A Web User Profiling Approach

Younes Hafri[1,2], Chabane Djeraba[2], Peter Stanchev[3], and Bruno Bachimont[1]

[1] Institut National de l'Audiovisuel, 4 avenue de l'Europe
94366 Bry-sur-Marne Cedex, France
{yhafri,bbachimont}@ina.fr
[2] Institut de Recherche en Informatique de Nantes, 2 rue de la Houssiniere
43322 Nantes Cedex, France
djeraba@irin.univ-nantes.fr
[3] Kettering University, USA
pstanchev@kettering.edu
3, Kettering University, Flint, MI 48504, USA

Abstract. People display regularities in almost everything they do. This paper proposes characteristics of an idealized algorithm that would allow an automatic extraction of web user profil based on user navigation paths. We describe a simple predictive approach with these characteristics and show its predictive accuracy on a large dataset from KDD-Cup web logs (a commercial web site), while using fewer computational and memory resources. To achieve this objective, our approach is articulated around three notions: (1) Applying probabilistic exploration using Markov models. (2) Avoiding the problem of Markov model high-dimensionality and sparsity by clustering web documents, based on their content, before applying the Markov analysis. (3) Clustering Markov models, and extraction of their gravity centers. On the basis of these three notions, the approach makes possible the prediction of future states to be visited in k steps and navigation sessions monitoring, based on both content and traversed paths.

1 Introduction

On the web today, sites are still unable to market each individual user in a way, which truly matches their interests and needs. Sites make broad generalizations based on huge volumes of sales data that don't accurate an individual person. Amazon.com tracks relationships in the buying trends of its users, and makes recommendations based upon that data. While viewing a DVD by Dupont Durand, the site recommends three other Dupont Durant DVDs that are often bought by people who buy first DVD. The agreement calls for the participating web documents to track their users so the advertisements can be precisely aimed at the most likely prospects for web documents. For example, a user who looks up tourist document about Paris might be fed ads for web documents of hotels in Paris. What would be far more useful would be if web sites were able to understand specific user's interests and provide them with the content that was relevant to them. Instead of requiring users to provide their interests, a website could learn the type of content that interests the users and automatically

X. Zhou, Y. Zhang, and M.E. Orlowska (Eds.): APWeb 2003, LNCS 2642, pp. 227–238, 2003.

place that information in more prominent locations on the page. Taking a film example, if a web site knew where you lived, it could provide you with information when an actor is touring in the user area. From an advertising standpoint, this technology could be used for better targeting of advertisements to make them more relevant to each user. It is evident that there are strong dependencies between web exploration and different domains and usages. The fundamental requirements that ensure the usability of such systems are: (1) obtaining compressed and exhaustive web representations, and (2) providing different exploration strategies adapted to user requirements. The scope of the paper deals with the second requirement by investigating an exploration based on historical exploration of web and user profiles. This new form of exploration induces the answer to difficult problems. An exploration system should maintain over time the inference schema for user profiling and user-adapted web retrieval. There are two reasons for this. Firstly, the information itself may change. Secondly, the user group is largely unknown from the start, and may change during the usage of exploration processes. To address these problems, the approach, presented in this paper, models profile structures extracted and represented automatically in Markov models in order to consider the dynamic aspect of user behaviors. The main technical contribution of the paper is the notion of probabilistic prediction, path analysis using Markov models, clustering Markov models and dealing with the high dimension matrix of Markov models in clustering algorithm. The paper provides a solution, which efficiently accomplishes such profiling. This solution should enhance the day-to-day web exploration in terms of information filtering and searching.

The paper contains the following sections. Section 2 situates our contribution among state of art approaches. Section 3 describes user-profiling based web exploration. Section 4 highlights the general framework of the system and presented some implementation results. Finally, section 5 concludes the paper.

2 Related Works and Contribution

The analysis of sequential data is without doubts an interesting application area since many real processes show a dynamic behavior. Several examples can be reported, one for all is the analysis of DNA strings for classification of genes, protein family modeling, and sequence alignment. In this paper, the problem of unsupervised classification of temporal data is tackled by using a technique based on Markov Models. MMs can be viewed as stochastic generalizations of finite-state automata, when both transitions between states and generation of output symbols are governed by probability distributions [1]. The basic theory of MMs was developed in the late 1960s, but only in the last decade it has been extensively applied in a large number of problems, as speech recognition [6], handwritten character recognition [2], DNA and protein modeling [3], gesture recognition [4], behavior analysis and synthesis [5], and, more in general, to computer vision problems. Related to sequence clustering, MMs has not been extensively used, and a few papers are present in the literature. Early works were proposed in [6,7], all related to speech recognition. The first interesting approach

not directly linked to speech issues was presented by Smyth [8], in which clustering was faced by devising a "distance" measure between sequences using HMMs. Assuming each model structure known, the algorithm trains an HMM for each sequence so that the log-likelihood (LL) of each model, given each sequence, can be computed. This information is used to build a LL distance matrix to be used to cluster the sequences in K groups, using a hierarchical algorithm. Subsequent work, by Li and Biswas [10,11], address the clustering problem focusing on the model selection issue, i.e. the search of the HMM topology best representing data, and the clustering structure issue, i.e. finding the most likely number of clusters. In [10], the former issue is addressed using standard approach, like Bayesian Information Criterion [12], and extending to the continuous case the Bayesian Model Merging approach [13]. Regarding the latter issue, the sequence-to-HMM likelihood measure is used to enforce the within-group similarity criterion. The optimal number of clusters is then determined maximizing the Partition Mutual Information (PMI), which is a measure of the inter-cluster distances. In [11], the same problems are addressed in terms of Bayesian model selection, using the Bayesian Information Criterion (BIC) [12], and the Cheesman-Stutz (CS) approximation [14]. Although not well justified, much heuristics are introduced to alleviate the computational burden, making the problem tractable, despite remaining of elevate complexity. Finally, a model-based clustering method is also proposed in [15], where HMMs are used as cluster prototypes, and Rival Penalized Competitive Learning (RPCL), with state merging is then adopted to find the most likely HMMs modeling data. These approaches are interesting from the theoretical point of view, but they are not tested on real data. Moreover, some of them are very computationally expensive.

Each visitor of a web site leaves a trace in a log file of the pages that he or she visited. Analysis of these click patterns can provide the maintainer of the site with information on how to streamline the site or how to personalize it with respect to a particular visitor type. However, due to the massive amount of data that is generated on large and frequently visited web sites, clickstream analysis is hard to perform 'by hand'. Several attempts have been made to learn the click behaviour of a web surfer, most notably by probabilistic clustering of individuals with mixtures of Markov chains [16,9,17]. Here, the availability of a prior categorization of web pages was assumed; clickstreams are modelled by a transition matrix between page categories. However, manual categorization can be cumbersome for large web sites. Moreover, a crisp assignment of each page to one particular category may not always be feasible. In the following section we introduce the problem and then describe the model for clustering of web surfers. We give the update equations for our algorithm and describe how to incorporate prior knowledge and additional user information. Then we apply the method to logs from a large commercial web site KDDCup (http://www.ecn.purdue.edu/KDDCUP/), discuss both method and results and draw conclusions.

3 User Profiling and Web Exploration

3.1 Mathematical Modeling

Given the main problem "profiling of web exploration", the next step is the selection of an appropriate mathematical model. Numerous time-series prediction problems, such as in [18], supported successfully probabilistic models. In particular, Markov models and Hidden Markov Models have been enormously successful in sequence generation. In this paper, we present the utility of applying such techniques for prediction of web explorations.

A Markov model has many interesting properties. Any real world implementation may statistically estimate it easily. Since the Markov model is also generative, its implementation may derive automatically the exploration predictions. The Markov model can also be adapted on the fly with additional user exploration features . When used in conjunction with a web server, this later may use the model to predict the probability of seeing a scene in the future given a history of accessed scenes. The Markov state-transition matrix represents, basically, "user profile" of the web scene space. In addition, the generation of predicted sequences of states necessitates vector decomposition techniques. The figure 1 shows the graph representing a simple Markov chain of five nodes and their corresponding transitions probabilities. The analogy between the transition probability matrix and the graph is obvious. Each state of the matrix corresponds to a node in the graph and similarly each probability transition in the matrix corresponds to and edge in the graph. A set of three elements defines a discrete Markov model: $< \alpha, \beta, \lambda >$ where α corresponds to the state space. β is a matrix representing transition probabilities from one state to another. λ is the initial probability distribution of the states in α. Each transition contains the

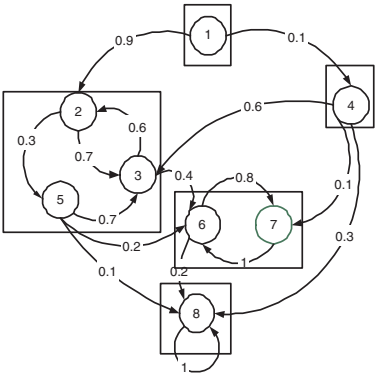


Fig. 1. Transition graph

identification of the session, the source scene, the target scene, and the dates of accesses.

The fundamental property of Markov model is the dependencies of the previous states. If the vector $\alpha(t)$ denotes the probability vector for all the states at time 't', then:

$$\alpha(t) = \alpha(t-1) \times \beta \tag{1}$$

If there are N states in the Markov model, then the matrix of transition probabilities β is of size N x N. Scene sequence modeling supports the Markov model. In this formulation, a Markov state corresponds to a scene presentation, after a query or a browsing. Many methods estimate the matrix β. Without loss of generality, the maximum likelihood principle is applied in this paper to estimate β and λ. The estimation of each element of the matrix $\beta[v, v']$ respect the following formula:

$$\beta[v, v'] = \phi(v, v')/\phi(v) \tag{2}$$

where $\phi(v, v')$ is the count of the number of times v' follows v in the training data. We utilize the transition matrix to estimate short-term exploration predictions. An element of the matrix state, say $\beta[v, v']$ can be interpreted as the probability of transitioning from state v to v' in one step. The Markovian assumption varies in different ways. In our problem of exploration prediction, we have the user's history available. Answering to which of the previous explorations are *good predictors* for the next exploration creates the probability distribution. Therefore, we propose variants of the Markov process to accommodate weighting of more than one history state. So, each of the previous explorations are used to predict the future explorations, combined in different ways. It is worth noting that rather than compute β and higher powers of the transition matrix, these may be directly estimated using the training data. In practice, the state probability vector $\alpha(t)$ can be normalized and threshold in order to select a list of *probable states* that the user will choose.

3.2 Predictive Analysis

The implementation of Markov models into a web server makes possible four operations directly linked to predictive analysis. In the first one, the server supports Markov models in a predictive mode. Therefore, when the user sends an exploration request to the web server, this later predicts the probabilities of the next exploration requests of the user. This prediction depends of the history of the user requests. The server can also support Markov models in an adaptive mode. Therefore, it updates the transition matrix using the sequence of requests that arrive at the web server. In the second one, prediction relationship, aided by Markov models and statistics of previous visits, suggests to the user a list of possible scenes, of the same or different web bases, that would be of interest to him, and then the user can go to next. The prediction probability influences the order of scenes. In the current framework, the predicted relationship does not strictly have to be a scene present in the current web base. This is because the predicted relationships represent user traversal scenes that could include explicit

user jumps between disjointing web bases. In the third one, there is generation of a sequence of states (scenes) using Markov models that predict the sequence of states to visit next. The result returned and displayed to the user consists of a sequence of states. The sequence of states starts at the current scene the user is browsing. We consider default cases, such as, if the sequence of states contains cyclic state, they are marked as "explored" or "unexplored". If multiple states have the same transition probability, a suitable technique chooses the next state. This technique considers the scene with the shortest duration. Finally, when the transition probabilities of all states from the current state are too weak, then the server suggests to the user, the go back to the first state. In the fourth one, we refer to web bases that are often good starting points to find documents, and we refer to web bases that contain many useful documents on a particular topic. The notion of profiled information focuses on specific categories of users, web bases and scenes. The web server iteratively estimate the weights of profiled information based on the Markovian transition matrix.

3.3 Path Analysis and Clustering

To reduce the dimensionality of the Markov transition matrix β, a clustering approach is used. It reduces considerably the number of states by clustering similar states into *similar groups*. The reduction obtained is about log N, where N is the number of scenes before clustering. The clustering algorithm is a variant of k-medoids, inspired of [19]. The particularity of the algorithm (algorithm 1) is the replacement of sampling by heuristics. Sampling consists of finding better clustering by changing one medoid. However, finding the best pair (medoid, item) to swap is very costly ($O(k(nk)2)$). That is why, heuristics have been introduced in [19] to improve the confidence of swap (medoid, data item). To speed up the choice of a pair (medoid, data item), the algorithm sets a maximum number of pairs to test (num_pairs), then choose randomly a pair and compare the dissimilarity. To find the k medoids, our algorithm begins with an arbitrary selection of k objects. Then in each step, a swap between a selected object O_i and a non-selected object O_h is made, as long as such a swap would result in an improvement of the quality of the clustering. In particular, to calculate the effect of such a swap between O_i and O_h , the algorithm computes the cost C_{jih} for all non-selected objects O_j. Combining all possible cases, the total cost of replacing O_i with O_h is given by: $T_{cih} = \sum C_{jih}$. The algorithm 1 is given bellow.

```
Clustering()
  {
   Initialize num_tries and num_pairs;
   min_cost = infinitive;
   for k = 1 to num_tries do
       current=k; randomly selected items in the entire data set.
       L=1;
       repeat
             xi = a; randomly selected item in current
             xh = a; randomly selected item in {entire data set current}.
             if TCih  < 0 then
                 current = currentxi+xh;
```

```
            else
                j = j+1;
            end if
        until (j <= num_pairs)
        if min_cost < cost(current) then
               best = current;
        end if
    end for
    return   best;
}
```

Algorithm 1. Clustering function

The algorithm go on choosing pairs until the number of pair chosen reach the maximum fixed. The medoids found are very dependant of the k first medoids selected. So the approach selects k others items and restarts num_tries times (num_tries is fixed by user). The best clustering is kept after the num_tries tries.

4 Approach Implementation

We have implemented the approach, and particularly the clustering method of navigation sessions. Our objective through this implementation is articulated around two points. The first one concerns the study of the different problems that held when we deal with a great number of sessions represented by Markov models. The second one concerns the extraction of the most representative behaviors of the web site. A representative behavior is represented by a Markov model (pages, transitions between pages and the average time in each page) for which the access frequencies to their pages are homogeneous. It means that, the ideal representative user behaviors have not pages accessed frequently and pages rarely accessed, such as presented in the figure presented bellow. Figure 2 represents page frequencies calculated on the basis of the exploration session database, extracted from KDD Cup logs of a commercial web site that will be detailed bellow. In the figure, a curve in two-dimension space is shown In ab-

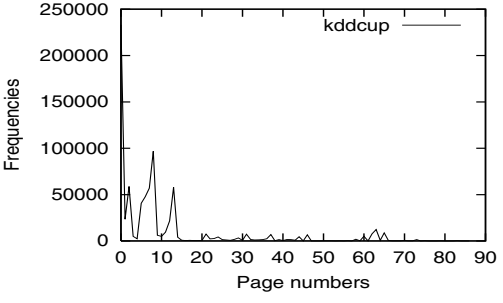


Fig. 2. Classical curve form

scise, we have the web page numbers. In the ordinate, we have the frequencies, which measure the number of access to the concerned page. The highest number in abscise represents the page numbers of the target web site. In this example, we have 90 pages. The frequency of the page number 0 is between 200000 and 250000. It means that the page number 0 has been accessed more than 200000 times and less than 250000 times. It is the highest frequency in the curve. We can deduce that the page number 0 corresponds to the root page or the main page of the web site. The frequency of the page number 80 is equal to 0. It means that no user accessed to this page. We note that the most frequently accessed pages are between 0 and 15; the less frequently accessed pages are between 15 and 46, and between 60 and 70. The pages between 48 and 58 and between 75 and 90 have never been accessed. Globally, the figure represents a typical state of web site access, where some pages are frequently accessed (ex. page number 0), others are rarely accessed (ex page number 30), and others are not accessed at all (ex. page number 90). In this typical state of web site pages, the cardinality of pages that are frequently accessed is generally less than the cardinality of pages that are rarely or never been accessed. Again, in the ideal result of our approach, we should obtain Markov models composed of a sub set of the target web site pages, and the curve of page frequencies should be homogeneous. It means that the curve should be stable, like in the figure 3. In the figure 3 the

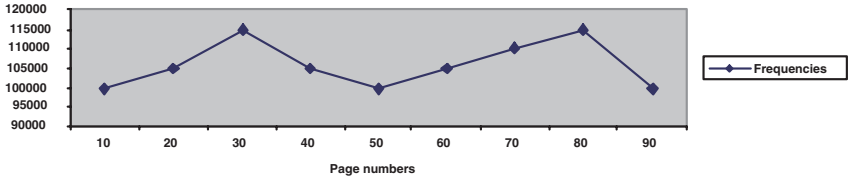


Fig. 3. Ideal curve form

page frequencies are between 100000 and 120000 access. So there is a stability of the curve compared to the classical curve form where the page frequencies are between 0 and 250000. This figure is just an example of an ideal curve form where the best behaviors contain pages that have homogeneous page frequencies.

4.1 Data Set

The used data set is provided by KDDCup (www.ecn.purdue.edu/KDDCUP/) which is a yearly competition in data mining that started in 1997. It's objective is to provide data sets in order to test and compare technologies (prediction algorithms, clustering approaches, etc.) for e-commerce, considered as a "killer

domain" for data mining because it contains all the ingredients necessary for successful data mining. The ingredients of our data set include many attributes (200 attributes), and many records (232000). Each record corresponds to a session.

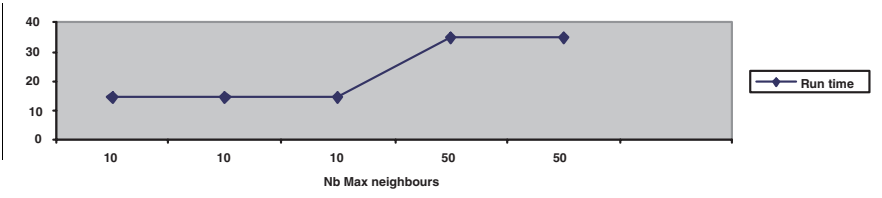
4.2 Results

The tests were carried out on a PC Pentium III with 500 MHz and 256 MB of RAM on the Data set of sessions composed of 90785 sessions (individual Markov models). In our tests, we considered different number of classes, iterations and maximum number of neighbors to compute run time and clustering distortion. We will focus our results on run time and distortion obtained when varying the maximum number of neighbors. So we fixed the number of classes and iterations to respectively 4 and 5. For different numbers of classes and iteration, we obtain similar results.

We tested the approach is the data set composed of 90785 markov models, and we supposed that there are 4 typical user behaviors (number of classes equal to 4) and five iteration of the clustering algorithm (number of iteration equal to 5). Previous experiments [19] proved that the distortion is conversely proportional to the number of iterations. That is why we concentrate our experiments on Run time and distortion values on the basis of respectively numbers of classes (clusters) and iterations. On the basis of the figure curve, we can highlight the

Table 1. Run Time (minutes) proposional to the number of neighbors

Run time	Classes	Iterations	Max Neighbors	Medoids	Distortion
15mn	4	5	10	123023, 294044, 328940, 343287	139004.3
15mn	4	5	10	13137, 145762, 387937, 21549	133472.47
15 mn	4	5	10	15368, 59465, 235239, 101451	136211.69
35 mn	4	5	50	101467, 233739, 145567, 5565	130836.16
35 mn	4	5	50	352195, 185969, 246696, 3782	131280.61



following conclusions. The run time execution is proportional to the maximum number of neighbors. For 10 neighbors, we have 15 minutes run time. For 50 neighbors, we have 35 minutes of execution. We think that the run time will be very high when the number of iteration and classes are high. However the

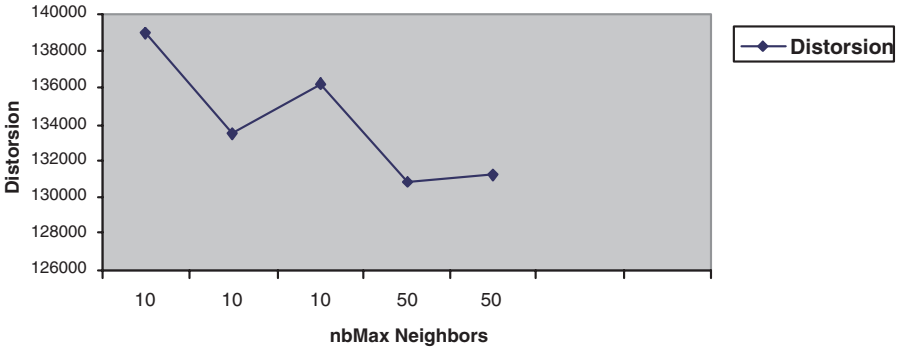


Fig. 4. Distortion conversely proportional to the number of the maximum neighbors

run time is less increasing than the maximum number of neighbors (figure 1). Another remark concerns the distortion (figure 4). The good quality of distortion is proportional to the maximum number of the neighbors. More generally, the results of tests showed some interesting points.

- The first point sub-lined the necessity to clean carefully the data set and to select the useful attribute before any application of the approach. In the data collections, we have a relation table with 200 attributes, and few of them are really useful to achieve our objective. We use only 19 attributes that specify the identification of the web pages, the identifier of the user session, the link relations between web pages and the time spent by the user in each web page.
- The second point sub-lined the necessity to create a new data set suitable for our approach. The original data set contain 230000 sessions, and only 90785 sessions are useful for our approach.
- The third point notes that the features of some attributes have been deleted, because they contain confidential information. So we don't know if they are useful or not in the quality of results as we don't know any thing about these attributes.
- The fourth point showed that the gravity centers of clusters are too small. The original sessions to be grouped are composed of 90 states that correspond to 90 pages visited or not by the user. However the gravity centers of clusters, obtained by our approach, are sessions composed of few pages, in several cases we obtain in our experiments gravity centers with less than 5 pages. We may explain this by the fact that the gravity center of a cluster represents the most typical session. And the most typical session is shared by the whole sessions in the cluster. And the shared point is necessary small when we consider a big number of sessions. The different tests showed that higher is the cardinality of the cluster, lesser is the volume of the gravity center. We think that this property is interesting to make accurate decision because the

site administrator obtains simple and easy to interpret gravity centers, as they are composed of few states and transitions.

- The fifth point concerns the sparse property of the Markov models of sessions. The original Markov models are high dimensional and too sparse. Each session is represented by a high number of states (90 states) and transitions, however not all state are used. This is the result of the fact that the data set corresponds to a web site composed of many pages, and few number of these pages are used in a session. Our approach is addressed to such voluminous sites. The problem of the high dimension and sparse Markov model matrix is that it needs important resources: too large central memory, powerful processor and a clustering algorithm adapted to this high dimensionality. In our experiment, we considered 90 pages, however many commercial web sites consider hundred pages.
- The sixth point concerns how web site administrators may use the results of our experiments. That is good to obtain the most representative behaviors, but how the representative behaviors (gravity centers of behavior clusters) may be exploited in the real e-commerce environment.

5 Conclusion

In this paper, we have presented a method to cluster and predict web surfer actions, based on their surfing patterns using Makov Models. These models, very suitable in modeling sequential data, are used to characterize the similarity between web sequences. To make this prediction possible, three concepts have been highlighted. The first one represents user exploration sessions by Markov models. The second one avoids the problem of Markov model high-dimensionality and sparsely by clustering web documents, based on their content, before applying Markov analysis. The third one extracts the most representative user behaviors (represented by Markov models) by considering a clustering method. Results shown that the proposed method is able to infer the natural clusters with patterns characterizing a complex and noisy data like the KDDCup ones.

References

1. Rabiner, L.R.: A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. of IEEE 77(2) (1989) 257–286
2. Hu, J., Brown, M.K., Turin, W.: HMM based on-line handwriting recognition. IEEE Trans. Pattern Analysis and Machine Intelligence, 18(10) (1996) 1039–1045
3. Hughey, R., Krogh, A.: Hidden Markov Model for sequence analysis: extension and analysis of the basic method. Comp. Appl. in the Biosciences 12 (1996) 95–107
4. Eickeler, S., Kosmala, A., Rigoll, G.: Hidden Markov Model based online gesture recognition. Proc. Int. Conf. on Pattern Recognition (ICPR) (1998) 1755–1757
5. Jebara, T., Pentland, A.: Action Reaction Learning: Automatic Visual Analysis and Synthesis of interactive behavior. In 1st Intl. Conf. on Computer Vision Systems (ICVS'99) (1999)

6. Rabiner, L. R., Lee, C.H., Juang, B. H., Wilpon, J. G.: HMM Clustering for Connected Word Recognition. Proceedings of IEEE ICASSP (1989) 405–408
7. Lee, K. F.: Context-Dependent Phonetic Hidden Markov Models for Speaker Independent Continuous Speech Recognition. IEEE Transactions on Acoustics, Speech and Signal Processing 38(4) (1990) 599–609
8. Smyth, P.: Clustering sequences with HMM, in Advances in Neural Information Processing (M. Mozer, M. Jordan, and T. Petsche, eds.) MIT Press 9 (1997)
9. Smyth, P.: Clustering sequences with hidden markov models. In M.C. Mozer, M.I. Jordan, and T. Petsche, editors, Advances in NIPS 9, (1997)
10. Li, C., Biswas, G.: Clustering Sequence Data using Hidden Markov Model Representation, SPIE'99 Conference on Data Mining and Knowledge Discovery: Theory, Tools, and Technology, (1999) 14–21
11. Li, C., Biswas, G.: A Bayesian Approach to Temporal Data Clustering using Hidden Markov Models. Intl. Conference on Machine Learning (2000) 543–550
12. Schwarz, G.: Estimating the dimension of a model. The Annals of Statistics, 6(2) (1978) 461–464
13. Stolcke, A., Omohundro, S.: Hidden Markov Model Induction by Bayesian Model Merging. Hanson, S.J., Cowan, J.D., Giles, C.L. eds. Advances in Neural Information Processing Systems 5 (1993) 11–18
14. Cheeseman, P., Stutz, J.: Bayesian Classification (autoclass): Theory and Results. Advances in Knowledge discovery and data mining, (1996) 153–180
15. Law, M.H., Kwok, J.T.: Rival penalized competitive learning for model-based sequence Proceedings Intl Conf. on Pattern Recognition (ICPR) 2, (2000) 195–198
16. Cadez, I., Ganey, S. and Smyth, P.: A general probabilistic framework for clustering individuals. Technical report, Univ. Calif., Irvine, March (2000)
17. Smyth, P.: Probabilistic model-based clustering of multivariate and sequential data. In Proc. of 7th Int. Workshop AI and Statistics, (1999) 299–304
18. Ni, Z.: Normal orthant probabilities in the equicorrelated case. Jour. Math. Analysis and Applications, n° 246, (2000) 280–295
19. Ng, R.T. and Han, J.: CLARANS: A Method for Clustering Objects for Spatial Data Mining. TJDE 14(5), (2002) 1003–1016