Supervised sampling for clustering large data sets

IOANNIS KOSMIDIS
CRiSM and Department of Statistics, University of Warwick
Coventry, CV4 9EL, UK
I.Kosmidis@warwick.ac.uk

DIMITRIS KARLIS
Department of Statistics, Athens University of Economics and Business
76 Patision Str, Athens, 10434, Greece
karlis@aueb.gr

June 12, 2010

Abstract

The problem of clustering large data sets has attracted a lot of current research. The approaches taken are mainly based either on the more efficient implementation or modification of existing methods or/and on the construction of clusters from a small sub-sample of the data and then the assignment of all observations in those clusters. The current paper focuses on the latter direction. An alternative supervised procedure to create the clusters is proposed. For learning the clusters, the procedure is using subsets of the data which are still constructed via sub-sampling but within partitions of the observation space. The general applicability of the approach is discussed together with tuning the parameters that it depends on to increase its ability. The procedure is applied to clustering the navigation patterns in the msnbc.com database.

*K*eywords: sub-sample; partition; hard clustering; clustering click-stream data

1 Introduction

In recent days, mainly due to the automatization in data collection procedures, the size of the data sets has dramatically increased posing challenges to their statistical treatment. Typical examples of such data sets are MRI pictures, transaction data from super markets and web-usage data, where millions of observations are available. In this respect, clustering methods can be very helpful to summarize these large data sets into few sets of equivalent, in some appropriate sense, observations or to identify useful patterns. However, most of the classical clustering methods (such as k-means, hierarchical clustering methods and model-based approaches and their variants) have been designed to cope with relatively small data sets and their scaling to huge data sets is not obvious or efficient.

Several of the approaches that have been suggested to reduce the computational burden in clustering huge data sets try to optimize the existing procedures through more efficient use of the data (see, for example, Bradley et al., 1998b; Chiu et al., 2001; Fraley et al., 2005). Others proceed by either selecting better starting values (for example Coleman and Woodruff, 2000; Ng and McLachlan, 2003) or by attempting a better usage of memory and computer resources (see Bradley et al., 1998a,b; Farnstrom et al., 2000). Some other ideas applicable in hierarchical clustering can be seen in Tantrum et al. (2004) and Vijaya et al. (2004). See, also Fraley et al. (2005) for an incremental algorithm.

Another class of approaches to clustering large data sets is to construct the clusters based on a small random sub-sample of the data. This approach dates back to Kaufman and Rousseeuw (1986) (CLARA algorithm). The key idea is to run the selected clustering algorithm to several random sub-samples of the data and then decide, according to some optimality criterion, which of the derived cluster configurations to use for the classification of the rest of the data. Due to their simplicity such approaches can be used to provide starting points to more sophisticated clustering methods (as is done for example in Steiner and Hudec, 2007, in order to construct a large numbers of prototype clusters which are subsequently combined into fewer ones using a more sophisticated model-based approach).

However, random sub-sampling in combination with hard clustering algorithms can return unstable results (see, for example Posse, 2001; Wehrens et al., 2004), especially when small groups are present in the data. A reason for this is that observations in the small groups possibly present in the data may not be selected via random sub-sampling.

To overcome such drawbacks, in the current paper, a model-free, supervised way of selecting subsets of the data is proposed, which aims to be robust to the deficiencies of random sub-sampling for clustering applications. The suggested procedure uses random sub-sampling and some k-medoids clustering method but within partitions of the observation space. In this way, small groups can be slightly over-represented in the resulting subsets and thus usual clustering algorithms can reveal their existence. Furthermore, the proposed procedure depends on several user-defined parameters and offers a way to take a finite number of subsets from the data and decide which is the best to use according to some optimality criterion. Finally the procedure applies to diverse types of data and clustering algorithms.

The remaining of the paper proceeds as follows: Section 2 describes the main idea and the contribution of the paper. In Section 3, the proposed procedure is applied on a simulated data set and Section 4 is concerned with the currently fashionable problem of web-clustering, where huge data sets are involved. Specifically, users are clustered according to their browsing behavior. Finally, a discussion and concluding remarks can be found in Section 5.

2 The proposed procedure

2.1 Motivation

When clustering large data sets, a common approach is to take a much smaller subset of the observations and use it for the necessary inferences, avoiding, in this way, the computational issues that arise when using all the observations in the data set. The most common approach is random sub-sampling (for example, as in the CLARA procedure). However, in this way the observations in small groups have correspondingly small probability of being selected, which might result in failure of detecting those small groups.

For example, consider an artificial setting of 504000 simulated observations of two real-valued characteristics A and B as shown in the left plot of Figure 1. The data form 9 well-separated groups: 5 groups with 100000 observations each (cyan) and 4 groups with 1000 observations each (red). The CLARA procedure is applied to this data for the construction of 9 clusters, taking, for robustness, 400 sub-samples of size 1000 each.

The resulting clustering is shown in the right plot of Figure 1. Clearly, despite the simplicity of the structure of the data and the large number of random sub-samples used, CLARA fails to correctly detect the true groups, breaking up the large groups and fusing the small groups into large clusters.

A different approach which would allow small groups to be slightly over-represented in the subsets could enable the clustering algorithms to construct corresponding small clusters. This can be done by partitioning the observation space and then retrieving random sub-samples within the partitions, because then the probability that an observation is selected in any partition may differ between observations.

2.2 Data-dependent partitions of the observation space

Let $x_1, \ldots, x_N$ be N observations belonging to an observation space Ω and define an appropriate, positive-valued measure $d(.,.)$ for measuring the dissimilarity between any two observations. Furthermore, let $\hat{\mu}$ be an element of Ω which, in some appropriate sense, is located centrally relative to the observations $x_1, \ldots, x_N$. For example, in Euclidean spaces, $\hat{\mu}$ may be set as the mean vector of $x_1, \ldots, x_N$.

Denote by $x_{(1)}, \ldots, x_{(N)}$ the observations sorted in increasing distance from $\hat{\mu}$ and without loss of generality, suppose that the observation space is Euclidean. Consider the creation of $h-1$ concentric hyper-spheres $F_1, \ldots, F_{h-1}$ with center $\hat{\mu}$ and radii $r_1 \leq \ldots \leq r_{h-1}$, respectively, which are defined by the data as

$$r_i = \max_{j \in \{1,\ldots,i[N/h]\}} d\left\{x_{(j)}, \hat{\mu}\right\} \quad (i = 1, \ldots, h-1),$$

where $[N/h]$ is N/h rounded to the closest integer. Furthermore, set $r_h = +\infty$.

Using the above construction, the observation space may be partitioned in partitions $I_1, \ldots, I_h$, where

$$\begin{aligned} I_1 &= F_1 \\ I_i &= F_i \cap F_{i-1}^c \quad (i = 2, \ldots, h), \end{aligned}$$

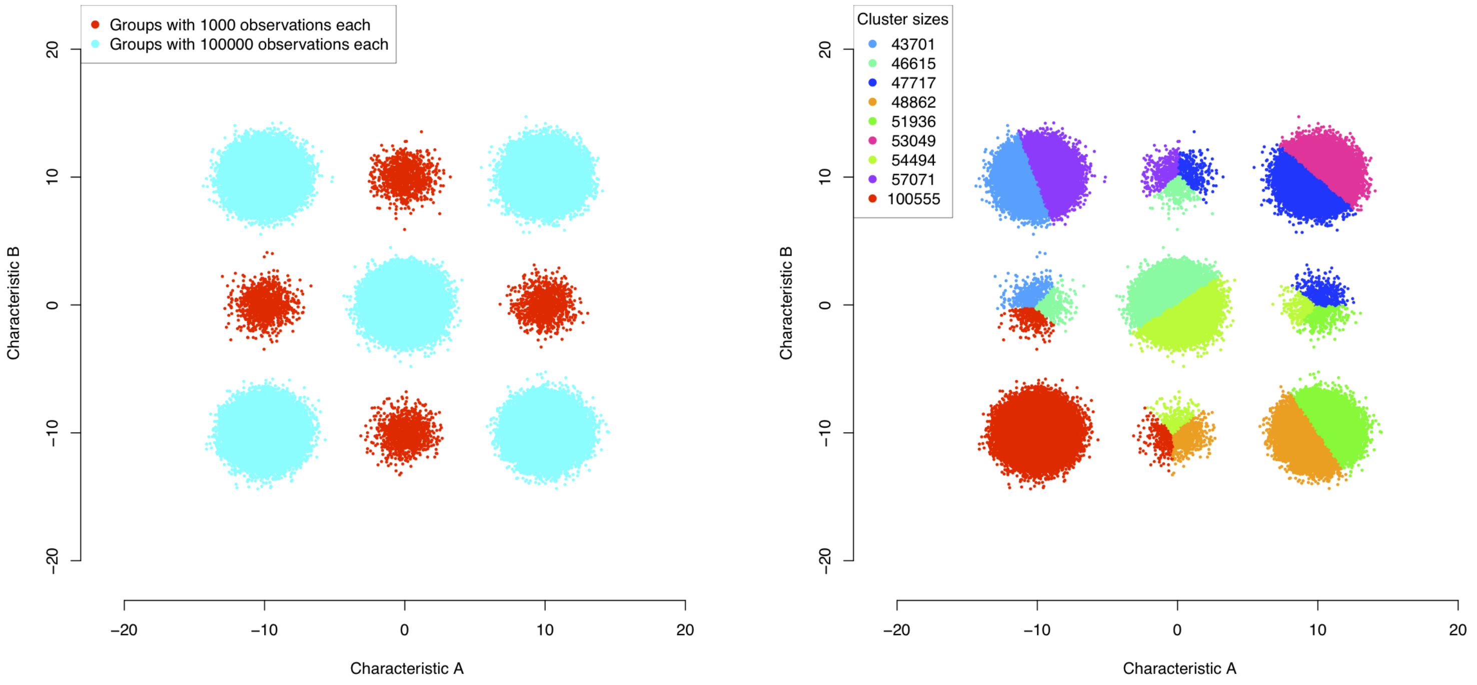
with $F_h = \Omega$.

Such a partitioning scheme depends exclusively on the observed data and results in partitions with $[N/h]$ observations each, where the last partition may contain more or fewer observations than the others depending on the value of N/h.

In this way, letting $r_0 = 0$, the observations x_j with $j \in D$, $D \subset \{1, \ldots, N\}$ are equivalent if and only if $r_{i-1} \leq d(x_j, \hat{\mu}) < r_i$ for all $j \in D$, for some $i = 1, \ldots, h$.

The above equivalence relations motivate the treatment of each partition independently. Random sub-sampling without replacement can be used to obtain s sub-samples of size n from each partition and a k-medoids clustering algorithm (for example the PAM algorithm in Kaufman and Rousseeuw (1990)) can be applied to each sub-sample to obtain c centroids. The resultant centroids may be thought as summarizing the information of the observations in each partition in terms of minimum average distance from the centroids, and thus the set of centroids from all partitions (much smaller in size than the whole data set) can be used to summarize the information contained in the original data. Then, a clustering algorithm is applied to this set of centroids and all observations in the data set are classified in the resultant clusters.

Figure 1: Left: Simulated data consisting of 504000 observations of two real-valued characteristics A and B. The data form 9 well-separated groups where the red groups contain 1000 observations each and the cyan groups 100000 observations each. Right: The resulting clustering from the application of CLARA with 400 sub-samples of size 1000 each.



2.3 Specification of the parameters h, s, n and c

The above construction requires the specification of the parameters h, s, n and c. If $h = 1$, the procedure is equivalent to obtaining all the centroids that result from the application of k-medoids with c centroids on each one of s random sub-samples with size n, where the sub-samples are taken from all the available data points. This, in turn, is similar — but not the same — to collecting the resulting centroids in the iterations of CLARA and using them as a representative subset of the data. A systematic way of specifying the values of those parameters may be constructed by noting that the number of elements sampled in the aforementioned elementary setting can serve as the reference amount of information for the definition of possible partitioning schemes.

Suppose that the resulting subsets are used for clustering the data in $m \leq M$ clusters. First, note that the size of the resultant subset of observations is hsc. The total number of elements that are sampled during the procedure is $k = hsn$. Thus, specifying k we can enumerate all possible combinations of positive integers h, s, n and c, for which $k = hsn$ is solved under the natural constraints

$$\begin{aligned} hsc &\geq M\,, \\ c &\geq 1\,, \\ n &\geq c\,, \\ [N/h] &\geq n\,, \\ h &> 1\,, \\ s &\geq 1\,. \end{aligned} \tag{1}$$

The first constraint ensures that the maximum number of clusters to be constructed using the subset is smaller than the size of the subset.

Clearly, the solution set after defining k is either empty or finite, because h, s, n and c are all positive integers. Thus, fixing k to an appropriate value, the procedure described in the previous section can be applied for all possible quadruplets (h, s, n, c) for which $k = hsn$ is solved subject to the constraints (1), and the subset that gives the best results for the original data can be selected according to some application dependent optimality criterion. As is done in later sections, c can also be fixed at some integer value instead of allowing it to vary freely. In this way, the number of settings that need to be considered is greatly reduced.

2.4 Steps of the suggested procedure

After having chosen $\hat{\mu}$, the procedure involves the following steps:

A. Initialization

A.1 Calculate $d_i = d(x_i, \hat{\mu})$, $i = 1, \ldots, N$.

A.2 Sort d_i, $i = 1, \ldots, N$, in increasing order.

A.3 Order the observations according to the ordering of d_is and obtain the ordered sample $x_{(1)}, \ldots, x_{(N)}$.

B. Main iterations

For every (h, s, n, c) that solves $k = hsn$ under the constraints (1), the following steps are repeated t times.

B.1 Construct h observation sets

$$\begin{aligned} G_1 &= \left\{x_{(1)}, \ldots, x_{([N/h])}\right\}, \\ &\vdots \\ G_{h-1} &= \left\{x_{((h-2)[N/h]+1)}, \ldots, x_{((h-1)[N/h])}\right\}, \\ G_h &= \left\{x_{((h-1)[N/h]+1)}, \ldots, x_{(N)}\right\}. \end{aligned}$$

B.2 For $j = 1$ to h

B.2.1 Obtain s random sub-samples of size n from G_j.

B.2.2 Apply a k-medoids clustering algorithm with c centroids to each sub-sample.

B.3 Apply a clustering algorithm on the subset of the data consisting of all the centroids that were obtained in the previous steps.

B.4 Classify all the observations in the data set in the resultant clusters.

B.5 Calculate some appropriate optimality criterion for the clustering on all observations of the data set.

Note that step B.5 can be expensive for extremely large data sets. In such cases, steps B.4 and B.5 can be omitted, calculating the optimality criterion merely for the clustering of the subset that results in step B.3.

2.5 Computational considerations

For any specific choice of (h, s, n, c), the total number of calculations required for the application of the proposed procedure is $O(N) + r(n, s, h, c)$. The $O(N)$ term reflects the effort required for calculating and sorting $d(x_i, \hat{\mu})$ $(i = 1, \ldots, n)$ (which only needs to be done once for all available quadruplets (h, s, n, c)) and for the classification of the observations in step B.4 of the procedure. The term $r(n, h, s, c)$ is appearing due to the additional calculations required for the application of a k-medoids clustering algorithm to each of the s sub-samples of size n. Nevertheless, the extra effort $r(n, h, s, c)$ is not very large given that the clustering algorithm is applied to a small subset of the observations contained in a partition.

3 Application to simulated data

To demonstrate its benefits and functionality, the proposed procedure is applied to the example of Section 2.1. Interest is in detecting well-separated small groups, and thus the optimality criterion for the selection of the best clustering is chosen to be the maximum average silhouette width. The average silhouette widths are calculated for the clustering that results from the application of PAM with $m = 9$ centroids on the each subset of

Table 1: Average silhouette widths when clustering the subsets corresponding to all possible settings of h, s and n for $k = 1000$, $c = 6$ and $M = 9$. The procedure is repeated $t = 10$ times per setting. For each subset 9 clusters are constructed.

Setting	h	s	n	hsc	Average silhouette widths for 10 repetitions per setting									
					1	2	3	4	5	6	7	8	9	10
1	2	1	500	12	0.37	0.45	0.31	0.26	0.34	0.34	0.22	0.28	0.45	0.37
2	2	2	250	24	0.67	0.79	0.61	0.71	0.76	0.76	0.61	0.65	0.75	0.62
3	2	4	125	48	0.67	0.68	0.60	0.59	0.68	0.65	0.71	0.63	0.69	0.59
4	2	5	100	60	0.72	0.63	0.65	0.65	0.64	0.69	0.62	0.60	0.68	0.70
5	2	10	50	120	0.77	0.70	0.74	0.67	0.74	0.74	0.71	**0.83**	0.65	0.82
6	2	20	25	240	**0.83**	0.63	**0.83**	0.65	0.52	0.63	0.53	0.54	0.64	0.57
7	2	25	20	300	0.61	0.70	0.71	0.62	0.51	0.61	0.60	0.62	0.62	0.59
8	2	50	10	600	0.41	0.53	0.52	0.62	0.54	0.52	0.53	0.63	0.43	0.63
9	4	1	250	24	0.33	0.31	0.38	0.32	0.29	0.27	0.31	0.45	0.40	0.46
10	4	2	125	48	0.40	0.54	0.51	0.41	0.45	0.41	0.41	0.49	0.43	0.44
11	4	5	50	120	0.71	0.58	0.71	0.49	0.68	0.60	0.44	0.70	0.52	0.64
12	4	10	25	240	0.70	0.72	0.63	0.60	0.61	0.54	0.51	0.62	0.60	0.69
13	4	25	10	600	0.52	0.39	0.50	0.63	0.63	0.39	0.43	0.52	0.43	0.62
14	5	1	200	30	0.57	0.66	0.43	0.53	0.56	0.42	0.52	0.53	0.54	0.47
15	5	2	100	60	0.61	0.66	0.72	0.65	0.74	0.82	0.64	0.72	0.68	0.70
16	5	4	50	120	0.72	0.67	0.72	0.73	0.82	0.74	0.82	0.73	0.61	0.82
17	5	5	40	150	0.71	0.73	0.53	0.65	0.71	0.81	0.81	0.82	0.82	0.82
18	5	8	25	240	0.53	0.63	0.71	0.63	0.74	0.66	0.73	0.62	0.73	0.54
19	5	10	20	300	0.63	0.72	0.53	0.74	0.71	0.65	0.53	0.53	0.50	0.71
20	5	20	10	600	0.42	0.41	0.52	0.54	0.52	0.42	0.41	0.42	0.52	0.61
21	5	25	8	750	0.42	0.52	0.51	0.41	0.44	0.52	0.52	0.52	0.43	0.51
22	8	1	125	48	0.52	0.48	0.51	0.56	0.42	0.53	0.53	0.54	0.41	0.55
23	8	5	25	240	0.53	0.54	0.53	0.43	0.52	0.63	0.61	0.52	0.55	0.43
24	10	1	100	60	0.70	0.51	0.58	0.58	0.60	0.60	0.57	0.51	0.59	0.63
25	10	2	50	120	0.70	0.63	0.63	0.73	0.72	0.64	0.72	0.81	0.72	0.72
26	10	4	25	240	0.54	0.61	0.63	0.70	0.63	0.52	0.72	0.39	0.42	0.53
27	10	5	20	300	0.63	0.62	0.62	0.61	0.54	0.61	0.62	0.73	0.70	0.63
28	10	10	10	600	0.42	0.61	0.61	0.42	0.54	0.52	0.51	0.51	0.50	0.63
29	20	1	50	120	0.63	0.59	0.64	0.79	0.70	0.61	0.79	0.63	0.59	0.61
30	20	2	25	240	0.60	0.62	0.64	0.53	0.50	0.41	0.51	0.62	0.64	0.61
31	20	5	10	600	0.42	0.52	0.53	0.42	0.53	0.42	0.52	0.52	0.53	0.52
32	25	1	40	150	0.71	0.61	0.71	0.62	0.70	0.69	0.80	0.63	0.71	0.73
33	25	2	20	300	0.48	0.61	0.71	0.62	0.52	0.61	0.62	0.52	0.59	0.54
34	25	4	10	600	0.41	0.41	0.45	0.53	0.51	0.52	0.54	0.52	0.43	0.52
35	25	5	8	750	0.43	0.43	0.53	0.42	0.53	0.43	0.42	0.53	0.42	0.43
36	40	1	25	240	0.43	0.54	0.41	0.63	0.51	0.54	0.41	0.51	0.53	0.51
37	50	1	20	300	0.40	0.52	0.41	0.51	0.62	0.40	0.52	0.63	0.53	0.41
38	50	2	10	600	0.43	0.52	0.51	0.51	0.53	0.53	0.43	0.43	0.54	0.45
39	100	1	10	600	0.40	0.52	0.45	0.53	0.44	0.42	0.43	0.42	0.42	0.42
40	125	1	8	750	0.43	0.54	0.42	0.52	0.51	0.42	0.43	0.53	0.42	0.43

observations (thus steps B.4 and B.5 are omitted). The required centrally located point $\hat{\mu}$ is chosen to be the sample mean vector.

For $k = 1000$ and for $c = 6$, Table 1 shows the possible solutions of $k = hsn$ under the constraints (1) along with the size of the resultant subset of observations and the average silhouette widths (in two significant places) in $t = 10$ repetitions per setting. The settings that result in the largest average silhouette width are setting 5 ($h = 2$, $s = 10$, $n = 50$) in the seventh repetition and setting 6 ($h = 2$, $s = 20$, $n = 25$) in the first and third repetitions, where the average silhouette width has value 0.83 (those cases are shown in boldface type on Table 1). The corresponding clustering of the whole data set for these two settings is identical and is shown in Figure 2.

In this artificial example the assumed observation groups are fully reconstructed by merely using partitioning, random sub-sampling and a hard clustering algorithm. Furthermore, the total number of observations sampled for producing Table 1 and the effort required for the application of the PAM procedure, are both comparable to the corresponding requirements for the application of CLARA in Section 2.1.

As a final observation, note that settings 15, 16, 17 also result in average silhouette widths greater than 0.8 and the corresponding clusterings of all the observations in the data set are equally good.

4 A hard clustering of the msnbc.com data

4.1 Description of the data

The proposed procedure is used to obtain a clustering of the *msnbc.com* data set[1]. The data consist of records for all the users who visited *msnbc.com* on the entire day of 28th September 1999. For each user the ordered sequence of URL hits is recorded. Nevertheless, what is given is the category where each URL belongs to and not the URL itself. The URL categories are coded as 1: front page, 2: news, 3: tech, 4: local, 5: opinion, 6: on-air, 7: misc, 8: weather, 9: health, 10: living, 11: business, 12: sports, 13: summary, 14: bulletin board service, 15: travel, 16: msn-news, 17: msn-sports. For example, two observations in the data are

```
user A: 6 9 4 4 4 10 3 10 5 10 4 4 4
user B: 1 2 1 14 14 14 14 14 14 14 14 14 14 14 14 1 2 2
```

Thus, user A first visits a URL of category 6, then a URL of category 9, then 4, followed by another of category 4, and so on. The length of the sequences in the data varies from 1 to 14795 URL hits, which combined with the fact that there are 989818 sequences makes the construction of clusters challenging. An attempt can be found in Cadez et al. (2003) where mixtures of first-order Markov models are used to cluster these data; through an Expectation Maximization (EM) algorithm variant, the models therein are trained using a random sub-sample of 100023 sequences and the result is evaluated using another sub-sample of 98687 sequences. The conclusion of Cadez et al. (2003) analyses was that there are 60 to 100 clusters on the data.

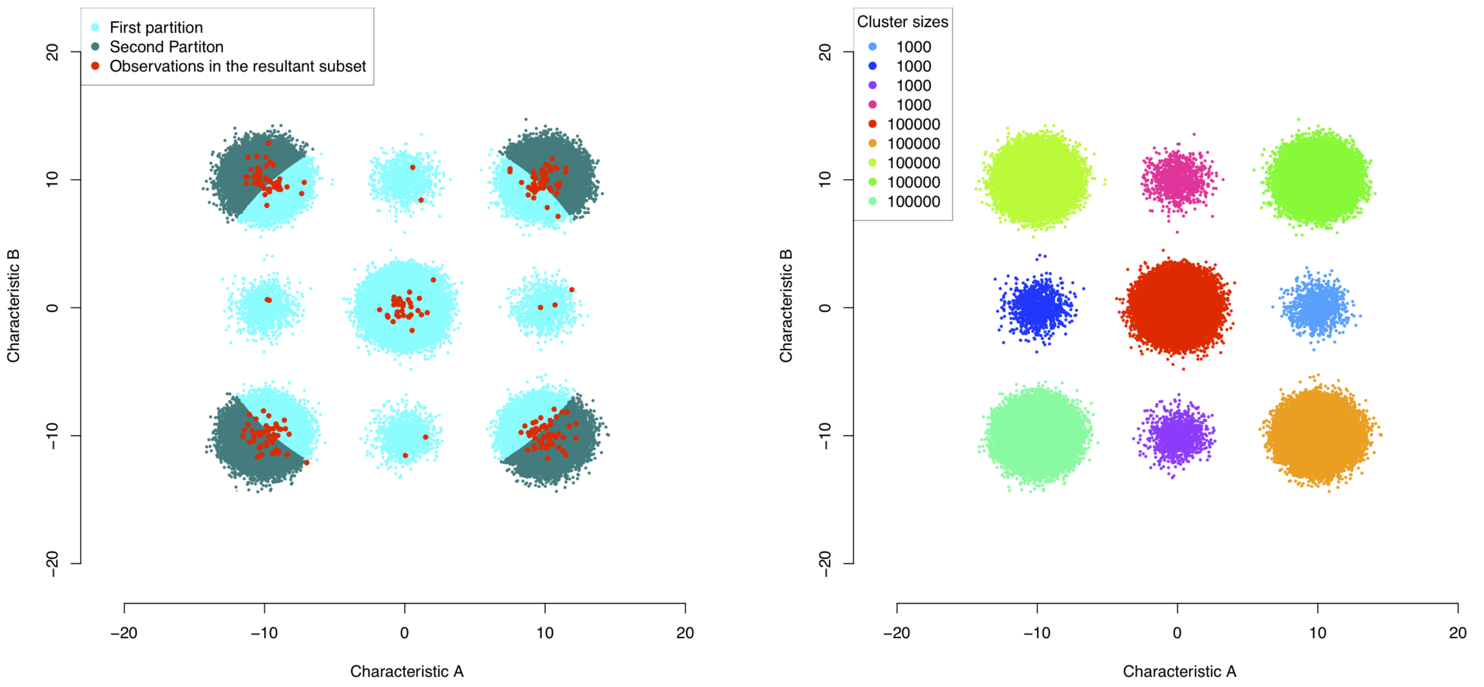
4.2 Clustering the msnbc.com data

In this section we attempt to group the msnbc.com data in $m \leq 20$ clusters using the proposed procedure with $k = 5000$ and $c = 10$. These parameter values result in 69 triplets (h, s, n) that solve $k = hsn$ under the constraints (1) and we choose to use only the 55 triplets that correspond to more than one subset per partition. The procedure is repeated $t = 10$ times for each setting. We have selected to work with up to 20 clusters mainly for illustration but also because a much larger number of clusters would have been hard to interpret and present.

The grouping of each subset in m clusters, $(m = 2, \ldots, 20)$ results by using the string edit distance for calculating dissimilarities (see, for example, Hay et al., 2004, for a description and the use of the string edit distance in the analyses of server-logs) and PAM.

[1] the data set is publicly available at http://kdd.ics.uci.edu/databases/msnbc/msnbc.html

Figure 2: The subset of observations and corresponding clustering of the data for the first repetition of setting 6.



The most dominant sequence in the data is `6` with 91805 occurrences (almost 9.2% of the data). Nevertheless, taking into account the nature of the string edit distance, the "centrally" located observation $\hat{\mu}$ in the data is set to the second most dominant sequence in the data, `1` (with 57552 occurrences), because the URLs of category "front page" are the most common in the msnbc.com data. The best clustering in m clusters, $m \in \{2, \dots, 20\}$, is chosen amongst at most 550 alternatives as the clustering that gives the smallest average distance from the medoids when all sequences in the data are assigned in the clusters (step B.5 of the proposed procedure). For example, in the construction of 11 clusters, Table 2 shows the average distances from medoids for the $t = 10$ repetitions per setting, in two significant places. The minimum value for the average distance from medoids is 3.31 and it is attained for several settings in Table 2. For example, for the setting 16 ($h = 4$, $s = 125$, $n = 10$) that value is attained in the third repetition. All those optimal settings produce very similar clusterings of the observations and so arbitrarily choosing one of them for each value of m does not affect the result. In this way, a list of clusterings in 2, 3, ..., and 20 clusters is constructed.

4.3 Assessing the number of clusters

For choosing an optimal number of clusters we use the concept of *open sequences*. An open sequence in the current context is defined as an ordered sequence of two URL categories or more, where the same category cannot occur two times consecutively (Büchner et al., 1999, used open sequences to describe navigational patterns). For example, the sequence $\{2, 14, 14, 1\}$ is not an open sequence because the URL category 14 appears two times consecutively. On the other hand, $\{2, 1, 14, 2\}$ is an open sequence and is contained in the sequence for user B in Subsection 4.1, because 2, 1, 14 and 2 appear in this specific order when we read the observed URL categories from left to right. The *support* of an open sequence is defined as the number of sequences in the data containing the open sequence divided by the total number of observations in the data set. The connection between open sequences and clusters is made via the *association rule "If the observation contains the open sequence then the observation belongs to cluster j"*. Then, the *confidence* of the above association rule (or, equivalently, the confidence of the open sequence in cluster j) is the number of observations in cluster j containing the open sequence divided by the total number of observations in the data containing it. High values for the confidence of an open-sequence in a cluster suggests that users containing that open sequence are highly associated with that cluster.

For each one of the optimal clusterings, the confidences of all l_ϵ open sequences that have support greater or equal than a threshold $\epsilon > 0$, are calculated. Then the functions

$$P(m) = \frac{m}{l_\epsilon(m-1)} \sum_{i=1}^{l_\epsilon} \sum_{j=1}^{m} w_{ij}(1 - w_{ij}) \, ,$$

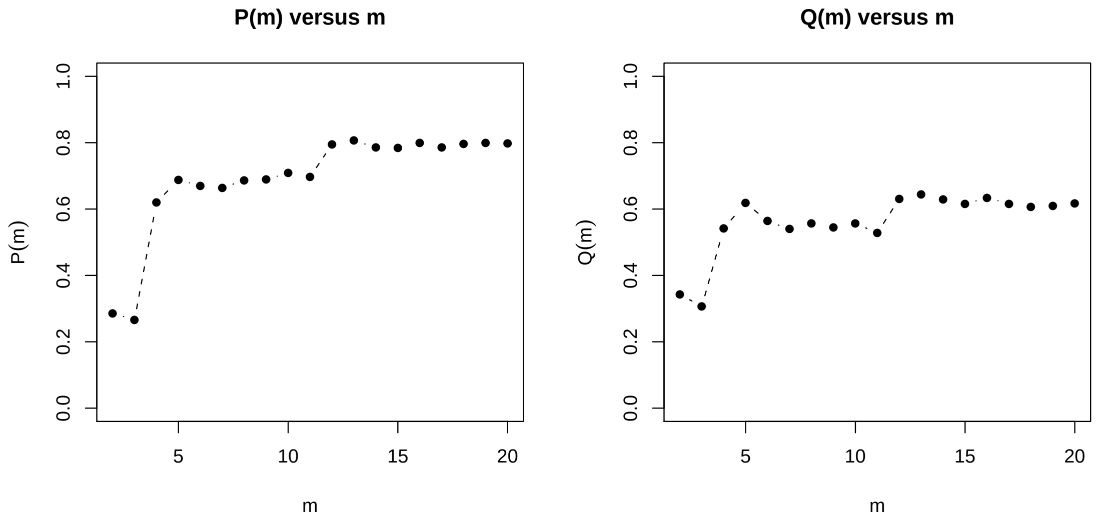
$$Q(m) = -\frac{1}{l_\epsilon \log m} \sum_{i=1}^{l_\epsilon} \sum_{j=1}^{m} w_{ij} \log w_{ij} \, ,$$

are evaluated at $m \in \{2, \dots, 20\}$, where w_{ij} is specific to each clustering in m clusters and denotes the confidence of the ith open sequence for the jth cluster ($i = 1, \dots, l_\epsilon$; $j = 1, \dots, m$). The dependence of w_{ij} to each clustering in m clusters is suppressed for notational simplicity. By the definition of confidence, $\sum_{j=1}^{m} w_{ij} = 1$ ($i = 1, \dots, l_\epsilon$). Note

Table 2: Average distance from medoids when clustering the subsets corresponding to the settings of h, s and n for $k = 5000$, $c = 10$ and $M = 20$. Only the settings with $s > 1$ are considered and the procedure is repeated $t = 10$ times per setting. For each subset 11 clusters are constructed.

Setting	h	s	n	hsc	Average distance from medoids for 10 repetitions per setting									
					1	2	3	4	5	6	7	8	9	10
1	2	2	1250	40	3.59	3.56	3.61	3.56	3.68	3.52	3.50	3.52	3.68	3.59
2	2	4	625	80	4.00	3.54	4.11	3.47	3.82	3.52	3.98	4.07	3.87	3.68
3	2	5	500	100	3.50	4.01	4.30	4.12	4.24	3.59	3.64	3.56	3.78	3.98
4	2	10	250	200	4.79	4.14	4.73	4.78	4.77	5.08	4.96	4.89	4.72	4.49
5	2	20	125	400	4.16	4.22	3.82	4.06	4.16	4.08	4.21	4.16	4.27	4.54
6	2	25	100	500	3.93	3.93	4.04	3.89	3.79	4.00	4.26	4.24	4.13	4.21
7	2	50	50	1000	3.96	3.58	3.84	3.58	3.66	3.56	3.64	3.73	3.72	3.57
8	2	100	25	2000	3.44	3.46	3.48	3.62	3.48	3.63	3.45	3.52	3.46	3.45
9	2	125	20	2500	3.54	3.52	3.35	3.37	3.55	3.51	3.40	3.42	3.55	3.59
10	2	250	10	5000	3.32	3.33	3.43	3.32	3.32	3.42	3.32	**3.31**	3.41	3.35
11	4	2	625	80	3.78	4.01	3.63	3.88	3.75	3.59	3.81	3.73	3.79	3.62
12	4	5	250	200	4.36	4.05	4.20	4.16	3.79	4.02	4.16	3.85	3.87	3.96
13	4	10	125	400	3.99	3.75	3.92	4.51	3.87	4.39	4.02	4.12	4.13	4.33
14	4	25	50	1000	3.61	3.45	3.73	3.51	3.77	3.56	3.77	3.72	3.72	3.69
15	4	50	25	2000	3.41	3.44	3.69	3.47	3.37	3.44	3.45	3.58	3.72	3.48
16	4	125	10	5000	3.33	3.42	**3.31**	**3.31**	**3.31**	3.33	3.33	3.32	3.32	3.33
17	5	2	500	100	4.05	3.60	3.62	3.78	3.72	3.77	3.90	3.69	3.66	3.71
18	5	4	250	200	4.18	4.17	3.91	3.93	3.94	3.75	4.07	4.25	3.79	3.74
19	5	5	200	250	4.84	4.14	3.75	4.13	4.00	4.10	3.89	4.19	3.80	3.81
20	5	8	125	400	4.11	4.01	3.87	4.03	3.87	3.75	3.83	4.15	4.05	3.89
21	5	10	100	500	3.77	3.78	3.73	3.83	3.64	3.84	3.87	3.83	3.97	3.83
22	5	20	50	1000	3.58	3.74	3.69	3.53	3.54	3.60	3.75	3.49	3.52	3.50
23	5	25	40	1250	3.59	3.64	3.79	3.47	3.41	3.74	3.67	3.79	3.49	3.70
24	5	40	25	2000	3.38	3.66	3.55	3.47	3.53	3.47	3.52	3.49	3.40	3.52
25	5	50	20	2500	3.43	3.40	3.44	3.34	3.34	3.47	3.48	3.34	3.42	3.34
26	5	100	10	5000	3.40	3.41	3.33	3.32	3.32	3.33	3.32	3.33	3.32	3.42
27	8	5	125	400	3.82	3.94	4.06	3.67	4.22	4.10	3.95	4.22	4.09	3.78
28	8	25	25	2000	3.41	3.45	3.40	3.62	3.45	3.40	3.53	3.44	3.44	3.35
29	10	2	250	200	3.68	3.81	3.72	3.74	3.62	3.63	3.60	3.74	3.75	3.77
30	10	4	125	400	3.98	4.32	3.79	3.92	3.83	3.70	3.64	3.73	3.65	4.32
31	10	5	100	500	3.83	3.71	3.68	4.03	3.59	3.67	3.55	3.71	3.68	3.81
32	10	10	50	1000	3.51	3.44	3.53	3.55	3.62	3.79	3.53	3.42	3.61	3.61
33	10	20	25	2000	3.74	3.40	3.44	3.43	3.45	3.43	3.52	3.40	3.56	3.43
34	10	25	20	2500	3.43	3.41	3.35	3.42	3.51	3.40	3.41	3.40	3.55	3.45
35	10	50	10	5000	**3.31**	3.41	3.40	3.32	3.32	3.33	3.36	3.32	3.33	**3.31**
36	20	2	125	400	3.73	3.77	3.78	3.55	3.82	3.64	3.62	3.56	3.58	3.61
37	20	5	50	1000	3.60	3.38	3.66	3.43	3.50	3.53	3.89	3.56	3.67	3.73
38	20	10	25	2000	3.42	3.53	3.34	3.54	3.67	3.37	3.32	3.42	3.38	3.36
39	20	25	10	5000	3.41	**3.31**	3.32	3.33	3.41	3.33	3.32	3.32	3.32	3.42
40	25	2	100	500	3.79	3.72	3.76	3.58	3.93	3.69	3.73	3.92	3.81	3.61
41	25	4	50	1000	3.40	3.53	3.42	3.55	3.34	3.45	3.54	3.64	3.69	3.75
42	25	5	40	1250	3.46	3.56	3.45	3.63	3.36	3.42	3.39	3.67	3.41	3.55
43	25	8	25	2000	3.41	3.64	3.43	3.33	3.43	3.50	3.34	3.44	3.43	3.32
44	25	10	20	2500	3.34	3.54	3.52	3.33	3.46	3.34	3.35	3.35	3.36	3.43
45	25	20	10	5000	3.33	3.32	3.41	3.41	3.43	3.50	3.33	3.42	3.33	3.34
46	40	5	25	2000	3.36	3.34	3.34	3.33	3.34	3.52	3.44	3.43	3.36	3.37
47	50	2	50	1000	3.43	3.56	3.56	3.49	3.56	3.51	3.70	3.54	3.58	3.46
48	50	4	25	2000	3.43	3.53	3.41	3.44	3.34	3.45	3.34	3.44	3.35	3.34
49	50	5	20	2500	3.42	3.34	3.34	3.42	3.44	3.34	3.34	3.33	3.50	3.34
50	50	10	10	5000	3.32	3.32	3.32	3.33	3.34	3.35	3.32	3.32	3.41	3.33
51	100	2	25	2000	3.43	3.42	3.52	3.35	3.36	3.43	3.42	3.33	3.43	3.43
52	100	5	10	5000	**3.31**	**3.31**	3.32	3.33	3.33	3.40	3.33	3.39	3.33	3.32
53	125	2	20	2500	3.34	3.34	3.32	3.43	3.50	3.32	3.42	3.42	3.33	3.43
54	125	4	10	5000	3.33	3.41	3.33	3.32	3.32	3.33	**3.31**	3.32	3.32	3.33
55	250	2	10	5000	3.32	3.41	3.32	**3.31**	3.32	3.32	3.41	3.32	**3.31**	3.42

Figure 3: The values of $P(m)$ and $Q(m)$ for the clusterings obtained in Subsection 4.2. The support threshold is set to $\epsilon = 0.02$



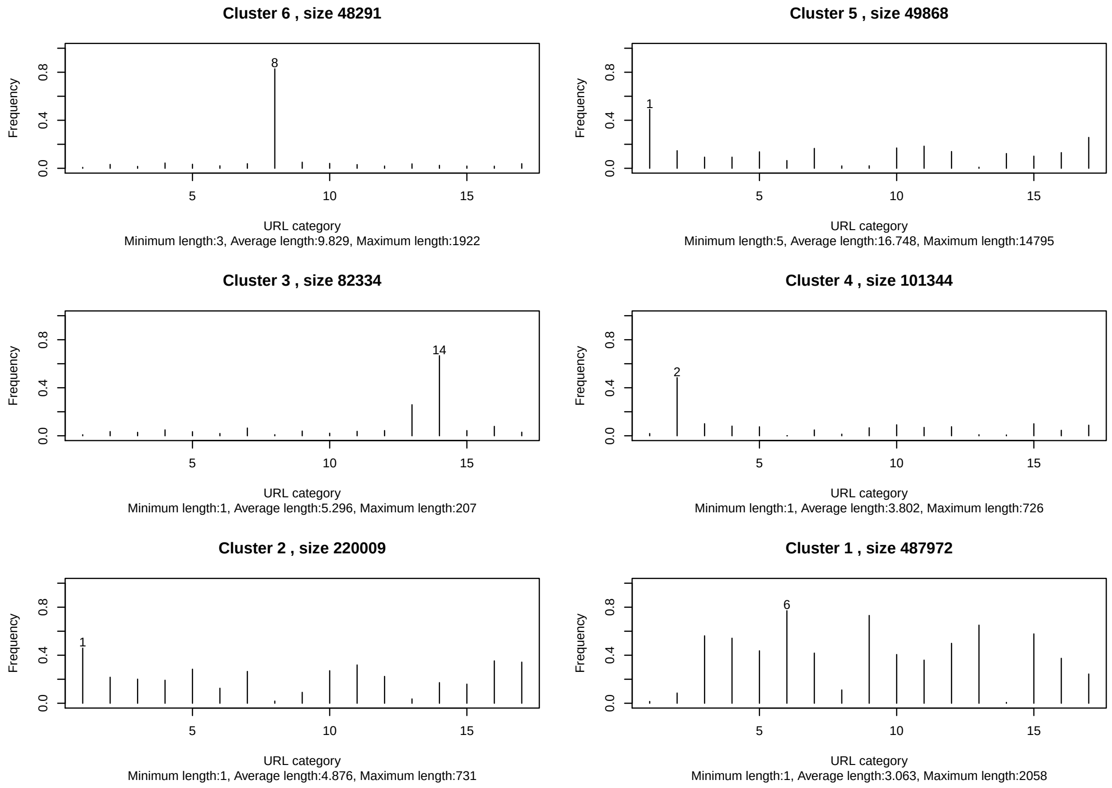
that a clustering where each open sequence has confidence 1 for some cluster and 0 for the rest would be ideal because then the l_ϵ open sequences are perfectly represented by the m clusters. In that case $P(m) = Q(m) = 0$. On the other hand, the worst case scenario is $w_{ij} = 1/m$ $(i = 1, \ldots, m\,;\, j = 1, \ldots, l_\epsilon)$ and hence $P(m) = Q(m) = 1$. Thus, given that $P(m)$ and $Q(m)$ take values in the interval $[0, 1]$, preferable clusterings could be considered the ones that correspond to small values of $P(m)$ and $Q(m)$. However, depending blindly on these criteria may return misleading results, especially for small m, because $P(m)$ and $Q(m)$ can take small values also when there exists a single, possibly big, cluster wherein all open-sequences have very high confidence. Hence, some conservatism is needed when assessing the number of clusters based on these criteria.

For the msnbc.com data, ϵ is set to 0.02 and the values of $P(m)$ and $Q(m)$ are calculated for the clusterings obtained in Subsection 4.2. The result is shown in Figure 3. The smallest value of both criteria results for $m = 3$ but the conservative choice of $m = 6$ seems to be satisfactory, because for 8 to 11 clusters the values of $P(m)$ and $Q(m)$ are almost unchanged and for 12 to 20 clusters $P(m)$ and $Q(m)$ take bigger values.

4.4 Description of the clusters

Each plot in Figure 4 shows the frequency of appearance of each URL category in the corresponding cluster relative to its frequency in the whole data set. Furthermore, each plot in Figure 5 shows the frequency of appearance of each URL category in the corresponding cluster relative to the total number of URLs appearing in that cluster. Hence, Figure 4 provides a rough between cluster analysis of URL categories (the sum of frequencies of each URL category among clusters is 1) and Figure 5 provides a within one (the sum of the frequencies of the URL categories in each cluster is 1). The cluster sizes and the minimum, average and maximum length of the sequences in each cluster are also displayed

Figure 4: The frequency of appearance of each URL category in the corresponding cluster relative to its frequency in the whole data set. The size of each cluster and the minimum, average and maximum length of the sequences within each cluster are shown on the title and the subtitle of the corresponding plot.


Cluster 6 , size 48291

Minimum length:3, Average length:9.829, Maximum length:1922

Cluster 5 , size 49868

Minimum length:5, Average length:16.748, Maximum length:14795

Cluster 3 , size 82334

Minimum length:1, Average length:5.296, Maximum length:207

Cluster 4 , size 101344

Minimum length:1, Average length:3.802, Maximum length:726

Cluster 2 , size 220009

Minimum length:1, Average length:4.876, Maximum length:731

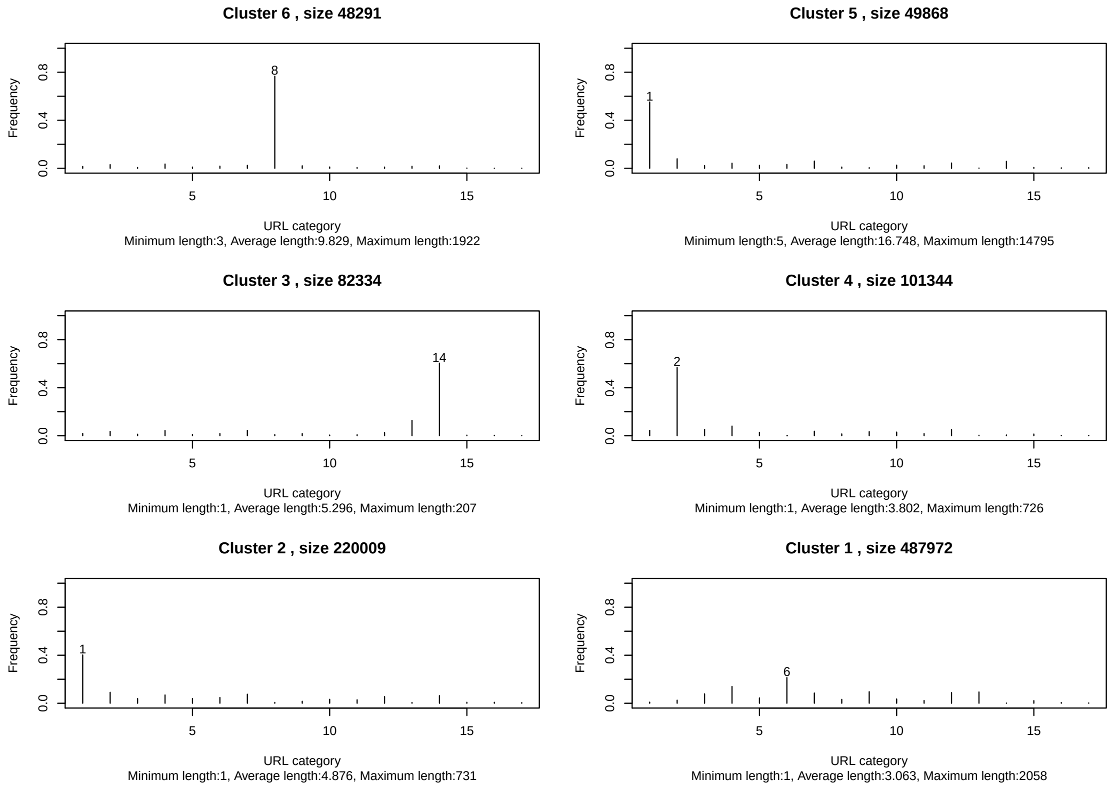
Cluster 1 , size 487972

Minimum length:1, Average length:3.063, Maximum length:2058

on the plots. Note that, Figure 4 and Figure 5 are merely a useful visualization of the by-row and by-column relative frequencies from the cross-tabulation of clusters and URL categories.

Clusters 5 and 6 are the smallest in size clusters and also have the highest average length of sequences. Specifically, cluster 5 contains the 13 of the 19 sequences with length greater than 1000 in the data set. The common feature characterizing those 13 sequences amongst all 19 is the high concentration of URLs of the “front page” category (category 1). Furthermore, about 80% of the hits to URLs of the “weather” category (category 8) is contained in cluster 6 and also, as Figure 5 reveals, about the 80% of the URLs in cluster 6 are of category “weather”. A closer look reveals that all users in cluster 5 have visited at least once a URL of category “front page” and all users in cluster 6 have visited at least once a URL of category “weather”. Cluster 3 contains about the 70% of the hits to URLs of category “bulletin board service” (category 14) and also, as Figure 5 reveals, about the 60% of the URLs in cluster 3 are of category “bulletin board service”. Similar conclusions can be drawn for cluster 4 which seems to be dominated by and also contain many of the hits in the data set to URLs of the “news” category. The two large in size clusters 1 and 2 do not have a similar transparent interpretation in terms of URL categories. By the

Figure 5: The frequency of appearance of each URL category in the corresponding cluster relative to the number of URLs appearing in that cluster. The size of each cluster and the minimum, average and maximum length of the sequences within each cluster are shown on the title and the subtitle of the corresponding plot.



plots of Figure 4, cluster 2 contains a moderate percentage of URLs in many categories and cluster 1 contains the majority of the hits in the data set to URLs of the categories 6, 9, 13 and 15 and about half the hits to URLs of the categories 3, 4, 5, 7, 10, 11, 12, 16. Furthermore, by the corresponding plots of Figure 5, the slight dominance of category 6 in cluster 1 and category 1 in cluster 2 are noted, but overall there are no significantly dominant URL categories within those clusters. The case gets clearer when considering higher-order interactions of URL categories.

The confidences of the open sequences with support threshold $\epsilon = 0.02$ are shown in Table 3, where confidences greater that 0.5 are shown in boldface type. Note that most of these open sequences are strongly associated with the observations in clusters 2 and 5. This suggests that clusters 2 and 5 accommodate similar browsing behaviours. It seems that, the separation of these two clusters is due to the fact that cluster 5 contains on average longer sequences than cluster 2. Exceptions are the open sequences $\{1, 2\}$, $\{1, 6\}$, $\{7, 4\}$, $\{2, 4\}$, $\{6, 7\}$, $\{7, 6\}$ and $\{6, 7, 6\}$. The last three of the aforementioned open sequences are strongly associated with the observations in cluster 1, with confidences 0.743, 0.741 and 0.813, respectively. Lastly, the open sequences for $\epsilon = 0.02$ do not show any strong association with clusters 3, 4 and 6. Especially, cluster 6 seems to be isolated from the

Table 3: The open sequences that have support greater than $\epsilon = 0.02$ in the msnbc.com data and their confidences for the selected clustering in the 6 clusters.

Open sequence	Support	Confidences					
		Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
$\{1,2\}$	0.072	0.050	0.437	0.018	0.230	0.253	0.012
$\{2,1\}$	0.042	0.014	**0.518**	0.006	0.039	0.412	0.011
$\{1,12\}$	0.041	0.024	**0.604**	0.013	0.049	0.302	0.008
$\{1,2,1\}$	0.038	0.000	**0.535**	0.000	0.000	0.455	0.010
$\{1,4\}$	0.037	0.044	**0.570**	0.015	0.063	0.294	0.014
$\{1,14\}$	0.037	0.007	0.498	0.207	0.016	0.262	0.010
$\{1,6\}$	0.035	0.338	0.439	0.014	0.000	0.195	0.014
$\{1,7\}$	0.034	0.077	**0.597**	0.017	0.033	0.256	0.020
$\{1,11\}$	0.032	0.033	**0.579**	0.014	0.039	0.326	0.009
$\{1,3\}$	0.031	0.036	**0.604**	0.015	0.082	0.255	0.008
$\{6,7\}$	0.030	**0.743**	0.130	0.029	0.000	0.068	0.030
$\{12,1\}$	0.026	0.009	**0.512**	0.006	0.012	0.451	0.010
$\{1,10\}$	0.026	0.083	**0.561**	0.013	0.068	0.263	0.012
$\{7,1\}$	0.025	0.024	**0.613**	0.007	0.007	0.329	0.020
$\{7,6\}$	0.025	**0.741**	0.132	0.024	0.000	0.074	0.029
$\{7,4\}$	0.024	0.316	0.324	0.044	0.106	0.135	0.075
$\{4,1\}$	0.023	0.016	**0.510**	0.006	0.013	0.442	0.013
$\{1,12,1\}$	0.023	0.000	0.485	0.000	0.000	**0.507**	0.008
$\{14,1\}$	0.022	0.004	**0.523**	0.052	0.004	0.404	0.013
$\{1,7,1\}$	0.022	0.000	**0.615**	0.000	0.000	0.369	0.016
$\{11,1\}$	0.022	0.006	**0.522**	0.004	0.005	0.454	0.009
$\{6,1\}$	0.021	0.152	**0.518**	0.010	0.000	0.304	0.016
$\{6,7,6\}$	0.021	**0.813**	0.091	0.021	0.000	0.052	0.023
$\{1,11,1\}$	0.020	0.000	**0.510**	0.000	0.000	0.482	0.008
$\{2,4\}$	0.021	0.128	0.245	0.041	0.335	0.211	0.039

other clusters for this particular choice of ϵ, because all open sequences are almost not associated with it.

5 Discussion and concluding Remarks

A supervised way for obtaining subsets of the data is developed for use in applications involving large data sets. The suggested approach consists of the construction of a data-dependent partitioning on the observation space, sub-sampling within the partitions and the application of a k-medoids clustering algorithm on each of the sub-samples. Then, the resultant centroids are returned as the subset. In this way, the probability that an observation is included in the final subset is allowed to differ between observations facilitating the detection of any small groups present in the data by standard clustering procedures.

The number of partitions, number and size of sub-samples per partition and the number of centroids per sub-sample for the application of the procedure are parameters specified by the user. In this respect, a systematic way of obtaining possible parameter values has been proposed, which results by merely fixing the total number of observations sampled during the procedure.

The suggested approach was used for efficiently obtaining a hard clustering of the msnbc.com data set in 6 clusters, and as the analysis indicated each of the clusters seems to contain observations with specific common characteristics. Importantly, that clustering could serve as a starting point for more sophisticated approaches in clustering the msnbc.com data.

Of course, despite the fact that, as is demonstrated both with simulated and real data, the suggested approach is computationally feasible for large data sets and can return good results, a formal justification for its general use is still lacking.

In the current paper, the observation space is partitioned in equi-sized partitions, each containing observations which are equivalent in terms of their distance from a centrally located observation $\hat{\mu}$. While this seemed a natural choice of partitioning scheme to the authors, other similar schemes may be constructed, for example, by allowing the number of observations per partition to vary or by constructing the partitions based on other more specific aspects of the data at hand.

Lastly, the procedure can be conveniently set up for parallel computation, for example, by treating one partition per thread for each parameter setting, or by treating one parameter setting per thread. The later parallelization strategy was used for the clustering of the msnbc.com data.

References

Bradley, P., U. Fayyad, and C. Reina (1998a). Scaling clustering algorithms to large databases. In *Proceedings of Knowledge Discovery and Data Mining conference*, pp. 9–15.

Bradley, P., U. Fayyad, and C. Reina (1998b). Scaling EM (expectation − maximization) clustering to large databases. Technical Report MSR-TR-98-35, Microsoft Research Report.

Büchner, A., M. Baumgarten, S. Anand, M. Mulvenna, and J. Hughes (1999). Navigation pattern discovery from Internet data. In *Proceedings of the Web Usage Analysis and User Profiling Workshop (WEBKDD '99), Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 25–30.

Cadez, I. V., D. Heckerman, C. Meek, P. Smyth, and S. White (2003). Model-based clustering and visualization of navigation patterns on a web site. *Data Mining and Knowledge Discovery 7*(4), 399–424.

Chiu, T., D. Fang, J. Chen, Y. Wang, and C. Jeris (2001). A robust and scalable clustering algorithm for mixed type attributes in large database environment. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 263–268.

Coleman, D. A. and D. L. Woodruff (2000). Cluster analysis for large datasets: An effective algorithm for maximizing the mixture likelihood. *Journal of Computational and Graphical Statistics 9*(4), 672–688.

Farnstrom, F., J. Lewis, and C. Elkan (2000). Scalability for clustering algorithms revisited. *SIGKDD Explorations Newsletter 2*(1), 51–57.

Fraley, C., A. Raftery, and R. Werhens (2005). Incremental model-based clustering for large datasets with small clusters. *Journal of Computational and Graphical Statistics 14*(3), 529–546.

Hay, B., G. Wets, and K. Vanhoof (2004). Mining navigation patterns using a sequence alignment method. *Knowledge and information systems 6*(2), 150–163.

Kaufman, L. and P. J. Rousseeuw (1986). Clustering large data sets. In E. S. Gelsema and L. N. Kanal (Eds.), *Pattern Recognition in Practice 2, Proceedings of an International Workshop held in Amsterdam, 1985*, pp. 425–437. Elsevier/North-Holland.

Kaufman, L. and P. J. Rousseeuw (1990). *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons.

Ng, S. K. and G. J. McLachlan (2003). On the choice of the number of blocks with the incremental em algorithm for fitting of normal mixtures. *Statistics and Computing 13*, 45–55.

Posse, C. (2001). Hierarchical model-based clustering for large datasets. *Journal of Computational and Graphical Statistics 10*(3), 464–486.

Steiner, P. and M. Hudec (2007). Classification of large data sets with mixture models via sufficient em. *Computational Statistics and Data Analysis 51*(11), 5416–5428.

Tantrum, J., A. Murua, and W. Stuetzle (2004). Hierarchical model based clustering of large datasets through fractionation and refractionation. *Information Systems 29*, 315–326.

Vijaya, P., M. Murty, and D. Subramaniam (2004). Leaders–subleaders: An efficient hierarchical clustering algorithm for large data sets. *Pattern Recognition Letters 25*, 505–513.

Wehrens, R., L. M. C. Buydens, C. Fraley, and A. E. Raftery (2004). Model-based clustering for image segmentation and large datasets via sampling. *Journal of Classification 21*(2), 231–253.