen, *Search Session Extraction: A User Model of Searching*. Tech. rep., Bell Labs, Jan 2002.

[24] C. Silverstein, M. Henzinger, H. Marais, and M. Moricz, "*Analysis of a very large web search engine query log*". *SIGIR Forum* **33(1)**, pp. 6–12, Fall 1999.

[25] C.-J. Wang, K. H.-Y. Lin, and H.-H. Chen, "*Intent boundary detection in search query logs*". In 33rd *SIGIR Conf. Information Retrieval*, pp. 749–750, Jul 2010.

[26] P. P. Ware, T. W. Page, Jr., and B. L. Nelson, "*Automatic modeling of file system workloads using two-level arrival processes*". *ACM Trans. Modeling & Comput. Simulation* **8(3)**, pp. 305–330, Jul 1998.

[27] J. Zilber, O. Amit, and D. Talby, "*What is worth learning from parallel workloads? a user and session based analysis*". In 19th *Intl. Conf. Supercomputing*, pp. 377–386, Jun 2005.