# Mining search logs for usage patterns

## Introduction

One of the greatest opportunities and challenges of the 21$^{st}$ century is the ever increasing significance of data. Data underpins our businesses and our economy, providing awareness and insight into in every sphere of life; from politics to the environment, arts and society. The everyday interactions between people and devices can be harnessed to power a new generation of products and services, allowing us to better understand human needs, aspirations and behaviour.

Of all the data to which we have access, there is none more valuable than the trace people leave when they *search* for digital information. In browsing the web, people reveal something about their behaviour and habits, but little about their *intent*. By contrast, when people *search* for information, they express in their own words their explicit needs and goals. This data represents a unique resource that offers extraordinary potential for delivering insights that can drive the next generation of digital services and applications.

Various studies have been undertaken to understand how and why people interact with search engines. Such studies have led to the creation of frameworks that describe distinct patterns of use, ranging from individual queries to entire information seeking episodes. These patterns may focus on information seeking behavior [9], the types of search tasks that users perform [10], their goals and missions [5], their task switching behavior [4], or the tasks, needs and goals that they are trying to address when using search systems [10, 11].

Moreover, the academic community is not alone in showing an interest in mining search logs. Two highly influential commercial organisations, ElasticSearch and LucidWorks, have both recently released independent logfile analysis platforms (Kibana and SiLK respectively [13, 14]). What unites all of these efforts is the belief that finding distinct, repeatable patterns of behaviour can lead to a better understanding of user needs and ultimately a more effective search experience. In this chapter, we explore the use of data mining techniques to find patterns in search logs, focusing on the application of open source tools and publicly available data.

## Getting started

Search logs are created when a search engine records its interactions with other agents. Typically, such agents are people interacting with the search engine via a web browser or other sort of client application. However, automated agents such as bots and indexing agents can also interact with a search engine, and their activity may also be recorded in the logs. Each interaction results in the addition of one or more records to the log, so these files can eventually grow to several Gigabytes in size.

Since there are many different types of search application, serving different audiences with different content for different needs, the type of information that gets stored in search logs also varies. Some logs can be relatively simple, storing just minimal information, while others can be highly detailed, with extensive metadata describing each interaction. There is a no ‘standard’ format for what should be recorded in a search log, or indeed how it should be formatted. But at the very least a search log should store information like the content of the query strings submitted by the user, and the items that they click on in response.

Since logs can vary so widely, it is prudent to start with one that is well known and publicly available. A good such example is the AOL search log, released in 2006 [15]. It consists of a compressed text file containing 21,011,340 queries by 657,426 users over a 3-month period. This log has a somewhat infamous history, due to concerns that users could potentially be identified by their searches [16]. However, it is useful as it is relatively large and the records contain a variety of fields that offer a range of data mining possibilities. A sample from the log is shown in Figure 1.

```
1326   cascadefamilymedical       2006-03-14 11:36:57
1326   cascadefamilymedical.com   2006-03-14 11:39:49
1326   milaniwheel.com     2006-03-14 12:37:30
1326   www.ameicaneaglewheel.com  2006-03-14 18:53:20
1326   www.ameicaneaglewheel.com  2006-03-15 12:27:48
1326   pop up adds  2006-03-15 20:07:38
1326   pop up adds  2006-03-15 20:08:29
1326   the childs wonderland company    2006-03-21 11:50:10
1326   the child's wonderland company   2006-03-21 11:59:03 6
       http://www.wonderlandtheatre.com
1326   the child's wonderland company   2006-03-21 12:00:55
1326   the child's wonderland company grand rapids michigan  2006-03-21 12:01:24
1326   the child's wonderland company grand rapids michigan  2006-03-21 12:01:59
1326   the childs wonderland co.  2006-03-21 21:20:42
1326   the child's wonderland co. 2006-03-21 21:22:16
```

Figure 1: A sample of records from the AOL log

The data includes the following fields: {AnonID, Query, QueryTime, ItemRank, ClickURL}.

- AnonID - an anonymous user ID number.
- Query - the query issued by the user, case shifted with most punctuation removed.
- QueryTime - the time at which the query was submitted for search.
- ItemRank - if the user clicked on a search result, the rank of the item on which they clicked is listed.
- ClickURL - if the user clicked on a search result, the domain portion of the URL in the clicked result is listed.

When analyzing search logs it is helpful to think of them in terms of user ‘sessions’: periods of continued interaction between a specific user and a search application [1-3]. By convention, a session typically begins with a query and ends with a webpage or some arbitrary time-based cut off, such as 30 minutes of inactivity [8]. So in the sample above we might infer that there are seven individual sessions, as illustrated in Figure 2.

```
SESSION1: 1326      cascadefamilymedical       2006-03-14 11:36:57
SESSION1: 1326      cascadefamilymedical.com   2006-03-14 11:39:49
SESSION2: 1326      milaniwheel.com     2006-03-14 12:37:30
SESSION3: 1326      www.ameicaneaglewheel.com  2006-03-14 18:53:20
SESSION4: 1326      www.ameicaneaglewheel.com  2006-03-15 12:27:48
SESSION5: 1326      pop up adds  2006-03-15 20:07:38
SESSION5: 1326      pop up adds  2006-03-15 20:08:29
```

```
SESSION6: 1326      the childs wonderland company    2006-03-21 11:50:10
SESSION6: 1326      the child's wonderland company   2006-03-21 11:59:03 6
       http://www.wonderlandtheatre.com
SESSION6: 1326      the child's wonderland company   2006-03-21 12:00:55
SESSION6: 1326      the child's wonderland company grand rapids michigan  2006-03-21
12:01:24
SESSION6: 1326      the child's wonderland company grand rapids michigan  2006-03-21
12:01:59
SESSION7: 1326      the childs wonderland co.  2006-03-21 21:20:42
SESSION7: 1326      the child's wonderland co. 2006-03-21 21:22:16
```

Figure 2: A sample from the AOL log divided into sessions

Evidently, the 30 minute cut-off is only a heuristic, as it can indicate two sessions where one would perhaps be more reasonable (e.g. sessions 3 & 4), and vice-versa. During a given session a user may work on multiple tasks, such as researching holiday destinations whilst simultaneously comparing alternative weather forecasts. Likewise, a given task may be performed across multiple sessions [4, 5] or across multiple channels such as desktop and mobile [17].

To find patterns in the AOL log, we need to first decide what features to extract. For example, we might use features such as the following (the citations indicate researchers who have successfully exploited them in published work):

1. **Query length** (Stenmark 2008, Wolfram 2007): the mean number of terms per query
2. **Session duration** (Chen & Cooper 2001, Stenmark 2008, Weber & Jaimes 2011): the time from first action to session end
3. **Number of queries** (Chen & Cooper 2001, Stenmark 2008, Wolfram 2007, Weber & Jaimes 2011): the number of keyword strings entered by the user
4. **Number of viewed hits / items retrieved** (Chen & Cooper 2001, Stenmark 2008, Weber & Jaimes 2011): the number of links clicked by the user
5. **Number of requested result pages** (Stenmark 2008): the number of explicit page requests
6. **Number of activities** (Stenmark 2008): the sum of all the interactions with the system

These features are all reasonably straightforward to extract from a log such as AOL using Python or any other scripting language, and the output can be represented as a set of feature vectors for each session, as shown in Figure 3.

```
1.00,172,2,0,0,2
1.00,0,1,0,0,1
1.00,0,1,0,0,1
0.00,0,0,0,1,1
3.00,51,1,0,1,2
5.00,709,3,1,2,6
4.00,94,2,0,0,2
```

Figure 3: A set of feature vectors from the AOL log

There are of course a great many other features we could extract or derive from this log, and the process of feature selection is something to which careful thought should be given. But for now let us consider another key question: What are the most effective techniques for revealing common patterns in such data? One such approach is to use an unsupervised learning technique, such as clustering [2, 6, 7].

## Using clustering to find patterns

In unsupervised learning there is no 'right answer', in the sense that different outputs can be obtained and no single output is necessarily more 'correct' than the others (although the patterns should be shown to be stable and repeatable, as discussed below). Instead, the utility of a particular clustering depends on the extent to which it provides useful insight into the phenomenon of interest, e.g. user behaviour. Consequently, it is prudent to adopt an exploratory strategy at the outset and try a range of approaches. A machine learning platform such as Weka [18], for example, provides a useful starting point as it offers an extensive range of algorithms for unsupervised learning and a helpful (but somewhat less polished) set of routines for result visualisation.

Let's start by taking a random sample of 100,000 sessions from the AOL log, and then applying feature scaling so that values are normalised over a given range, e.g. a standard deviation of 1 and mean of zero. This ensures that features with particularly wide ranges in the raw data do not disproportionately influence the clustering process [12]. We can then use Weka to apply a clustering algorithm such as Expectation Maximization [19] and then visualise the results, to give us a chart like that shown in Figure 4.

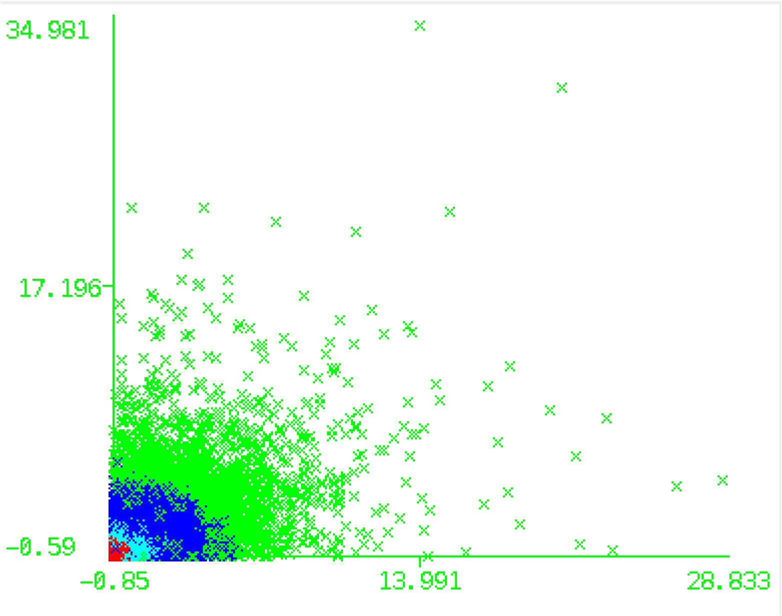

Figure 4: 100,000 AOL sessions, plotted as queries vs. clicks

This reveals four clusters, projected onto a two-dimensional space defined by the number of queries and the number of clicks. Note that there is nothing special about the number four; if you take a different log or use different features you'll almost certainly get a different result. So it is important to perform multiple iterations, to ensure that the patterns are stable, i.e. replicable across different samples from the same population.

Let’s focus for a moment on the image above: what does it tell us? Well, not a lot so far: the presence of 4 clusters may be significant but the projection doesn’t deliver much insight into that. Instead, we need a chart which illustrates how the individual features vary, such as that shown in Figure 5.

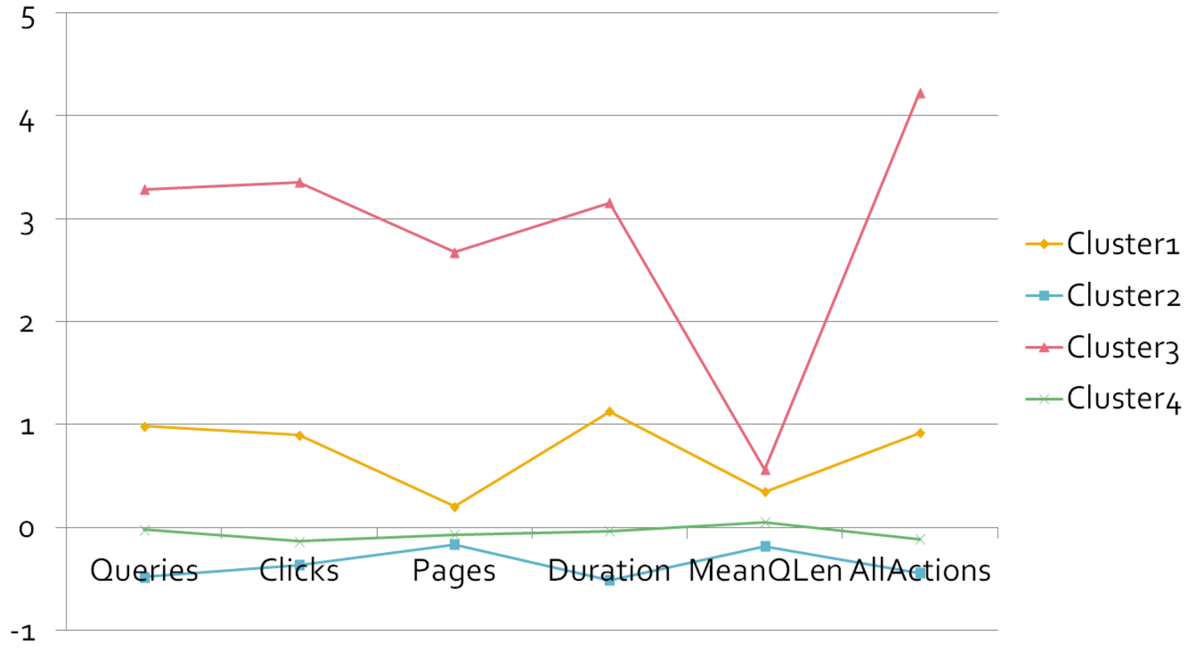

Figure 5: Four clusters based on six features

This shows how the mean values for the four clusters vary across the six features. But now it’s the data that is somewhat underwhelming: it suggests that we have a group of users who are relatively active (i.e. demonstrate a lot of interactions), a group who do relatively little and two other groups in between. This is hardly the kind of insight that will have market research professionals fearing for their jobs.

So at this point it may be prudent to review our choice of features: after all we have just six so far that were selected in a relatively informal manner. To be maximally useful (e.g. indicative of a latent variable such as ‘behaviour type’), we’d want them to be relatively independent, i.e. uncorrelated with each other. But if we run a Pearson correlation across the ones above we find the opposite: most of them are actually highly correlated, particularly pairs such as ‘All Actions’ and ‘Duration’. In this instance, these features may be ‘drowning out’ interesting patterns which we might otherwise see. So let’s try dropping ‘All Actions’ and adding in two new features, which are less likely to be correlated with overall activity levels:

- **Term use frequency** (Wolfram 2008): the mean frequency of usage of each query term within the session
- **Query interval** (Wolfram 2008): the mean time between query submissions within a session

Now the results seem a bit more interesting (Figure 6).

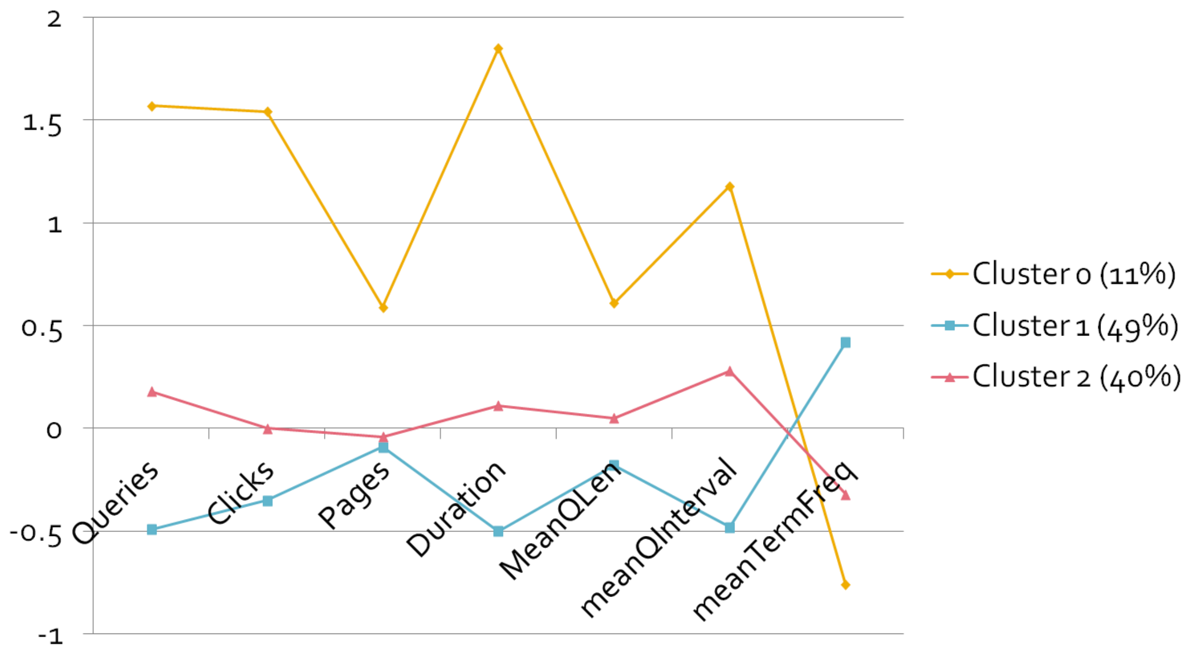

Figure 6: Three clusters based on seven features

We could interpret these features as follows:

- Cluster 0 seems to be a relatively small group users who are highly engaged, in long sessions but issuing queries that are diverse / heterogeneous (possibly changing topic repeatedly)
- Cluster 1 seems to be a large group of users who engage in short sessions but often repeat their query terms and do more paging than querying
- Cluster 2 seems to be a middle or 'general' group, whose defining characteristics we'll know more about when we've extracted more features (see below).

## Replication and validation

From this point there is evidently a number of different ways we could extend the analysis. But instead of just adding more features & applying ever more exotic visualisations to new data sources, it is perhaps more prudent to take a moment to reflect on the process itself and confirm that what we are doing really is valid and repeatable. One way to do this is to replicate other published studies and compare the results. For example, Wolfram's (2008) paper used the following features [2]:

1. **Session size** – the number of queries submitted for the session
2. **Terms used per query** – the average number of terms used per query over the session
3. **Term popularity** – the average frequency of usage of each query term within the session when
4. **Term use frequency** – the average frequency of usage of each query term within the session
5. **Query interval** – the average time between query submissions within a session
6. **Pages viewed per query** – the average number of page requests per query within a session

He applied these to a number of search logs and showed (among other things) evidence of four distinct behavioural patterns in a sample of ~65,000 web search sessions from the Excite 2001 data set [20]. If we can replicate his results, then we not only vindicate the specific conclusions he reached, but more importantly, provide evidence that our approach is valid and scalable to new data sources and research questions.

When we apply EM using this feature set to the AOL data, we find four clusters as he did, but the patterns are very different (note that display order of the features is changed here to facilitate comparison with Wolfram's results, and their labels have been simplified). But crucially, the results aren't even replicable within themselves: a further three samples of 10,000 sessions produces widely different outcomes (5, 6 and 7 clusters respectively). Even increasing the sample size to 100,000 seems to make little difference, with 7, 13, 6 and 6 clusters produced on each iteration (despite the suggestion in Wolfram's paper that subsets of 50k to 64k sessions should produce stable clusters).

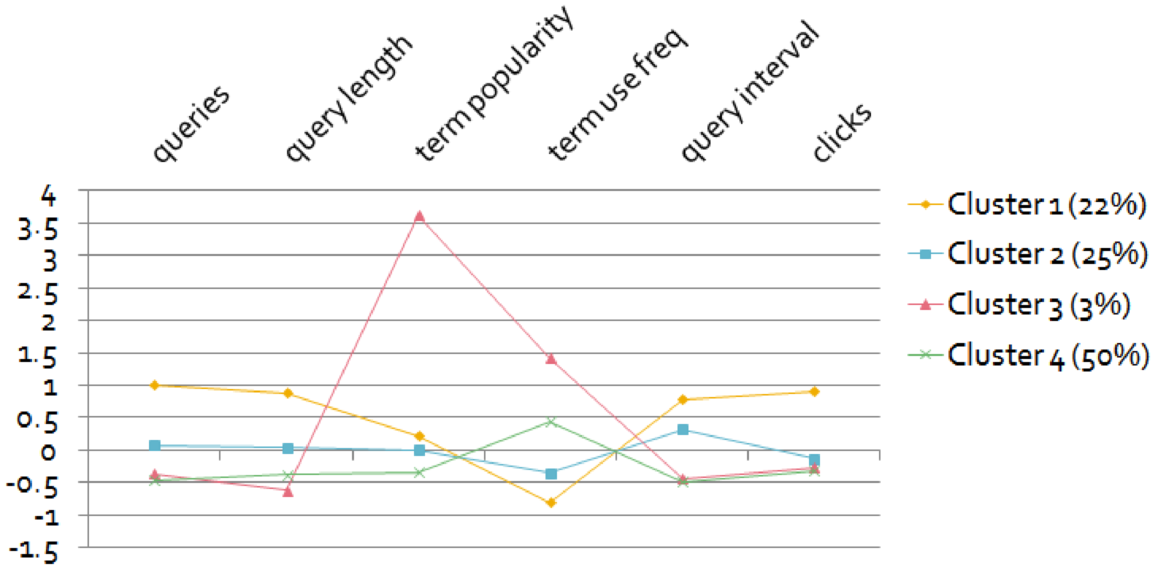

Figure 7: Applying EM using Wolfram's 6 features to 10,000 sessions from AOL

So why are we seeing such different results? One interpretation may be of course that these insights are indeed an authentic reflection of changes in user behaviour due to differences in context (e.g. a different search engine, time period, demographic, etc.)

But let's pause for a moment and examine the pattern in more detail. There is something very odd happening with term popularity now: we see a small cluster (just 3% of the sessions) where this feature seems to be something of an outlier, compressing the remaining traces into a narrow band. Indeed, the phenomenon becomes even more pronounced when we take a sample of 100,000 sessions (Figure 8).

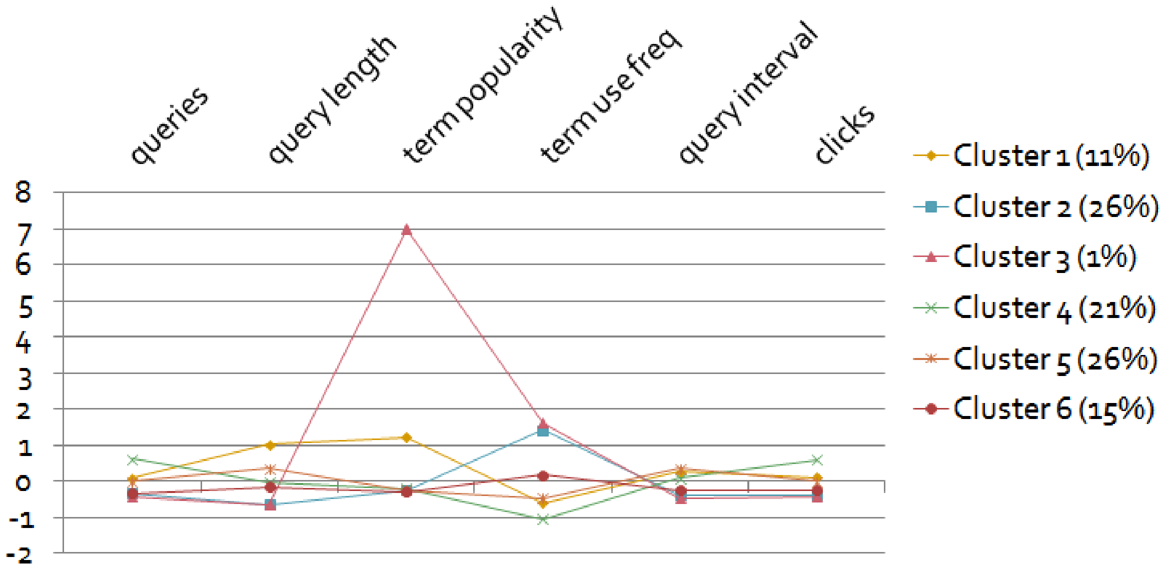

Figure 8: Applying EM using Wolfram's 6 features to 100,000 sessions from AOL

Perhaps this is an artefact of the clustering algorithm? Let's try XMeans [21] instead (which is a variant of kMeans [22] where the value for k is determined automatically). In this iteration, XMeans finds a local optimum at k=10, so the number of clusters is different. But the overall pattern, with a small cluster (1% of sessions) representing outlier values for term popularity is again clearly visible (Figure 9):

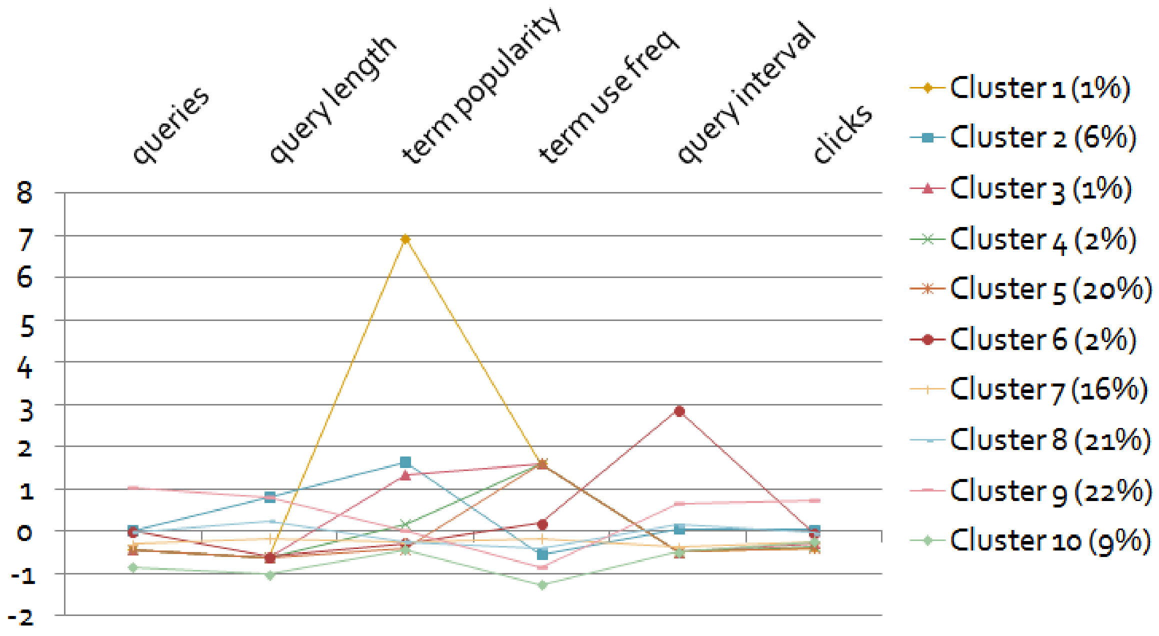

Figure 9: Applying XMeans (k<=10) and Wolfram's 6 features to 100,000 sessions from AOL

So something else must be at play. It turns out that there is indeed an artefact of the data which is causing this: there are a small number of sessions which contain just a single query, consisting solely of the character '-'. Precisely why they are present in the log is a matter for speculation: they may have been the default query in some popular search application, or a side-effect of some automated service or API, etc. But sessions like these, along with other robot-generated sessions, aren't generally helpful when trying to understand human behavioural patterns. Instead, they are best removed prior to analysis. Of course, there are no 100% reliable criteria for differentiating robot traffic from human, and what should be removed is a matter for judgement, often on a case-by-case basis [23]. In this case, including these single character queries appears to be counter-productive.

So now, with a new sample of 100,000 sessions excluding these outlier queries, we see EM produce a very different output (Figure 10).

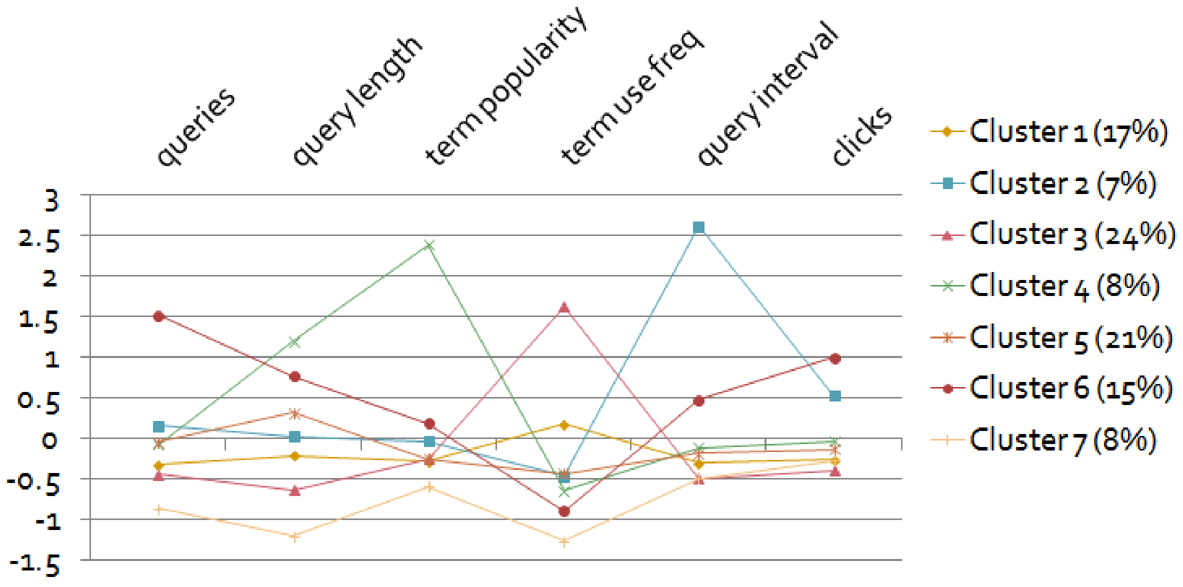

Figure 10: Applying EM and Wolfram's 6 features to 100,000 filtered sessions from AOL

This pattern is much more stable, with four iterations producing 7, 7, 7 and 9 clusters respectively. At this point we can start to speculate on what these patterns may be telling us. For example:

- Cluster 6 appears to be a group of users that engage in longer sessions, with many queries and many page views (clicks), but few repeating terms
- Cluster 4 appears to be a smaller group who seem to specialise in relatively long but popular queries (an odd combination!), also with few repeating terms
- Cluster 3 appears to be a relatively large group who make greater use of repeated terms, but are otherwise relatively unengaged (with shorter sessions and fewer page views)

And so on. Evidently, the patterns above are somewhat hard to interpret due to the larger number of clusters and lines on the chart. What would happen if we tried to determine the optimum number ourselves, rather than letting XMeans find one for us? One way of investigating this is to specify different values for k *a priori*, and see how the within-cluster sum of squared errors (which is calculated by Weka as part of its output) varies on each iteration. For example, varying k from 2 to 10 gives us the function shown in Figure 11.

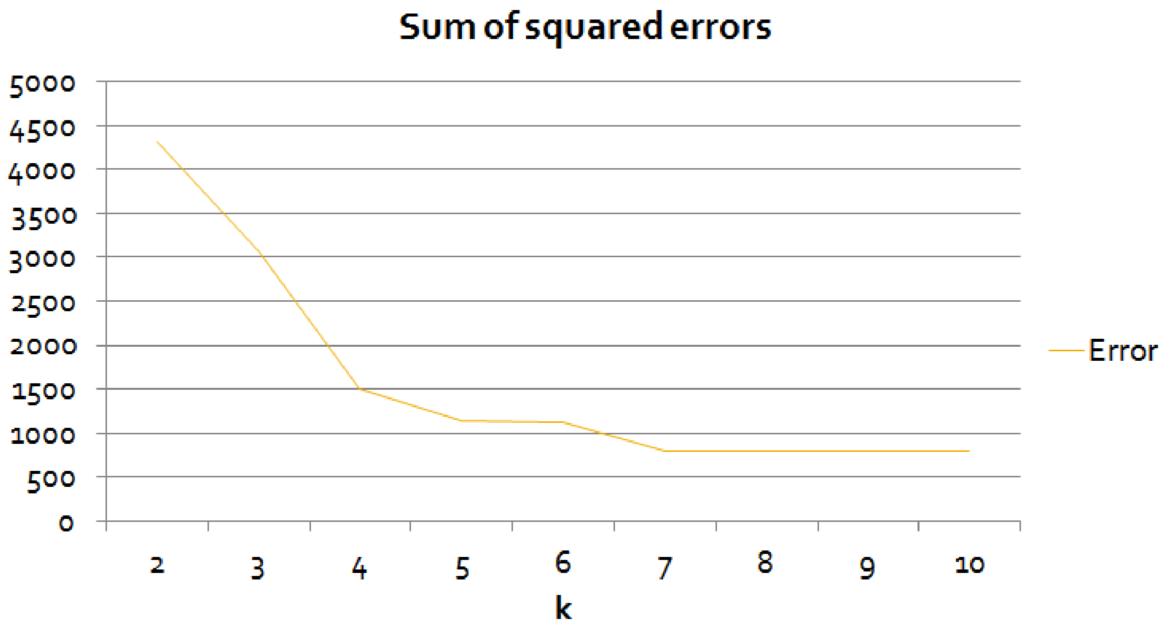

Figure 11: Sum of squared errors by k for 100,000 filtered sessions from AOL

As we can see, there is an 'elbow' around k=4 and another around k=7. This implies that either of these two values may be good choices for a local optimum. We've already seen the output for k=7 (which is the optimum that xMeans found), so now let's try kMeans with k=4 (Figure 12).

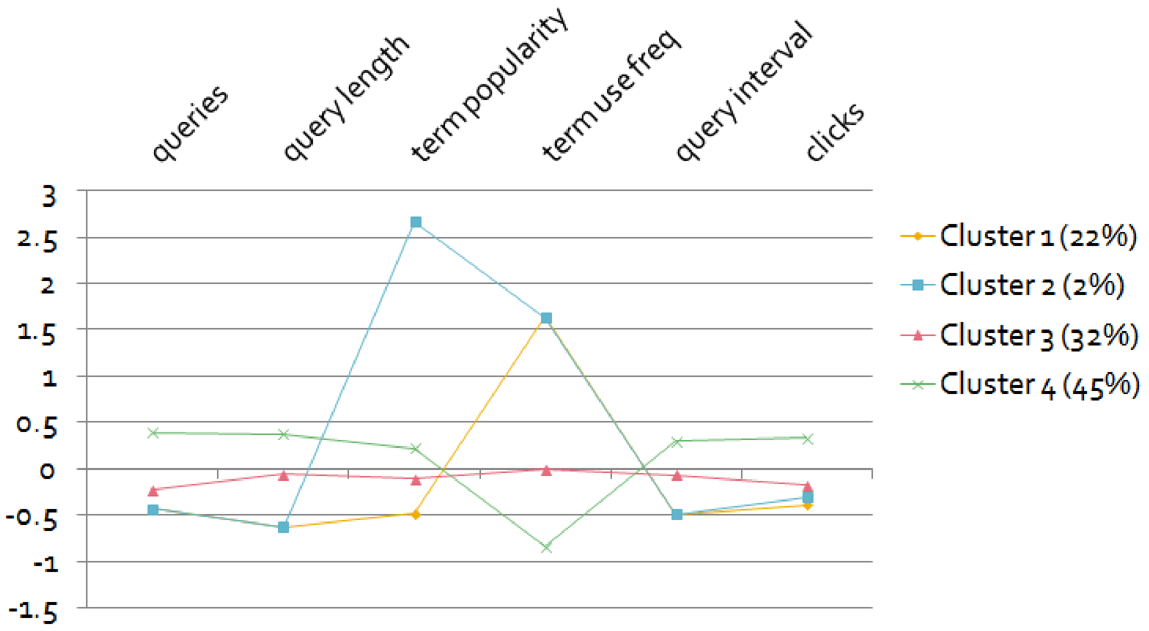

Figure 12: Applying kMeans (k=4) and Wolfram's 6 features to 100,000 sessions from AOL

This time the groups are somewhat easier to differentiate. For example, we might suggest that:

- Cluster 3 represents a baseline or 'generic' cluster of users who hover around the average for all measures

- Cluster 4 represents a relatively large group of users who engage in longer sessions (with more queries and page views) but are diverse in their interests, with few repeated terms
- Cluster 1 represents a smaller group who are the converse to cluster 4, engaging in shorter sessions but with more repeated terms
- Cluster 2 represents a tiny group (2%) of users who are similar to cluster 1 but focus on highly popular queries

Evidently, there are other ways we could analyse this data, and there are other ways we could interpret the output. What we've presented above is really just a starting point for exploration. So for now, let's draw some of the threads together and review what we've learnt.

## Conclusions

- **Start with the end in mind**: Mining search logs for usage patterns is inherently an exploratory activity, so it makes sense to consider a number of different approaches. In each case, the utility of the results should be judged by the extent to which they provide useful insight into the phenomenon of interest, e.g. user behaviour.
- **There is no 'right' answer**: As in many investigations of naturalistic phenomena, there is a tendency to look for patterns that align with our expectations. However, those expectations themselves are a subjective, social construct. The fact that we can produce multiple interpretations of the same data underlines the need for a common perspective when comparing patterns in search logs, and to apply recognised models of information seeking behaviour when interpreting the outputs.
- **Look for orthogonal features**: The hypothesis underlying much search log analysis of this type is that 'behaviour type' is a latent variable whose influence can be measured indirectly via features such as those above. Therefore, to be maximally useful these features should be highly correlated with the behaviours we wish to observe, but relatively independent of each other.
- **Apply Occam's razor**: It is tempting to select features based on whatever a particular data source offers, and include as many as possible in the learning process. But not all are equally useful, and some can indeed 'drown out' the influence of more important signals. So rather than starting from what the data can offer, identify the information seeking behaviours you'd like to explore, and try to find the features that most closely align with them.
- **Replicate to validate**: As researchers, our instincts are to explore the unknown, to solve the unsolvable, and to favour novelty over repetition. But sometimes it befits us to focus on the replication: by applying new techniques to old data, we validate our methodology and build a more reliable baseline for our own experimental work.

## References

1. Jansen, B. J. (2006). Search log analysis: What is it; what's been done; how to do it. *Library and Information Science Research*, 28(3): 407–432.
2. Wolfram, D., Wang, P., and Zhang, J. (2008). Modeling Web session behaviour using cluster analysis: A comparison of three search settings, In *Proceedings of the American Society for Information Science and Technology*, 44(1): 1550-8390.

3. Weber, I., and Jaimes, A. (2011). Who uses web search for what: and how? In *Proceedings of the fourth ACM international conference on Web search and data mining (WSDM '11)*. ACM, New York, NY, USA, 15-24.
4. Spink, A., Park, M., Jansen, B.J., and Pedersen, J. (2006). Multitasking during web search sessions. *Information Processing and Management, 42(1): 264-275.*
5. Jones, R., and Klinkner, K.L. (2008). Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proceedings of the 17th ACM conference on Information and knowledge management (CIKM '08)*. ACM, New York, NY, USA, 699-708.
6. Chen, H-M., and Cooper, M.D. (2001). Using clustering techniques to detect usage patterns in a web-based information system. *Journal of the American Society for Information Science and Technology*, 52(11): 888–904.
7. Stenmark, D. (2008). Identifying clusters of user behavior in intranet search engine log files. *Journal of theAmerican Society for Information Science and Technology,* 59(14): 2232-2243.
8. D. He and A. Goker (2000). Detecting session boundaries from Web user logs. Proceedings of the BCS-IRSG 22nd annual colloquium on information retrieval research, pp 57–66.
9. Kellar, M., Watters, C., and Shepherd, M. (2007). A field study characterizing Web-based information seeking tasks. *Journal of the American Society for Information Science and Technology*, 58(7): 999-1018.
10. Li, Y., and Belkin, N.J. (2008). A faceted approach to conceptualizing tasks in information seeking. *Information Processing and Management*, 44(6): 1822-1837.
11. Broder, A. (2002). A taxonomy of web search. *SIGIR Forum*, 36(2): 3-10.
12. Rokach, L., Maimon, O. (2005) Clustering Methods. *The Data Mining and Knowledge Discovery Handbook*, 321-352.
13. http://www.elasticsearch.org/overview/kibana/
14. http://www.lucidworks.com/lucidworks-silk/
15. G. Pass, A. Chowdhury, C. Torgeson, "A Picture of Search" *The First International Conference on Scalable Information Systems*, Hong Kong, June, 2006.
16. http://en.wikipedia.org/wiki/AOL_search_data_leak
17. Wang, Y., Huang, X., & White, R. W. (2013). Characterizing and supporting cross-device search tasks. In *Proceedings of the sixth ACM international conference on Web search and data mining* (pp. 707-716). ACM.
18. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.
19. http://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm
20. Spink, A., Jansen, B. J., Wolfram, D., and Saracevic, T. 2002. *From E-sex to E-commerce: Web Search Changes*. IEEE Computer. 35(3), 107 – 111.
21. http://weka.sourceforge.net/doc.packages/XMeans/weka/clusterers/XMeans.html
22. http://en.wikipedia.org/wiki/K-means_clustering
23. Zhang,Y., Moffat, A. (2006) Some Observations on User Search Behavior. Proceedings of the 11th Australasian Document Computing Symposium, pp. 1-8.