Clustering Strategies for Detecting Changes on Web Usage Data

Da Silva, Alzennyr, Lechevallier, Yves, Rossi, Fabrice
Project AxIS, INRIA-Rocquencourt, Domaine de Voluceau, B.P. 105, 78153 Le Chesnay cedex, France
E-mail: Alzennyr.Da Silva,Yves.Lechevallier, Fabrice.Rossi@inria.fr

De Carvalho, Francisco
Centro de Informatica - CIn / UFPE, Av. Prof. Luiz Freire, s/n, CDU, 50740-540 Recife, Brazil
E-mail: fatc@cin.ufpe.br

1. INTRODUCTION

The data stream model has recently attracted attention for its applicability in numerous types of data (Henzinger et al, 1998). The clustering problem is a difficult problem for the data stream domain. Previous algorithms on clustering data streams such as those discussed in (Hulten et al, 2001) assume that the clusters are to be computed over the entire data stream. However, the exploration of the stream over different time windows can provide the users with a much deeper understanding of the evolving behaviour of the clusters. In this paper, we study the clustering problem for the data stream on the Web usage framework. We propose three news strategies for a divide-and-conquer like approach which splits the data into pieces and clusters each of these pieces.

2. THE K-MEANS CLUSTERING ALGORITHM

The k-means algorithm (MacQueen, 1967) is a partitional clustering method whose aim is to furnish a partition of a set of elements E in K clusters $C_1, \ldots, C_K$ and its corresponding set of prototypes $c_1, \ldots, c_K$. The traditional k-means algorithm is executed in the following steps:

- **Initialization phase**: Let $c_1, \ldots, c_K$ be the initial prototypes, random and distinct objects of E.
- Step t:

  Allocation phase:

  An element x_i of E is assigned to the cluster C_i iff $d(x_i, c_i)$ is minimum:

  $C_i^t = \{x_i \in E | d(x_i, c_i) \leq d(x_i, c_j) \forall j \neq i (j = 1, \ldots, K)\}$

  Let $P^t = (C_1^t, \ldots, C_K^t)$ be the partition of E in K clusters at step t.

  Representation phase:

  The prototypes $c_1, \ldots, c_K$ are updated according to the current elements present in each cluster

  For $j = 1, \ldots, K$ update $c_j = \frac{1}{|C_j^t|} \sum_{x_i \in C_j^t} x_i$

  Let $(c_1^t, \ldots, c_K^t)$ be the updated prototypes at step t.

- **Stopping condition**: If $P^{t+1} = P^t$ then STOP, else GO TO Step t.

3. LOCAL INDEPENDENT CLUSTERING

In this clustering, we have one clustering process applied in each sub-period analysed separately. The final partitions are thus independent. At the end of this procces we have a partition for each time sub-period:

- **Partitioning phase**: Split the entire data set E into Z blocks according to a time constraint and generate the data partition $\{W_1, \ldots, W_Z\}$.
- For $z = 1$ to Z

Table 1. Description of the Navigation Variables

N^o	Field	Meaning
1	IDNavigation	Navigation code
2	NbRequests_OK	Number of successful requests (status = 200) into the navigation
3	NbRequests_BAD	Number of failed requests (status $\neq$ 200) into the navigation
4	PRequests_OK	Percentage of successful requests (= NbRequests_OK/ NbRequests)
5	NbRepetitions	Number of repeated requests into the navigation
6	PRepetitions	Percentage of repetitions (= NbRepetitions / NbRequests)
7	TotalDuration	Total duration of the navigation (in seconds)
8	AvDuration	Average of duration (= TotalDuration / NbRequests)
9	AvDuration_OK	Average of duration among successful requests (= TotalDuration_OK/NbRequests_OK)
10	NbRequests_SEM	Number of requests related to pages in the site's semantic structure
11	PRequests_SEM	Percentage of requests related to pages in the site's semantic structure (=NbRequests_Sem/ NbRequests)
12	TotalSize	Total size of transferred bytes in the navigation
13	AvTotalSize	Average of transferred bytes (= TotalSize / NbRequests_OK)
14	MaxDuration_OK	Duration of the longest request in the navigation

Apply the k-means clustering algorithm on the elements of W_z

4. LOCAL PREVIOUS CLUSTERING

In this clustering, we use the clustering prototypes performed in the preceding time sub-period to obtain a partition on the elements belonging to the current sub-period:

- **Partitioning phase**: Split the entire data set E into Z blocks according to a time constraint and generate the data partition $\{W_1, \ldots, W_Z\}$.
- Apply the k-means clustering algorithm on the elements of W_1
- For $z = 2$ to Z

  Let $c_1, \ldots, c_K$ be the prototypes of the resulting clustering on W_{z-1}

  Apply the `Step t` of the k-means clustering algorithm on the elements of W_z

It is important to notice that for the first sub-period, the k-means is executed until the convergence. For the other sub-periods, a partition is generated by the simple affectation of the elements to the updated prototypes $c_1, \ldots, c_K$ coming from previous partitions.

5. LOCAL DEPENDENT CLUSTERING

Here, a complete clustering is started with the prototypes of the clusters from the previous time sub-period:

- **Partitioning phase**: Split the entire data set E into Z blocks according to a time constraint and generate the data partition $\{W_1, \ldots, W_Z\}$.
- Apply the k-means clustering algorithm on the elements of W_1
- For $z = 2$ to Z

  Let $c_1, \ldots, c_K$ be the prototypes of the resulting clustering on W_{z-1}

  Apply the k-means clustering algorithm on the elements of W_z

It is important to notice that the k-means is executed until de convergence in all the sub-periods. However, the clustering process is started with the updated prototypes $c_1, \ldots, c_K$ coming from previous partitions.

6. RESULTS

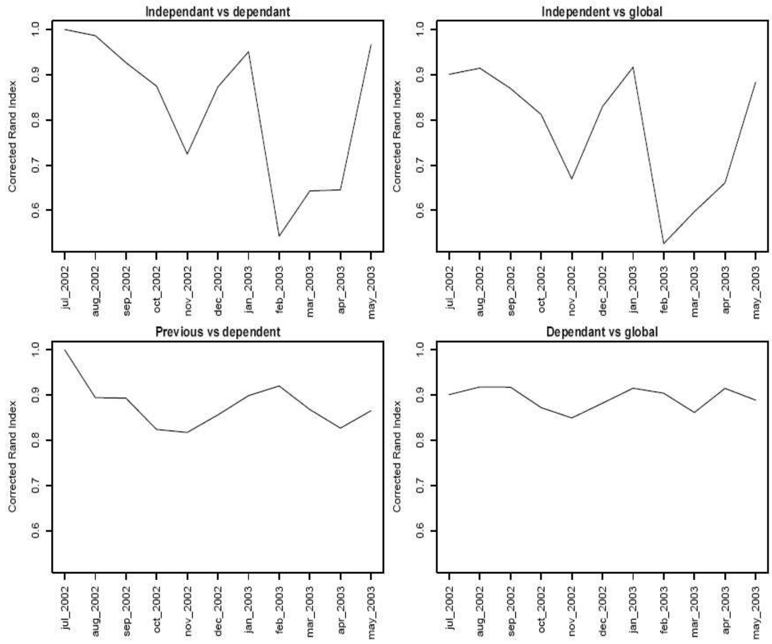

Figure 1. Corrected Rand index values computed partition-by-partition.

We analyse a Web usage data set of reference from 1^{st} July 2002 to 31^{st} May 2003[1]. A *navigation* constitutes the trajectory of a user in the site and is defined as a succession of requests coming from the same user, and there are no more than 30 minutes apart (Tanasa and Trousse, 2004).

For all the experiments, our clustering strategies are applied on navigation table (cf Table 1) split by months. We defined an a priori number of clusters equal to 10. The number of executions is equal to 100, except when the algorithm is initialized with the results obtained from a previous execution.

To analyse the results, we apply for a cluster-by-cluster analysis the F-measure [van Rijsbergen, 1979]. For a global analysis, the corrected Rand index [Hubert and Arabie, 1985] to compare two partitions. In both criteria, the value 1 indicates a perfect agreement and values near 0 correspond to cluster agreements found by chance.

The values of the corrected Rand index reveal that the results from the local independent clustering are very different from those of the global and local dependent clustering (cf. Figure 1). These differences are confirmed by the F-measure. As we obtain 10 values (one per cluster) from the F-measure for each month, we trace the corresponding boxplot to summarize these values (cf. Figure 2). From the confrontation between the local independent clustering and the global clustering, we can see that there are almost always low values, i.e., certain clusters resulting from the local independent clustering are not found by the global clustering. We can also notice that the local previous clustering does not give very different results from those obtained by the local dependent clustering.

By using a cluster-by-cluster confrontation via the F-measure between the global clustering and the local dependent and independent clustering, we refine the analysis. What appears quite clearly is that the clusters are very stable over time if we apply the local dependent clustering method. In fact, no value is lower than 0.877, which represents a very good score. On the other hand, in the case of local independent clustering, we detect clusters that are very different from those obtained by the global clustering (some values are lower than 0.5).

7. CONCLUSIONS

[1]This web site is available at the following address: `http://www.cin.ufpe.br/`

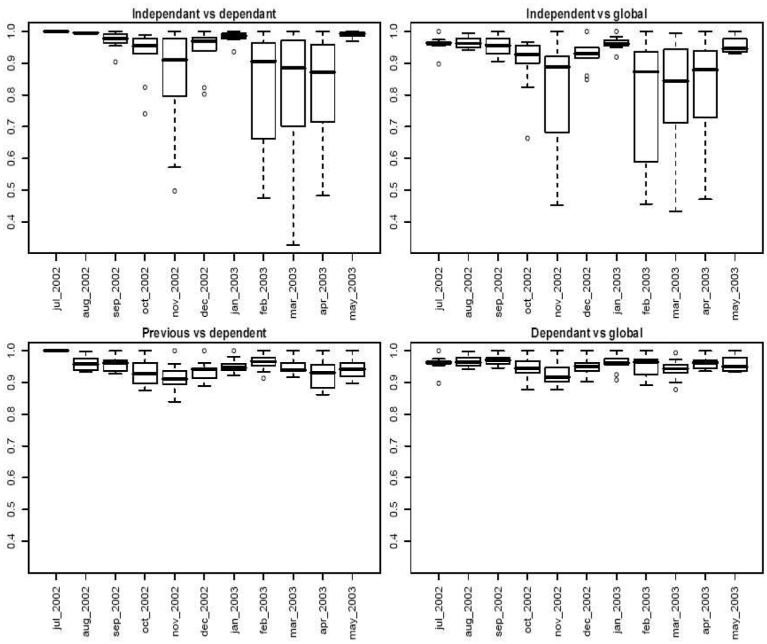

Figure 2. F-measure values computed cluster-by-cluster.

This article proposeds several strategies for improving the k-means clustering algorithm in order to detect changes in data streams. The proposed improvements are fairly easy to incorporate. All variants have been compared with the traditional form of the algorithm. A study case was performed on data stream recording Web usage traces. Through our experiments, we have shown that the analysis of dynamic data by time sub-periods offers a certain number of advantages such as making the method sensitive to cluster changes over time. Furthermore, as our approach splits the data and concentrates the analysis on fewer sub-sets, some constraints regarding hardware limitations could be overcome.

ACKNOWLEDGMENTS

The authors would like to thank the collaboration project between INRIA and FACEPE (France/Brazil) and CAPES (Brazil) for their support for this research work.

REFERENCES

M. Henzinger, P. Raghavan and S. Rajagopalan. Computing on Data Streams. Digital Equipment Corporation, TR-1998-011, 1998.

L. Hubert and P. Arabie. Comparing partitions. Journal of Classification, vol. 2, pp. 193-218, 1985.

G. Hulten, L. Spencer, and P. Domingos. Mining Time Changing Data Streams, Proc. Seventh ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 97–106, 2001.

J.B. MacQueen. Some method for classification and analysis of multivariate observations. In Proc. of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967.

D. Tanasa and B. Trousse. Advanced data preprocessing for intersites web usage mining. IEEE Intelligent Systems, vol. 19, n.2, pp. 59-65, 2004.

C. J. van Rijsbergen. Information Retrieval. Butterworths, London, second edition, 1979.