Jiming Liu Yiuming Cheung
Hujun Yin (Eds.)

Intelligent Data Engineering and Automated Learning

4th International Conference, IDEAL 2003
Hong Kong, China, March 21-23, 2003
Revised Papers

Springer

Volume Editors

Jiming Liu
Yiuming Cheung
Hong Kong Baptist University
Department of Computer Science
7/F, Sir Run Run Shaw Building, Hong Kong
E-mail: {jiming/ymc}@comp.hkbu.edu.hk

Hujun Yin
University of Manchester Institute of Science and Technology
Department of Electrical Engineering and Electronics
Manchester, M6O 1QD, UK
E-mail: H.Yin@umist.ac.uk

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

CR Subject Classification (1998): H.2.8, F.2.2, I.2, F.4, K.4.4, H.3, H.4

ISSN 0302-9743
ISBN 3-540-40550-X Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin GmbH
Printed on acid-free paper SPIN: 10927861 06/3142 5 4 3 2 1 0

Extraction of Dynamic User Behaviors from Web Logs

Younes Hafri[1], Bruno Bachimont[1], and Peter Stachev[3]

[1] Institut National de l'Audiovisuel, 4 avenue de l'Europe
94366 Bry-sur-Marne Cedex, France
{yhafri,bbachimont}@ina.fr
[2] Kettering University, USA
pstanchev@kettering.edu
3, Kettering University, Flint, MI 48504, USA

Abstract. Our paper proposes an approach which makes possible prediction of future states to be visited in k steps corresponding to k web pages hyper-linked, based on both content and traversed paths. 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.
The original application of the approach concerns the exploitation of multimedia archives in the perspective of the Copyright Deposit that preserves French's WWW documents. The approach may be the exploitation tool for any web site.

1 Introduction

The future web sites, particularly web services, will amass detailed records of which uses their web documents and how people use them. However, this "new" future industry highlights the most ambitious effort yet to gather disparate bits of personal information into central databases containing electronic information on potentially every person who surfs the web bases. Profiling explorations are in the interest of the user, providing more customized and directed services through the web bases.

Profiling is a business concept from the marketing community, with the aim of building databases that contain the preferences, activities and characteristics of clients and customers. It has been for a long time a part of the commercial sector. It has developed significantly with the growth of e-commerce, the Internet and information retrieval. The goal of profiling web exploration is to have the most complete picture of the web users we can. Mining the web servers also achieves the used cookies, carries out profiling scenes off-line. The long-term objective is to create associations between potential commercial web sites and commercial

J. Liu et al. (Eds.): IDEAL 2003, LNCS 2690, pp. 584–595, 2003.

marketing companies. These sites make use of specific cookies to monitor client's explorations at the web site and record data that users may have provided to the exploration server. User likes, dislikes, browsing patterns and buying choices are stored as a profile in a database without user knowledge or consent.

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. More precisely, this paper proposes an approach that extracts automatically web user profiling based on user navigation paths. Web user profiling consists of the best representative behaviors, represented by Markov models. 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.

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. Section 5 presented some implementation results. Finally, section 6 concludes the paper.

2 Related Works and Contribution

The problem of modeling and predicting user's accesses on web site has attracted a lot of research interest. It has been used to improve the web performance through various caching strategies such as [Agg 99], [Bad 98], [Fan 99], [Per 99] and prefetching [Fan 99], [Sch 99] [Pad 96]; adaptive web sites [Per 99], web log mining [Zai 98]; intelligent agents that detects user web topics [Chu 97]; extraction of the most interesting pages [Woo 96], [Sha 97]; recommend related pages improve search engines, personalize the browsing in a web site and continuous Markov models to influence caching priorities between primary, secondary and tertiary storages [Stu 98].

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 [Rab 89]. 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 [Rab 89], handwritten character recognition [Bro 96], DNA and protein modeling [Hug 96], gesture recognition [Eic 98], behavior analysis and synthesis [Jeb 00], 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 [Rab 89], [Lee 90], all related to speech recognition. The first interesting approach not directly linked to speech issues was presented by Smyth [Smy 97], 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 [Bis 99], [Bis 00], 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 [Bis 99], the former issue is addressed using standard approach, like Bayesian Information Criterion [Sch 78], and extending to the continuous case the Bayesian Model Merging approach [Sto 93]. 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 the second paper [Bis 00], the same problems are addressed in terms of Bayesian model selection, using the Bayesian Information Criterion (BIC) [Sch 78], and the Cheesman-Stutz (CS) approximation [Che 96]. Although not well justified, much heuristics is 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 [Law 00], 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 (see example bellow) 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 [Cad 00], [Smy 97], [Smy 99]. 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.

The core issue in prediction is the development of an effective algorithm that deduces the future user requests. The most successful approach towards this goal has been the exploitation of the user's access history to derive predictions. A thoroughly studied field in this area is Web prefetching. It shares all the characteristics that identify the Web prediction applications like the previously mentioned.

In general, there is two prefetching approaches. Either the client will inform the system about its future requirements or, in a more automated manner and transparently to the user, the system will make predictions based on the sequence of the client's past references. The first approach is characterized as informed and the latter as predictive.

3 User Profiling and Web Exploration

3.1 Web Features

We consider a robot named WPE (Web Profile Extractor) that extracts, invisibly, features from web user explorations. WPE is synchronized with the playing of scene that resulted from user browsing or queries.

To enable WPE to compile meaningful reports for its client web sites and advertisers and to better target advertising to user, WPE collects the following types of non-personally-identifiable feature about users who are served via WPE technology:

- User IP address. A unique number assigned to every computer on the Internet. Information, which WPE can infer from the IP address, includes the user's geographic location, company, and type and size of organization.
- User domain type (com, net or edu.).
- Standard feature included with every communication sent on the Internet. Information, which WPE can infer from this standard feature, includes user browser version and type (i.e., Netscape or Internet Explorer), operating system, browser language, service provider (i.e., Wanado, Club internet or AOL), local time, etc.
- Manner of using the scene or shot visit within a WPE client's site.
- Affiliated advertisers or web publishers may provide access with non-personally-identifiable demographic feature so that user may receive ads that more closely match his interests. This feature focuses on privacy protections as the access-collected non-personally-identifiable data. WPE believes that its use of the non-personally-identifiable data benefits the user because it eliminates needless repetition and enhances the functionality and effectiveness of the advertisements the users view which are delivered through the WPE technology.

3.2 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 [Niz 00], 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 graph representing a simple Markov chain of five nodes and their corresponding transitions probabilities.

Markov model creations depend of an initial table of sessions in which each tuple corresponds to a user access.

```
INSERT INTO [Session] (id, idsceneInput)
SELECT[InitialTable].[SessionID],
      [InitialTable].[SessionFirstContent]
FROM OriginalTable
GROUP BY [InitialTable].[SessionID],
         [InitialTable].[SessionFirstContent]
ORDER BY [InitialTable].[SessionID];
```

3.3 Markov Models

A set of three elements defines a discrete Markov model: $< \alpha, \beta, \lambda >$

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 a.

Each transition contains the identification of the session, the source scene, the target scene, and the dates of accesses. This is an example of transition table creation.

```
SELECT regroupe.Session_ID, regroupe.Content_ID AS idPageSource,
       regroupe.datenormale AS date_source,
       regroupe2.Content_ID AS idPageDest,
       regroupe2.datenormale AS date_dest INTO transitions
FROM [2] regroupe sessions from call4] AS regroupe,
     [2] regroupe sessions from call4] AS regroupe2
WHERE (((regroupe2.Session_ID)=regroupe!Session_ID)
      AND ((regroupe2.Request_Sequence)=regroupe!Request_Sequence+1);
```

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.

3.4 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 user requests. The server can also supports 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 Markovian transition matrix.

3.5 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 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.

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

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.

More precisely, the data come from a concrete study carried out with truths users on a commercial site named www.gazelle.com (now on close state). The site sold data mining technologies. The data set describes Gazelle.com customer sessions that correspond to customer explorations.

4.2 Data Preparation

The provided data were generated by the requests of the users on the site and were recorded by the waiter web. We thus had a whole of tuples of which each one related to a request on the waiter. Each tuple was described by more than 200 attributes, as in the previous section, of which little was interesting for us.

Indeed, there was many information concerning the requests (execution time, chains of connection ...), and also all the answers to a form of user information, that is to say in all more than 200 attributes. Thus, it was necessary to consider the attributes that are useful to create Markov model ingredients (states, transitions, probabilities), and to present them under a format useful for the treatments. For example, we deleted the attribute that calculates the number of entries in a web site, because it is not interesting, as we suppose that a web site has only one entry. Generally, all the users entered by the same page (main page).

The following database schema was used. It is composed of 19 attributes that are of interest for us to construct Markov models. So we reduced the number of attribute from 200 attributes to 19 attributes.

```
Number, Request Query String, Request Dates, Date_Time, Request
Sequence, Template Request, Visit Dates,Date_Time,Request Dates,First
Request Date_Time, Session Cookie ID,Session ID,Count,First Query
String,First Referrer, First Template, First ID,Content Level,Content
ID, Path
```

After selecting the suitable attribute, we create a table of sessions in the following way. The original data set is stored in a file "clicks.data" that contains large number of tuples. Each tuple contains a sequence of values of the attributes separated by commas. The file weigh is about 1.2 GB. Thus, it was impossible to load it in the central memory, and use it directly by the codes of our approach. That is why, we used relational database management system (Microsoft Access) to store the data set in a relational table. So the access to tuples is managed by the relational data base management system. We used such tool because it is easy to use, and it supports very powerful module of importation. The data set stored in the table concerns exclusively selected attributes. To identify the meaning of he values, we used the file "clicks.names" in which each line contains an attribute and its description.

The table contains some non-valid entries due to errors of importation (the corresponding tuples were unusable). We thus selected the good tuples and created another table under mySQL. So, from 232000 tuples that corresponds to discounted sessions, we obtained only 90785 sessions that are valid for Markov model modeling. That is due to the presence of null data in the file starting, and a significant number of sessions comprising only one request. We noticed that there is an average of 5 transitions by session, so an average of 5 pages from 90 pages by session.

4.3 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.

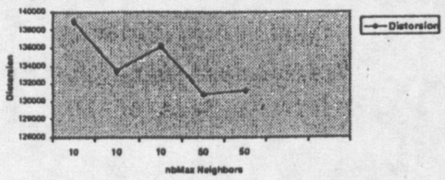

Fig. 1. Distortion conversely proportional to the number of the maximum neighbors

On the basis of the figure curve, we can highlight the 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 run time is less increasing than the maximum number of neighbors (figure ??). Another remark concerns the distortion (figure 1). 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

The objective of this paper is to propose an approach that extracts automatically web user profiling based on user navigation paths. Web user profiling consists of the best representative behaviors, represented by Markov models. 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.

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

(15.) * Younes Hafri, Bruno Bachimont, Peter Stanchev, “Extraction of Dynamic user behaviors from Web log”, ***Lecture Notes in CS*** 2690, 2003, 584-595.