The SKM Algorithm: A K-Means Algorithm for Clustering Sequential Data

José G. Dias[1] and Maria João Cortinhal[2]

[1] Department of Quantitative Methods,
ISCTE Business School and UNIDE,
Av. das Forças Armadas, Lisboa 1649–026, Portugal
jose.dias@iscte.pt
[2] Department of Quantitative Methods,
ISCTE Business School and CIO,
Av. das Forças Armadas, Lisboa 1649–026, Portugal
maria.cortinhal@iscte.pt

Abstract. This paper introduces a new algorithm for clustering sequential data. The SKM algorithm is a K-Means-type algorithm suited for identifying groups of objects with similar trajectories and dynamics. We provide a simulation study to show the good properties of the SKM algorithm. Moreover, a real application to website users' search patterns shows its usefulness in identifying groups with heterogeneous behavior. We identify two distinct clusters with different styles of website search.

Keywords: clustering, sequential data, K-Means algorithm, KL distance.

1 Introduction

Clustering is the partition of a data set into subsets (clusters), so that the data in each subset share similar characteristics. The application of clustering algorithms has been extensive. For example, in Marketing, market segmentation means the identification of groups of customers with similar behavior given a large database of customer data containing their properties and past buying records; in Biology, the taxonomical classification of plants and animals given their features; or, in earthquake studies in which clustering observed earthquake epicenters allows the identification of dangerous zones. In this research we focus on the clustering of sequential data.

Let us have a data set of n objects to be clustered. An object will be denoted by i ($i = 1, \ldots, n$). Each object is characterized by a sequence of states $\mathbf{x}_i$. Let $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ denote a sample of size n. Let x_{it} denote the state of the object i at position t. We will assume discrete time from 0 to T_i ($t = 0, 1, \ldots, T_i$). Note that the length of the sequence may differ among objects. Thus, the vector $\mathbf{x}_i$ denotes the consecutive states x_{it}, with $t = 0, \ldots, T_i$. The sequence $\mathbf{x}_i = (x_{i0}, x_{i1}, ..., x_{iT_i-1}, x_{iT_i})$ can be extremely difficult to characterize, due to its possibly huge dimension ($T_i + 1$). A common procedure to simplify the sequence

H. Geffner et al. (Eds.): IBERAMIA 2008, LNAI 5290, pp. 173–182, 2008.

is by assuming the Markov property. It states that the occurrence of event x_t depends only upon the previous state x_{t-1}; that is, conditional on x_{t-1}, x_t is independent of the states at the other time points. From the Markov property, it follows that the probability of a sequence $\mathbf{x}_i$ is

$$p(\mathbf{x}_i) = p(x_{i0}) \prod_{t=1}^{T_i} p(x_{it}|x_{i,t-1}) \tag{1}$$

where $p(x_{i0})$ is the initial distribution and $p(x_{it}|x_{i,t-1})$ is the probability that object i is in state x_{it} at t, given that it is in state $x_{i,t-1}$ at time $t-1$ (For an introduction to Markov chains, see [9]). A first-order Markov chain is specified by its transition probabilities and initial distribution. Hereafter, we denote the initial and the transition probabilities as $\lambda_j = P(x_{i0} = j)$ and $a_{jk} = P(x_t = k|x_{t-1} = j)$, respectively. The parameters of the Markov chain can be estimated by

$$\hat{\lambda}_j = \sum_{i=1}^{n} I(x_{i0} = j) \tag{2}$$

$$\hat{a}_{jk} = \frac{\sum_{i=1}^{n} n_{ijk}}{\sum_{r=1}^{K} \sum_{i=1}^{n} n_{ijr}}, \tag{3}$$

where K denotes the number of states, $I(x_{i0} = j)$ is the indicator function at time point 0, and n_{ijk} is the number of transitions from state j to state k for object i.

This paper introduces a K-Means-type algorithm that allows the clustering of this type of data. Section 2 describes a K-means algorithm for sequential data. Section 3 analyzes the performance of the algorithm. Section 4 illustrates the application of the algorithm in clustering web users based on their longitudinal pattern of search. The paper concludes with a summary of main findings, implications, and suggestions for further research.

2 The SKM Algorithm

2.1 The K-Means Algorithm

The K-means algorithm [7] is one of the simplest unsupervised learning algorithms that solves the clustering problem. That is, this procedure defines a simple and fast way to determine a partition of a data set into a certain number of clusters (assume S clusters) fixed a priori so that the within group sum of squares is minimized. The K-means algorithm consists of the following steps:

1. Set the number of clusters, S;
2. Generate randomly initial cluster centroids;
3. Assign each object i to the cluster s that has the closest centroid;
4. Recalculate the positions of the centroids;
5. If the positions of the centroids did not change the algorithm ends, otherwise go to Step 2.

However, it has been shown that the conventional K-means algorithm is inappropriate for the discovery of similar patterns in sequential data (e.g., for web usage patterns, see [13]). For web mining purposes, [8] proposed clustering web users using a K-means algorithm based on the KL-divergence which measures the "distance" between individual data distributions. A similar approach is adopted by [16], who looked at the number of times a given user visited a given webpage. [12] suggested using self-organizing maps (SOMs) of user navigation patterns. On the other hand, [14] suggests the clustering of objects at different time points and then, the analyzes of the evolution of the clusters found. However, none of these approaches accounts for the sequential structure of data at the individual level. This means that consecutive states in a sequence are, in fact, treated as independent observations conditional on cluster membership, an assumption that is rather unrealistic. The proposed SKM algorithm circumvents this problem.

2.2 The SKM Algorithm

The SKM (Sequential K-Means) algorithm is a K-Means-based algorithm that uses the Kullback-Leibler distance [4] to cluster sequential data. Let us define some notation first. Let $z_{is} = 1$ if object i belongs to cluster s, and 0 otherwise. Then, transition probabilities within each cluster are the *centroids* and are defined by:

$$\hat{a}_{sjk} = \frac{\sum_{i=1}^{n} z_{is} n_{ijk}}{\sum_{r=1}^{K} \sum_{i=1}^{n} z_{is} n_{ijr}}. \tag{4}$$

Let d_{is} be a measure of distance of object i to the prototype of cluster s. Because we want to measure the distance or divergence of each object i to each centroid s, and provided that the centroid of each cluster is defined by the set of transition probabilities $\hat{a}_{sjk}$, the Kullback-Leibler (KL) distance is the appropriate distance for this type of centroids (probabilities or proportions) and yields[1]

$$d_{is} = \sum_{j=1}^{K} \sum_{k=1}^{K} \hat{a}_{sjk} \ln \left(\frac{\hat{a}_{sjk}}{p_{ijk}} \right), \tag{5}$$

[1] In our clustering procedure we do not compute the distance between objects i and j. In that case, it would be desirable to have a symmetric distance, i.e., the distance between objects i and j should be the same as the one between objects j and i ($d_{ij} = d_{ji}$). Because the KL distance between i and j is different from the KL distance between j and i, we could have obtained a symmetric KL distance by computing the mean of these two KL distances as in [8]. In our clustering procedure symmetry is not needed or even desirable. By computing the distance between each object i and the centroid of each cluster s, the KL distance uses as weight function $\hat{a}_{sjk}$ that is a more stable distribution than p_{ijk} as the former is based on all objects in cluster s (equation 4) and the latter is just based on observed transitions for object i.

where p_{ijk}, the transition probability from state j to state k for object i, is defined by

$$p_{ijk} = \frac{n_{ijk}}{\sum_{k=1}^{K} n_{ijk}}. \tag{6}$$

The SKM is an iterative algorithm that starts from a randomly generated solution. Given the number of clusters, a randomly generated allocation of objects is generated, and at each iteration step objects are reallocated according to the Kullback-Leibler distance; that is, each object is assigned into the closest cluster's centroid. This process is repeated until some termination criterion is met. Moreover, two termination criteria are defined: the maximum number of iterations - $MaxIter$ - and the absence of change in the allocation of objects between two consecutive iterations. Because these K-Means type algorithms suffer from local optima, it is desirable to repeat the iterative process with different random starting values. Out of a set of R runs, we select the solution that maximizes the classification log-likelihood:

$$\ell = \sum_{i=1}^{n} \sum_{s=1}^{S} z_{is} \left[\log \pi_s + \sum_{j=1}^{K} I(x_{i0} = j) \log \lambda_{sj} + \sum_{j=1}^{K} \sum_{k=1}^{K} n_{ijk} \log a_{sjk} \right], \tag{7}$$

where $\hat{\lambda}_{sj}$ – the initial probabilities within each cluster – and $\hat{\pi}_s$ – the proportion of objects in cluster s – are defined by:

$$\hat{\lambda}_{sj} = \frac{\sum_{i=1}^{n} z_{is} I(x_{i0} = j)}{\sum_{i=1}^{n} z_{is}}, \tag{8}$$

$$\hat{\pi}_s = \frac{1}{n} \sum_{i=1}^{n} z_{is}. \tag{9}$$

The pseudo code for the SKM algorithm is:

SKM(S, x_{i0}, n_{ijk}, $MaxIter$)
$Iter \leftarrow 1$
Randomly generate a matrix Z such that:
$z_{is} \in \{0, 1\}$, $\sum_{s=1}^{S} z_{is} = 1$, and $\sum_{i=1}^{n} z_{is} > 0$
REPEAT
Update the previous allocation of objects:$Z\_Old \leftarrow Z$
Update the centroids ($\hat{a}_{sjk}$)
Compute d_{is} for each object i and cluster s
Determine $s_i = \arg\min_{s \in \{1,\ldots,S\}} d_{is}$, for each object i
Update $Z : z_{i,s_i} \leftarrow 1$ and $z_{is} \leftarrow 0$, for all $s \neq s_i$
$Iter \leftarrow Iter + 1$
UNTIL ($Iter = MaxIter$ or $Z\_Old = Z$)
Compute $\hat{\pi}_s$, $\hat{\lambda}_{sj}$, and the classification log-likelihood (ℓ)
RETURN ($\ell, \hat{a}_{sjk}$, $\hat{\pi}_s$, $\hat{\lambda}_{sj}$)

where Z and Z_Old represent matrices with allocations z_{is}, and $Iter$ counts the number of iterations.

The SKM algorithm was implemented in MATLAB 7.0 [6].

3 Simulation Study

This section analyzes the performance of the proposed algorithm using synthetic data sets. We set:

1. Sample size: $n = 1200$;
2. Sequence length: $T = 50$;
3. Number of clusters: $S = 3$;
4. Number of states: $K = 3$;
5. Number of runs: $R = 50$.

The clusters' sizes (π_s) and initial probabilities (λ_{js}) are the same across clusters and states, respectively. They are defined as $\pi_s = S^{-1}$ and $\lambda_{js} = K^{-1}$. Thus, all cluster size proportions are 0.333 (π_s) and the probability of starting in a given state within each cluster is 0.333 (λ_{js}). In order to obtain different levels of separation of these three clusters, the transition probabilities are defined as

$$a_{sjk} = \begin{cases} \alpha_s & , j = k \\ (1-\alpha_s)/(K-1) & , j \neq k, \end{cases} \tag{10}$$

where

$$\alpha_s = \begin{cases} 0.5 - \delta & , s = 1 \\ 0.5 & , s = 2 \\ 0.5 + \delta & , s = 3. \end{cases} \tag{11}$$

The δ parameter is set to 0.4, 0.3, 0.2 and 0.1, which yields four data sets – Study1_40, Study1_30, Study1_20 and Study1_10 – with increasing cluster overlapping. For example, for Study1_40 the diagonal probabilities are 0.1, 0.5, and 0.9 for cluster 1, 2, and 3, respectively.

For each data set, the SKM algorithm was run with $R = 50$ different randomly generated initial solutions. The effect of the starting solutions on the results was analyzed by a percentage deviation based on the classification log-likelihood (ℓ). Let $Best_\ell$ be the maximum classification log-likelihood out of 50 and ℓ_r the classification log-likelihood obtained at run r.[2] The percentage deviation is defined by:

$$Dev_r = 100 \times \frac{\ell_r - Best_\ell}{Best_\ell} \tag{12}$$

Table 1 depicts the maximum (max), the mean, the minimum (min), and the standard deviation (stdev) values of the percentage deviation for each data set.

[2] To avoid the comparison of the best classification log-likelihood with itself only 49 runs are applied.

Table 1. Minimum, mean, maximum, and standard deviation values of percentage deviations for Study1 data sets

Data Set	min	mean	max	stdev
Study1_10	0.005	0.027	0.051	0.010
Study1_20	0.000	0.006	0.009	0.003
Study1_30	0.000	0.002	0.003	0.001
Study1_40	0.001	0.002	0.002	0.000

Exception made for minimum values, the percentage deviation increases as the level of separation decreases. However, the average and standard deviation values lead us to conclude that SKM algorithm is not significantly dependent of the initial solution.

Table 2 depicts the best result out of 50 runs of the SKM algorithm for each data set. Globally one concludes that the results are very close to the true values and the differences are due to sampling error in the simulation study. Indeed, in all four data sets the SKM algorithm is able to retrieve their cluster structure. As expected the Study1_10 is the most difficult one because the clusters are not very well separated as for example in Study1_40, where groups show very different dynamic behavior. For example, cluster 1 tends to move in a very fast way between states, whereas cluster 3 tends to stay in the same state. Because in Study1_10 clusters have more similar patterns of change the results are slightly more difficult to retrieve comparing to the remaining three data sets.

To compare the relative performance of the SKM algorithm for these four data sets we compute the Kullback-Leibler divergence between the true values and the SKM results (Table 3). We conclude that Study1_10 has the most difficult structure to be retrieved (0.050). Interestingly the cluster structure of Study1_40 is more difficult than Study1_30 structure. This has to do with the existence of rare transitions between states in cluster 3 of Study1_40 data set (the true value of the probability of transition between different states is 0.05) that introduces some instability in the computation of the centroids of the SKM algorithm. Setting Study_10 as the standard the KL proportion (Table 3) gives each distance as a proportion of the maximum distance (Study1_10). For instance, we infer that Study1_10 is more difficult comparing with Study1_20 than Study1_20 comparing with Study1_30.

4 Application

The analysis of the sequence of web pages requested by each web user visiting a web site allows a better understanding and prediction of users' behavior and further improvements of the design of the web site. For example, web mining of online stores may yield information on the effectiveness of marketing and web merchandizing efforts, such as how the consumers start the search, which products they see, and which products they buy [10,5]. Substantial effort has

Table 2. Best SKM results for Study1 data sets

Data set	Cluster								
	$s=1$			$s=2$			$s=3$		
	$k=1$	$k=2$	$k=3$	$k=1$	$k=2$	$k=3$	$k=1$	$k=2$	$k=3$
Study1_40									
π_s		0.338			0.316			0.347	
λ_{s1}		0.363			0.309			0.361	
λ_{s2}		0.338			0.330			0.341	
λ_{s3}		0.299			0.362			0.298	
a_{s1k}	**0.103**	0.456	0.442	**0.519**	0.242	0.239	**0.874**	0.063	0.063
a_{s2k}	0.458	**0.105**	0.437	0.258	**0.493**	0.249	0.058	**0.884**	0.058
a_{s3k}	0.439	0.451	**0.111**	0.246	0.256	**0.499**	0.056	0.059	**0.885**
Study1_30									
π_s		0.353			0.308			0.339	
λ_{s1}		0.331			0.314			0.344	
λ_{s2}		0.333			0.368			0.319	
λ_{s3}		0.336			0.319			0.337	
a_{s1k}	**0.213**	0.397	0.390	**0.505**	0.249	0.246	**0.787**	0.105	0.108
a_{s2k}	0.395	**0.216**	0.389	0.250	**0.503**	0.247	0.100	**0.796**	0.104
a_{s3k}	0.389	0.398	**0.213**	0.246	0.243	**0.511**	0.109	0.112	**0.779**
Study1_20									
π_s		0.359			0.293			0.348	
λ_{s1}		0.362			0.339			0.345	
λ_{s2}		0.350			0.345			0.325	
λ_{s3}		0.288			0.316			0.330	
a_{s1k}	**0.301**	0.342	0.357	**0.502**	0.249	0.249	**0.693**	0.158	0.150
a_{s2k}	0.338	**0.311**	0.351	0.246	**0.496**	0.258	0.156	**0.694**	0.151
a_{s3k}	0.347	0.350	**0.303**	0.246	0.250	**0.505**	0.151	0.149	**0.693**
Study1_10									
π_s		0.353			0.269			0.378	
λ_{s1}		0.312			0.344			0.330	
λ_{s2}		0.305			0.347			0.317	
λ_{s3}		0.383			0.310			0.352	
a_{s1k}	**0.395**	0.320	0.284	**0.486**	0.239	0.275	**0.606**	0.195	0.199
a_{s2k}	0.292	**0.417**	0.291	0.241	**0.450**	0.309	0.206	**0.613**	0.181
a_{s3k}	0.315	0.322	**0.364**	0.230	0.200	**0.569**	0.202	0.237	**0.561**

Table 3. KL divergence between true values and best SKM results and KL proportion using Study_10 as the standard one

Data Set	KL divergence	KL proportion
Study1_40	0.0170	0.23
Study1_30	0.0088	0.18
Study1_20	0.0195	0.39
Study1_10	0.0500	1.00

Table 4. Cluster sizes and initial proportions

Parameters	Cluster 1	Cluster 2
π_s	0.5890	0.4110
λ_{s1}	0.2350	0.5319
λ_{s2}	0.0981	0.0122
λ_{s3}	0.0625	0.0117
λ_{s4}	0.0278	0.0783
λ_{s5}	0.0105	0.0004
λ_{s6}	0.1674	0.0365
λ_{s7}	0.0071	0.0034
λ_{s8}	0.0251	0.1800
λ_{s9}	0.0547	0.1041
λ_{s10}	0.0177	0.0015
λ_{s11}	0.0170	0.0049
λ_{s12}	0.0750	0.0127
λ_{s13}	0.1178	0.0039
λ_{s14}	0.0747	0.0073
λ_{s15}	0.0068	0.0112
λ_{s16}	0.0017	0.0000
λ_{s17}	0.0001	0.0000

been put on mining web access logs in an attempt to discovering groups of users exhibiting similar browsing patterns [15].

We apply the SKM algorithm to the well-known `msnbc.com anonymous web` data in `kdd.ics.uci.edu/databases/msnbc/msnbc.data.html`. This dataset describes the page visits on `msnbc.com` on September 28, 1999. Each sequence in the data set corresponds to page views of a user during a twenty-four hour period. The original number of users is 989818 and each event in the sequence is classified into the following categories (states): 1) `frontpage`, 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) `bbs` (bulletin board service), 15) `travel`, 16) `msn-news`, and 17) `msn-sports`. This dataset has been used by others [1,3]. In our study, we used a sample of 5000 sequences with at least one transition.

The SKM algorithm allows the clustering of the web users into S clusters, each of which contains individuals with similar browsing pattern. For K-Means-like algorithms the number of clusters is set a priori. In this application, we set the number of clusters based on the Bayesian Information Criterion (BIC) of Schwarz [11]. We set $S = 2$ (see, e.g., [1,3])[3].

Table 4 and Figure 1 provide a summary of the SKM algorithm's best results. The size of each cluster ($\hat{\pi}_s$) is provided in Table 4. Cluster 1, the largest (58.9%),

[3] One difficulty in applying K-means like algorithms is that the number of clusters has to be set in advance. Whenever the number of groups is not known *a priori* we suggest a BIC-like decision rule [11], i.e., we select the number of clusters S that minimizes $C_S = -2\ell + d \times log(n)$, where ℓ and d are the classification log-likelihood and the number of free parameters, respectively.

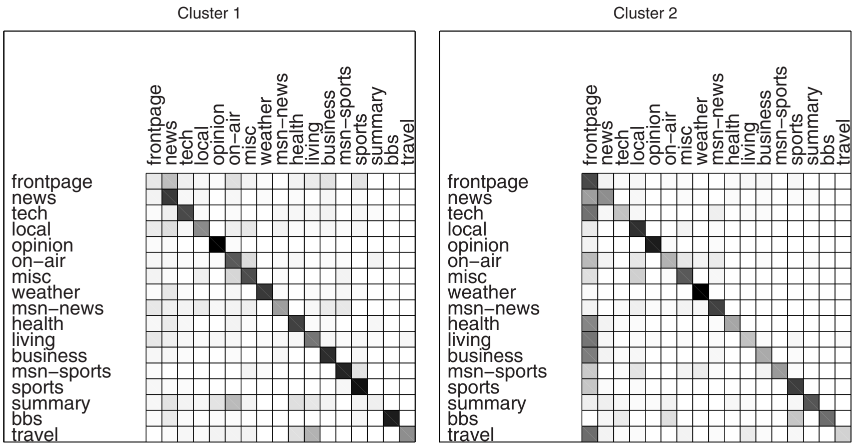

Fig. 1. Transitions matrix within each cluster. For the minimum and maximum values of the transitions probabilities (0 and 1), we use white and black, respectively. Values in between with a gray color which is obtained by a linear grading of colors between white and black. Note that the origin states are in the rows and the destination states in the column, which means that the row totals are equal to 1.

is still very heterogeneous with web users starting their browsing mainly from `frontpage` (23.5% of the web users in this cluster start their sequence in this state), `on-air` (16.7%), and `summary` (11.8%). Moreover, this cluster has a very stable pattern of browsing (Figure 1) almost absorbing for most of the states. Cluster 2 (41.1% of the sample) is rather stable. Indeed, most of them start their search from `frontpage` (53.2%) or `weather` (18.0%) states and tend to stay in these states. On the other hand, even users starting from other states tend to move to `frontpage` (Figure 1).

5 Conclusion

In this paper we provided a new K-Means algorithm for clustering sequential data. It is based on the Kullback-Leibler distance as an alternative to the standard Euclidean distance. We illustrate its performance based on synthetic data sets. The application of the algorithm in a web mining problem allows the identification of the clustering structure of web users using a well-known data set. Future research could extend our findings using synthetic data sets in such a way that can provide evidence of the performance of the SKM algorithm. In particular, the comparison between the SKM algorithm with the model-based clustering approach as in [1,2] would allow a better understanding of the statistical properties of the SKM algorithm. Another topic for further investigation is the definition of rules in the selection of the number of clusters to set *a priori*.

References

1. Cadez, I., Heckerman, D., Meek, C., Smyth, P., White, S.: Model-based clustering and visualization of navigation patterns on a web site. Data Mining and Knowledge Discovery 7(4), 399–424 (2003)
2. Dias, J.G., Willekens, F.: Model-based clustering of life histories with an application to contraceptive use dynamics. Mathematical Population Studies 12(3), 135–157 (2005)
3. Dias, J.G., Vermunt, J.K.: Latent class modeling of website users' search patterns: Implications for online market segmentation. Journal of Retailing and Consumer Services 14(4), 359–368 (2007)
4. Kullback, S., Leibler, R.A.: On information and sufficiency. The Annals of Mathematical Statistics 22(1), 79–86 (1951)
5. Lee, J., Podlaseck, M., Schonberg, E., Hoch, R.: Visualization and analysis of clickstream data of online stores for understanding Web merchandising. Data Mining and Knowledge Discovery 5(1-2), 59–84 (2001)
6. MathWorks.: MATLAB 7.0. Natick, MA: The MathWorks, Inc (2004)
7. MacQueen, J.B.: Some Methods for classification and Analysis of Multivariate Observations. In: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)
8. Petridou, S.G., Koutsonikola, V.A., Vakali, A.I., Papadimitriou, G.I.: A divergence-oriented approach for web users clustering. In: Gavrilova, M.L., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3981, pp. 1229–1238. Springer, Heidelberg (2006)
9. Ross, S.M.: Introduction to Probability Models, 7th edn. Harcourt/Academic Press, San Diego (2000)
10. Shahabi, C., Zarkesh, A.M., Adibi, J., Shah, V.: Knowledge discovery from users Web-page navigation. In: Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE 1997). High Performance Database Management for Large-Scale Applications, pp. 20–29. IEEE Computer Society, Los Alamitos (1997)
11. Schwarz, G.: Estimating the dimension of a model. Annals of Statistics 6, 461–464 (1978)
12. Smith, K.A., Ng, A.: Web page clustering using a self-organizing map of user navigation patterns. Decision Support Systems 35(2), 245–256 (2003)
13. Spiliopoulou, M., Pohle, C.: Data mining for measuring and improving the success of Web sites. Data Mining and Knowledge Discovery 5(1-2), 85–114 (2001)
14. Spiliopoulou, M., Ntoutsi, I., Theodoridis, Y., Schult, R.: MONIC: modeling and monitoring cluster transitions. In: KDD 2006, pp. 706–711 (2006)
15. Vakali, A., Pokorny, J., Dalamagas, T.: An overview of web data clustering practices. In: Lindner, W., Mesiti, M., Türker, C., Tzitzikas, Y., Vakali, A.I. (eds.) EDBT 2004. LNCS, vol. 3268, pp. 597–606. Springer, Heidelberg (2004)
16. Yang, Y.H., Padmanabhan, B.: GHIC: A hierarchical pattern-based clustering algorithm for grouping Web transactions. IEEE Transactions on Knowledge and Data Engineering 17(9), 1300–1304 (2005)