Automatic categorization of web pages and user clustering with mixtures of hidden Markov models

Alexander Ypma and Tom Heskes

SNN, University of Nijmegen
Geert Grooteplein 21,
6525 EZ Nijmegen, The Netherlands
Email: {ypma,tom}@mbfys.kun.nl
Web: www.mbfys.kun.nl/~ypma

Abstract. We propose mixtures of hidden Markov models for modelling clickstreams of web surfers. Hence, the page categorization is learned from the data without the need for a (possibly cumbersome) manual categorization. We provide an EM algorithm for training a mixture of HMMs and show that additional static user data can be incorporated easily to possibly enhance the labelling of users. Furthermore, we use prior knowledge to enhance generalization and avoid numerical problems. We use parameter tying to decrease the danger of overfitting and to reduce computational overhead. We put a flat prior on the parameters to deal with the problem that certain transitions between page categories occur very seldom or not at all, in order to ensure that a nonzero transition probability between these categories nonetheless remains. In applications to artificial data and real-world web logs we demonstrate the usefulness of our approach. We train a mixture of HMMs on artificial navigation patterns, and show that the correct model is being learned. Moreover, we show that the use of static 'satellite data' may enhance the labeling of shorter navigation patterns. When applying a mixture of HMMs to real-world web logs from a large Dutch commercial web site, we demonstrate that sensible page categorizations are being learned.

Keywords: web usage mining, data mining, navigation patterns, automatic categorization, clustering, hidden Markov models, mixture models, user profiling

1 Introduction

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 (connecting 'remote' pages that are often visited cooperatively, detecting common 'exit traces'), 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 [1, 12, 13]. 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 this paper we extend this existing approach by learning the most likely categorization of a page along with the inter-category transitions, i.e. we model a clickstream of a particular surfer type by a hidden Markov model (HMM). In order to incorporate heterogeneity of users (there will be several types of surfers on a web site), we postulate a *mixture* of HMMs (mHMM) to model clickstreams. In our formulation we make the membership of a user to a user type explicit, which then allows for inclusion of additional user data.

In the following section we introduce the problem with a small example, and then describe the model for clustering of web surfers. We give the update equations for training the mHMM model with the Expectation-Maximization algorithm and describe how to incorporate prior knowledge and additional (static) user information. Then we apply the method to artificial data and to logs from a large commercial Dutch web site, discuss both method and results and draw conclusions.

2 Mixtures of hidden Markov models for web usage mining

In order to clarify the web mining problem, we start with a small example.

2.1 Web mining: an example

Consider a web log file, that contains entries of the following form:

```
194.79.42.11 - [06:54:49] "GET /common/graphics/tools_catalog_on.gif HTTP/1.0" 304 0 "-"
146.8.233.251 - [06:55:03] "HEAD / HTTP/1.0" 500 305 "-"
217.142.71.136 - [06:55:02] "GET /br1/custsvc/cs_category_list.jsp
209.199.168.175 - [06:54:50] "GET /en/financiele/results.html HTTP/1.0" 404 207 "-"
146.8.233.251 - [06:54:58] "HEAD / HTTP/1.0" 302 0 "-"
213.79.178.45 - [06:54:42] "GET /common/graphics/products/product.gif HTTP/1.0" 200 1842 "-"
194.79.42.11 - [06:54:49] "GET /common/graphics/welcome HTTP/1.0" 304 0 "-"
146.8.233.251 - [06:55:04] "HEAD / HTTP/1.0" 302 0 "-"
217.142.71.136 - [06:54:40] "GET /common/tools_header.gif HTTP/1.0" 200 353 "/br1/index.jsp"
```

We want to derive information about the click behaviour of web users automatically from this file[1], since manual processing is not doable. We assume that each web user can be uniquely identified; in this paper we make the (simplifying) assumption that a user's IP-adress acts as a unique identifier. Each different page that is requested from a web site gets a unique page id (which is only based on

[1] The IP-adresses and URL-s in the displayed excerpt were anonymized, so any trace to actual persons or web-sites is coincidental

its URL, not on additional page information like keywords or a semantic description). If we retain the ordering in the page requests and if we assume that a time difference of 30 minutes between two http-requests of the same user indicates different sessions, we end up with several (possibly intertwined) clickstreams of the following form:

```
Y11 = 8 9 10 11 11 11          Y12 = 8 8 9 12 13 13 14 15 14
Y21 = 1 2 4 3 5 7 6 5 7 6 4 3  Y22 = 1 2 2 2 2 4 2 4 3
```

where $\mathbf{Y^{1,j}}$ and $\mathbf{Y^{2,j}}$ are clickstreams no. $j = 1, 2$ for certain **user 1** and **user 2**, respectively. The problem is now to assign a new clickstream, like

```
Ynew = 8 9 9 10 11 11 153 154 155 9 9 9 8 9 10 11 11 11 11 24
```

to the user type that it resembles best. We learn a model for each user type k from a set of training patterns $\{Y_t^{i,j}\}$ and we assume that click behaviour can be described by modelling transitions between **page categories** $X_t \in 1, \ldots, m$, rather than between individual pages $Y_t^{i,j} \in 1, \ldots, M$. This gives a computational advantage since there are less categories than individual pages ($m < M$) and it is more meaningful to model inter-category transitions than inter-page transitions. To continue our example, one type of user (e.g. "general-interest user") may typically start by doing a *search*, followed by *downloading* a report or a piece of software and finally checks the *latest corporate news*. Another type of user (e.g. "device-interest user") may first enter the pages concerning the *retail shop* of the company, then he browses through the set of *consumer products*, tries to *troubleshoot* the device that he owns and concludes by checking out new *promotions* offered by the company. Here, the *italic* phrases are page categories and a particular web page will be assigned to one (or a few) categories.

In our approach, we learn the probability that a user of type k jumps from page category i to category j, which is represented by entries in the transition matrix $A^k(i, j)$. Moreover, we learn the categorization of a web page by determining the probability that a certain page $Y_t = l$ is observed in a clickstream at time t, given that the page category at time t is $X_t = i$. This is represented by entries in the observation matrix $B^k(i, l)$; if we assume a common categorization for all user types, we end up with just one common observation matrix B.

2.2 Problem formulation

Consider a population of web surfers, denoted by $\{i\}, i = 1, \ldots, N$. Each surfer i generates n_i clickstreams $Y^{i,j} = \{Y_t^{i,j}\}, t = 1, \ldots, T_{ij}$, i.e. possibly unequal-length ordered sequences of pages that have been visited in a surfing-session. An element $Y_t^{i,j}$ denotes the observed page id (in the range 1, ..., M) at time t of clickstream no. j that is generated by user no. i. All clickstreams generated by surfer i are collected into $Y^i = \{Y^{i,1}, \ldots, Y^{i,n_i}\}$.

We assume K clusters of web surfers and our overall model Θ consists of separate models for each cluster, $\Theta = \{\Theta_k\}, k = 1, \ldots, K$. The cluster label ("surfer type") is given by the variable $C \in 1, \ldots, K$. We model the dynamics

in the traces at the level of page categories X, which are hidden state variables (unknown beforehand) in a hidden Markov model. The inter-category transitions are governed by a (cluster dependent) transition matrix A^k and the initial state distribution vector Π^k, the categorization of a page is given by the observation matrix B^k. If we have additional (static) information about a surfer (like demographic information, area code, etc.) this may also give us an indication about the surfer type. If the B matrix is *shared* (section 2.5), we have the graphical model shown in figure 1.

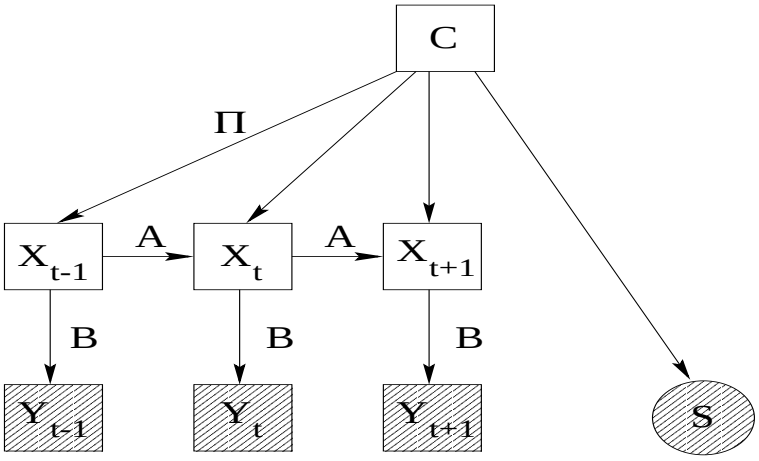

Fig. 1. A mixture of hidden Markov models and additional user data, presented as a graphical model. The variables Y_t (discrete) and S (continuous or discrete) are observed, the category variables X_t (discrete) are hidden. Note that in this figure the observation matrix B is independent of the cluster label C (which occurs with a shared B matrix), since there are no arrows from C to the Y-nodes. Furthermore, note that S and Y are independent given C.

2.3 Dynamic model

We now address the dynamic part of the model (with cluster-dependent observation matrices B^k), the static part and the sharing of B is included in section 2.5. The likelihood of the (dynamic) data from all users is

$$P(Y|\Theta) = \prod_{i=1}^{N} P(Y^i|\Theta) \tag{1}$$

which expresses the assumption that users are independent of each other. The likelihood of the (dynamic) data of user i is given by the mixture

$$P(Y^i|\Theta) = \sum_{k=1}^{K} P(Y^i|c_i = k, \Theta_k)\alpha_k \tag{2}$$

where

$$P(Y^i|c_i = k, \Theta_k) = \prod_{j=1}^{n_i} P(Y^{i,j}|c_i = k, \Theta_k) \tag{3}$$

is the likelihood of user data i given that the user type is known. The latter equation expresses that different surfing sessions of a user are modelled as independent events given the user type[2]. The mixture coefficients obey $\sum_k \alpha_k = 1$. The likelihood of a particular sequence $Y^{i,j}$, given that user i is in cluster k, is given by the well-known quantity for HMMs

$$P(Y^{i,j}|c_i = k, \Theta_k) = \sum_X P(Y^{i,j}, X|c_i = k, \Theta_k) \tag{4}$$

where

$$\begin{gathered} P(Y^{i,j}, X|c_i = k, \Theta_k) = \\ \Pi^k(X_1)B^k(Y_1^{i,j}|X_1) \prod_{t=1}^{T_{ij}-1} A^k(X_{t+1}|X_t)B^k(Y_{t+1}^{i,j}|X_{t+1}) \end{gathered} \tag{5}$$

2.4 EM algorithm

We can train (the dynamic part of) the model from the previous section using the EM algorithm. In the update equations we use the following definitions [8]:

$$\begin{aligned} \gamma_t^{i,j,k}(x) &= P(X_t = x|Y^{i,j}, \Theta_k) \\ \xi_t^{i,j,k}(x, x') &= P(X_t = x, X_{t+1} = x'|Y^{i,j}, \Theta_k) \end{aligned} \tag{6}$$

E-step This involves an update of the (hidden) memberships c_{ik}:

$$c_{ik} := P(c_i = k|Y^i, \Theta) = \frac{\alpha_k P(Y^i|c_i = k, \Theta_k)}{\sum_l \alpha_l P(Y^i|c_i = l, \Theta_l)} \tag{7}$$

[2] Session-to-session effects are not always entirely explained by the user label. If significant user-specific session correlations are present, equation (3) should be adapted.

M-step This involves an update of the parameters $\alpha_k, \Pi^k, A^k, B^k$:

$$\begin{aligned}
\hat{\alpha}_k &= \frac{1}{N}\sum_{i=1}^{N} P(c_i = k|Y^i, \Theta) \\
\hat{\Pi}^k(x) &= \frac{\sum_i c_{ik} \sum_{j=1}^{n_i} \gamma_1^{i,j,k}(x)}{\sum_i c_{ik} \sum_{j=1}^{n_i} \sum_x \gamma_1^{i,j,k}(x)} \\
\hat{A}^k(x,x') &= \frac{\sum_i c_{ik} \sum_{j=1}^{n_i} \sum_{t=1}^{T_{ij}-1} \xi_t^{i,j,k}(x,x')}{\sum_i c_{ik} \sum_{j=1}^{n_i} \sum_{t=1}^{T_{ij}-1} \gamma_t^{i,j,k}(x)} \\
\hat{B}^k(x,y) &= \frac{\sum_i c_{ik} \sum_{j=1}^{n_i} \sum_{t=1 \wedge Y_t^{i,j}=y}^{T_{ij}} \gamma_t^{i,j,k}(x)}{\sum_i c_{ik} \sum_{j=1}^{n_i} \sum_{t=1}^{T_{ij}} \gamma_t^{i,j,k}(x)}
\end{aligned} \tag{8}$$

2.5 Including static user data and prior information

As pointed out by Smyth [13], once the memberships are made explicit in a mixture model involving both dynamic data Y and static user data S, we can easily combine the two types of information to enhance the labelling of a user. If we assume that the dynamic and static data are independent given the cluster label, we may extend our original 'dynamic' mixture model (2) with static data to $P(Y^i, S^i|\Theta) = \sum_k P_k(Y^i, S^i|c_k, \Theta^k)\alpha_k$, leading to a modified E-step

$$c'_{ik} := P(c_i = k|Y^i, S^i, \Theta) = \frac{\alpha_k P_k(Y^i) P_k(S^i)}{\sum_l \alpha_l P_l(Y^i) P_l(S^i)} \tag{9}$$

The M-step equations for the dynamic and the static model separately remain the same, except that now the joint membership c'_{ik} is employed. It is very likely that additional information is not available for all surfers. In this case, we can set the probability of static data in cluster k to 1.

We remark that prior knowledge on the dynamics can be taken into account in the following manner [9]. Consider a reestimated transition probability of the form $\hat{P}(i,j) = n_{ij}/n_i$, with n_{ij} the transition count from state i to j and n_i the number of transitions from i. If our prior knowledge takes the form of an additional pseudo-sequence of length $\beta + 1$, which is divided into β_{ij} transitions from i to j, the Bayesian MAP estimate is

$$\hat{P}_{\mathrm{MAP}}(i,j) = \frac{n_{ij} + \kappa\beta_{ij}}{n_i + \kappa\beta_i}, \tag{10}$$

where $\beta_i = \sum_j \beta_{ij}, n_i = \sum_j n_{ij}$ and $0 \leq \kappa \leq 1$ determines the extent to which the prior or the data is used. A similar trick can be applied to the 'prior probability' Π over states and the observation probabilities. Especially the latter quantity may easily tend to zero in cases with small sample sizes (limited number of observations, large dimensionality of the observables).

From our update formula for $\hat{B}^k$ it is clear that each mixture component has a private observation matrix. However, for our application the 'interpretation' of a category should preferably not be too different for different user types (e.g. 'Mercedes-Benz.html' should be categorized as 'cars', regardless the user's interests). Therefore, we constrain the observation matrices in all clusters to be equal (a.k.a. *parameter tying*). This has the additional advantage that we decrease the danger of overfitting in cases with a large number of observables (i.e. sites with many pages and relatively small number of visitors).

2.6 Computational complexity

Since the datasets encountered in practice may be very large (a one-day log file of the web site analysed in section 3 is already .25 GB) the scalability of the proposed algorithm is an important issue. We mention that the computational complexity at the level of one EM iteration is

- linear in the total number of sequences $\sum_i n_i$ (and hence in the number of surfers N, assuming that the number of clickstreams per user can be bounded to a reasonable number),
- linear in the number of surfer types K,
- quadratic in the dimensionality of the hidden state space $|X|$, and
- linear in the number of samples in an individual clickstream T_{ij}

During each iteration, the costly step is the inference step where for each of the sequences (i.e. pairs (i, j)) and for each of the mixture components k the quantities $\gamma_t^{i,j,k}(x), \xi_t^{i,j,k}(x, x')$ and c_{ik} (for all surfers i) have to be computed. The cost of inference in an HMM for a particular component and sequence is $\mathcal{O}(|X|^2 T_{ij})$, see [5]. Hence, we see that the algorithm scales linearly with all relevant quantities except for the number of categories. However, we expect that for a certain web site the number of relevant categories will not depend too strongly on the number of considered clickstreams, i.e. if we consider twice as many clickstreams the appropriate number of categories will not increase as much (provided that the initial set of clickstreams was 'rich enough')[3].

By sharing the observation matrix we diminish the danger of overfitting. It is hard to predict how the likelihood surface changes when we include more data and larger models (and therefore how many iterations are necessary for convergence), though for mixtures of Markov chains this does not lead to superlinear behaviour [2].

[3] As an aside, a reviewer remarked that this assumption is indeed a plausible one. Furthermore, in this reviewer's experience, the categorization of pages is straightforward, even with dynamic page creation. This strengthens our assumption that meaningful categories may be obtained by automating the laborious task of page categorization, while at the same time allowing for a better scalable user clustering procedure

3 Experiments

3.1 Artificial data

We generated artificial clickstreams using a 4-component mHMM. Every single HMM in the mixture model had 2 hidden states and was able to emit 3 observables. The A^k, B^k and Π^k parameters were markedly different for each component k. All 4 components were almost equally likely to occur. The generated sequences were distributed over 15 users, where each user received only sequences with the same label (so that the *user* could be labelled meaningfully as well). The resulting user labelling was: $\{1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 4, 4, 4, 4\}$. Users with label 1 produced 3 sequences, others produced 4 sequences. Each sequence had length 250 samples. We then trained an (oversized) 8-component mHMM

Table 1. Learned memberships from 4-component mixture data. Each entry in the table gives the membership c_{ik} of user i to component k; nonzero entries are in boldface.

$_i \backslash ^k$	comp. 1	comp. 2	comp. 3	comp. 4	comp. 5	comp. 6	comp. 7	comp. 8
1	0	0	0.0	0	0.0	**0.0049**	0	**0.9951**
2	0	0	0.0	0	0.0	**0.0003**	0	**0.9997**
3	0	0	0.0	0	0.0	**0.0001**	0	**0.9999**
4	0	0	**1.0**	0	0.0	0.0000	0	0.0000
5	0	0	**1.0**	0	0.0	0.0000	0	0.0000
6	0	0	**1.0**	0	0.0	0.0000	0	0.0000
7	0	0	**1.0**	0	0.0	0.0000	0	0.0000
8	0	0	**1.0**	0	0.0	0.0000	0	0.0000
9	0	0	**1.0**	0	0.0	0.0000	0	0.0000
10	0	0	0.0	0	**1.0**	0.0000	0	0.0000
11	0	0	0.0	0	**1.0**	0.0000	0	0.0000
12	0	0	0.0	0	0.0	**0.9998**	0	**0.0002**
13	0	0	0.0	0	0.0	**1.0000**	0	0.0000
14	0	0	0.0	0	0.0	**0.9986**	0	**0.0014**
15	0	0	0.0	0	0.0	**1.0000**	0	0.0000

on these sequences for 1000 cycles in order to check whether our model was able to learn the correct parameters and user labelling, even if the number of components present in the data was smaller than the number of components in the model. We observed the learned memberships shown in table 1. It is clear that our model learns the correct user labelling; moreover, the number of underlying mixture components can be estimated from this table as well.

In the second experiment we performed parameter tying of the observation matrix. First, a 3-component mHMM was trained for 50 cycles on 34 sequences of 300 samples each, generated from a 3-component mHMM (2 hidden states, 3 observables); the sequences were distributed over 9 users. It was observed that good correspondence exists between the learned and the target A and B matrices (figure 2) and that all users are labelled correctly.

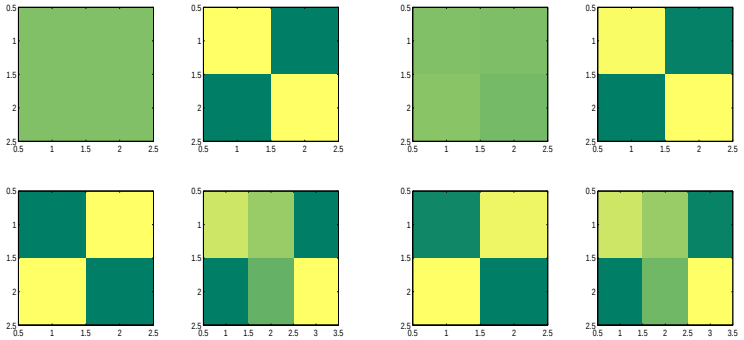

Fig. 2. Left 4 images: target A^1, A^2, A^3, B matrices; right 4 images: learned matrices

We then distributed 83 sequences from a 3-component mHMM (same parameter settings as above: 2 states, 3 observables, shared observation matrix) over 50 users (1 or 2 sequences each), where the sequence length was now much shorter (100 samples). For each user, an additional 6-D static data vector was drawn from a (spherical) Gaussian corresponding to each component; the Gaussians for different components were largely overlapping. We determined the number of erroneously labelled users with either static or dynamic data separately or jointly. The resulting error rates (dynamic: 4 %, static: 32 %, joint: 0 %) indicate that shorter sequences and a larger number of users give rise to erroneous labelling when using dynamical information only; combining dynamic with static information, however, leads to an improved user labelling.

3.2 Automatic categorization of web logs

We applied an mHMM to (an excerpt of) a one-day log file from a large commercial web site in The Netherlands. The raw entries in the file were 'sessionized' [3] and irrelevant entries (like images) were removed. A training set was created out of clickstreams from 400 users. Then we trained a 4-component mHMM with 12 states and common observation matrix (with 134 different observables) for 500 cycles on the training set and we inspected the shared observation matrix B, the per-cluster transition matrices A^k and prior vectors Π^k.

We observed the shared observation matrix displayed in figure 3a. In this figure the horizontal axis denotes page-id, the vertical axis denotes state-id and a greyvalue at position (i, j) denotes the probability of observing page i in state j. The resulting page categorization was: 2,4,6,11: "shop info", 3,5: "start, customer/ corporate/ promotion", 1,8,12: "tools", 7,9,10: "search, download/ products". The 'semantic labeling' was done afterwards by manual inspection of the pages 'assigned to' a state. In the figure, the horizontal axis was reordered in such a way that pages that were assumed to belong to the same category have nearby page-id. We emphasize that this reordering was done *after* the learning phase, and only used to facilitate the presentation of the results. It can be seen

in the figure that similar pages (based on our manual assessment afterwards) are indeed assigned to particular categories.

Inspection of the learned state priors (vertical axis in figure 3b) for each of the 4 user types (horizontal axis) and the state transition matrices (one typical example is plotted in figure 4a) reveals that users are mainly assigned to 2 different user types with markedly different starting states. After inspection of the number of users that were assigned (with a large confidence) to each of the 4 user types, we noticed that the main clustering was w.r.t. user types '2' (app. 250 'general interest' users) and '3' (app. 150 'shop' users). All four transition matrices were fairly similar; the user labeling was apparently done based on the starting state. This can be understood, since 'shop' users are generally starting their session from a different page then the 'general interest' users. The other 2 components were dedicated to one particular user each.

We compared the results with an experiment with 6 assumed states and observed that three main page categories appear to be present: "shop info" (with its own starting page), "tools" and a "general interest" category, which is entered via the general starting page or via an initial search action. The

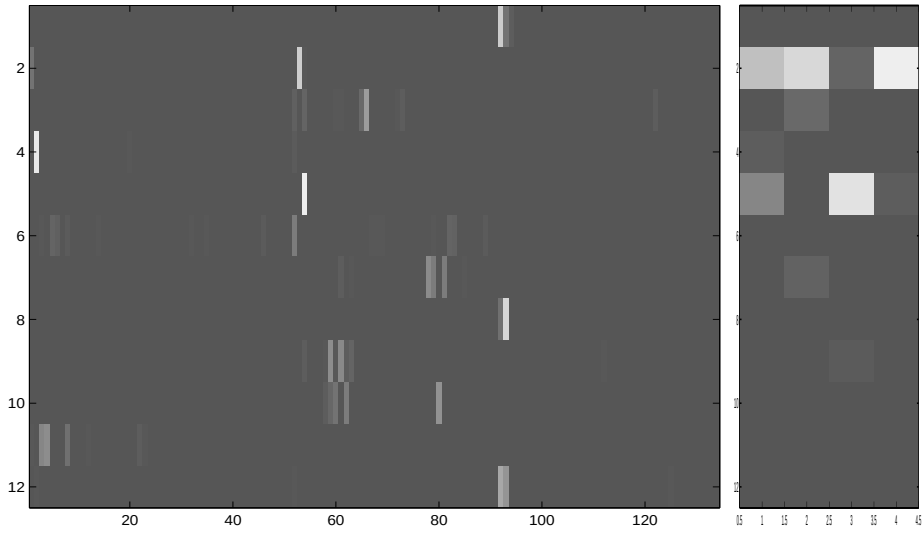

Fig. 3. a. (left): Learned 12-state observation matrix B; the shuffling of the page-ids is such that 1-51 = 'shop', 52-57 = 'general start & error', 58-64 = 'about & search', 65-77 = 'customer service', 78-91 = 'products', 92-94 = 'tools', 95-121 = 'download', 122-130 = 'corporate' and 130-134 = 'promotion'. It can be observed that 'shop' and 'tools' pages are assigned to distinct page categories. Furthermore, states tend to 'specialize' on certain 'general' page topics, e.g. 'customer, corporate, promotion' (from state 3) and 'search, download, products' (from states 7,9,10); b. (right): learned initial state vectors $\Pi^k, k = 1, \ldots, 4$. In both subfigures, large greyvalues denote high probabilities

transition matrices were again fairly similar (one instance is shown in figure 4b). In the figure, we reshuffled the state indices such that neighbouring state-ids are 'semantically similar'. This was done in order to match the ordering in the 'semantically similar' states of the 12-states experiment. Again the state-ids were reshuffled *after* the training procedure had taken place, in order to facilitate a visual comparison between the 12-state and the 6-state transition matrices. Indeed, the transition structure is comparable to that in the previous experiment (the transitions in figure 4a appear to be a 'zoomed in' version of the transitions in figure 4b).

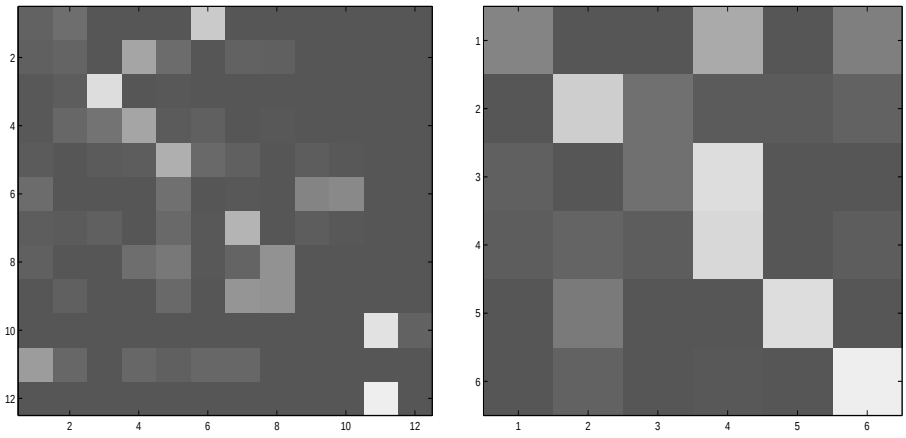

Fig. 4. a. (left): Learned 12-state transition matrix A^k; the shuffling of page-ids is such that 1-3 are the 'shop' category, 4-9 is the 'general' category and 10-12 is the 'tools' category; b. (right): learned 6-state transition matrix A^k; the shuffling of page-ids is such that 1,2 are the 'shop' category, 3,4,5 is the 'general' category and 6 is the 'tools' category

We verified our conjecture that 2 main clusters of users were present in the data by training an mHMM to half of the data from 3500 users with 2 and 3 mixture components respectively. In both cases we varied the number of states in the model. Then we computed the out-of-sample score on the remaining half of the data Y_{test} for each model $\Theta^{K,M}$ as

$$\text{score}(Y_{\text{test}}, K, M) = -\frac{\sum_i \log \sum_k \alpha_k P(Y^i_{\text{test}} | c_i = k, \Theta^{K,M}_k)}{\sum_{i,j} \text{length}(Y^{i,j}_{\text{test}})} \tag{11}$$

where $\Theta^{K,M}_k$ is the kth component of the mHMM with K components and M states. From our visual inspection of the learned categorization and user labelling we expected 2 user clusters and 3 categories to be present. However, it can be seen in figure 5 that there is no significant difference between the generalization performance of the 2 and 3 component models. Furthermore, including

more states leads to models with more predictive power. Including more than 14 states seems again to worsen the predictive power. This is an indication that it may be difficult to determine the 'meaningfulness' of the model purely on the basis of a criterion that measures the predictive quality. In [2] a similar problem was mentioned with respect to the choice of the number of components in a mixture of Markov chains. Comprising two disjunct components[4] into one may lead to models with more predictive power, but less interpretability. Note that our 'manual clustering and categorization' was based on the interpretability of the model.

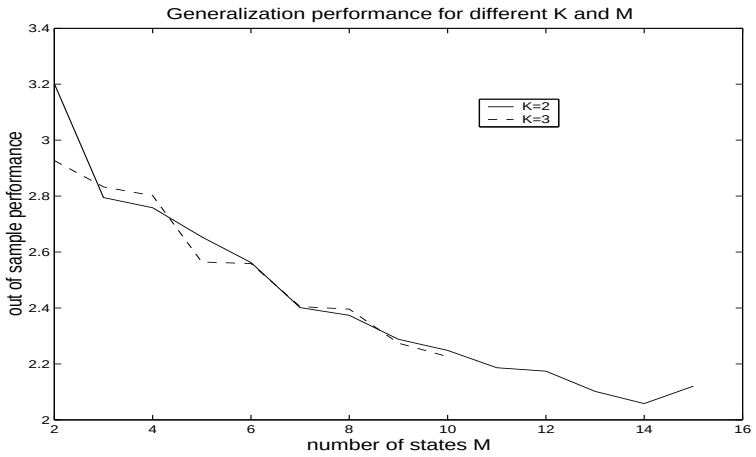

Fig. 5. Generalization performance of different mHMMs applied to web logs. There is no significant difference in performance between models with 2 or 3 mixture components. Including many states in an mHMM is preferable over including a few states. However, including more than 14 states seems to worsen generalization

4 Discussion

4.1 Discussion of results.

We noticed that two different choices for the number of categories underlying a set of clickstreams gave rise to semantically similar models for the data. This was concluded after a semantic analysis of the pages assigned to a state and by allowing that different page categories may consist of disjunct *groups* of states. In a practical setting, such a semantic analysis is not desirable. Then it becomes important to obtain a principled estimate of the number of states and the number of mixture components that is optimal for the data, e.g. by cross-validation or using

[4] E.g. comp. 1 = start in a, then go to b; comp. 2 = start in c, then go to d

model selection criteria. It is however not clear whether such a model selection that is based on predictive power will also lead to the most *interpretable* model, which is an issue of further research. The main aim of this paper, however, was to show that a meaningful page categorization may be learned simultaneously with the user labelling and inter-category transitions, making use of clickstreams and possibly static auxiliary data only.

4.2 Connection to prior work.

Several authors have addressed (parts of) the web usage mining problem considered in this paper. The use of Markov models for modelling navigational patterns has been studied in [7, 10]. In this work, each web page or page request corresponds to one state in a Markov model. In [10] this gave rise to successful predictions of HTTP-requests based on a stream of previous requests. However, the author mentioned the following limitations of the approach: lack of training data may give rise to low-quality models and cases with a large number of pages result in transition matrices that become intractable. In [7], the first-order Markov assumption was questioned, though a first-order model is considered more stable over a period of time than higher-order Markov models and typical clickstreams are usually short [4]. In our approach, the transitions are modelled at the level of page categories, which alleviates the intractability problem (transition matrices now scale with the number of page categories, which is expected to increase less than the number of pages). Moreover, we expect that the first-order Markov assumption will be more valid at the level of page categories[5].

Automatic clustering of user sessions and web pages were deemed the two interesting clustering tasks in the web usage domain [14]. Automatic categorization of web pages based on usage and content has been studied in [6]. After a content-based filtering of pages (e.g. by using a search engine), the connection structure between pages is used to identify hubs and authorities; the most relevant pages with respect to these criteria can then be categorized as belonging to this topic or query. Other approaches (e.g. see the overview in [11]) are based on a supervised content-based categorization of pages. In our approach we combine user clustering and page categorization in an unsupervised manner, using the link structure of the data only.

As stated before, our model is an extension of the work by [1]. If the states in our model are observed (e.g. when pages have been categorized manually) we have in fact a mixture of Markov chains, with update equations for Π, A and memberships c_{ik} that are similar to [1]. Moreover, we mention that an algorithm for training a mHMM has already been proposed in [12]. There it was stated that a mHMM is equivalent to one large 'overall' HMM with block-diagonal structure

[5] A reviewer added that many web sites are designed to promote the Markov assumption, i.e. each web page tries to motivate the user to stay at the web site and provides search tools and links to influence the user's next selection. This reviewer also expects that modelling state transitions at the level of page categories strengthens the Markov assumption

(which is due to the time-invariance of the cluster membership). We can see this by noting that $c_{ik} = P(c_i = k|Y^{i,j}, \Theta) = \sum_{\{x\} \in \mathcal{X}^k} P(X_t = x|Y^{i,j}, \Theta)$, where $\mathcal{X}^k$ denotes the states in the 'overall HMM' that correspond to cluster k. Despite the fact that memberships can be derived here from the 'overall HMM', the explicit formulation of memberships in our case offers a more intuitive way to combine clickstreams with static data.

4.3 Future extensions.

In future research, it may be useful to use the inter-page link structure of a web site as prior information about (im)possible inter-category transitions. For streamlining of a web site it may be necessary to use higher-order (hidden) Markov models. Moreover, user interests (and click behaviour) may change over time. Hence it may be interesting to analyze the user behaviour on, e.g., different days of the week. Finally, methods are needed to determine the optimal number of clusters and states for the problem at hand, though our results demonstrate that already suboptimal choices may lead to meaningful models.

5 Conclusion

We presented a method to cluster web surfers, based on their surfing patterns and additional information about the user. Since we learn the categorization of a web page along with the inter-category transition matrices and the cluster memberships, there is no need for a laborious prior labelling of pages. We demonstrated our method on artificial clickstreams and we retrieved a meaningful page categorization in an application to actual surfing patterns at a large commercial web site.

6 Acknowledgements

This work is supported by the Dutch Technology Foundation STW, project NNN.5321 "Graphical models for data mining". We thank KPN Research (current name: TNO Telecom) and KPN.com for supplying the web logs from the kpn.com web site.

References

1. I. Cadez, S. Gaffney, and P. Smyth. A general probabilistic framework for clustering individuals. Technical report, Univ. Calif., Irvine, March 2000.
2. I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Visualization of navigation patterns on a web site using model-based clustering. Technical report, Univ. Calif., Irvine, March 2000.
3. R. W. Cooley. *Web usage mining: discovery and application of interesting patterns from web data*. PhD thesis, University of Minnesota, USA, 2000.

4. B. A. Huberman, P. L T. Pirolli, J. E. Pitkow, and R. M. Lukose. Strong regularities in world wide web surfing. *Science*, (280):95–97, 1998.
5. M. I. Jordan, Z. Ghahramani, T. S. Jaakola, and L. K. Saul. An introduction to variational methods for graphical models. In *Learning in graphical models*. Kluwer Academic Publishers, 1998.
6. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. of 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.
7. M. Levene and G. Loizou. Computing the entropy of user navigation in the web. Technical report, Department of Computer Science, University College London, 1999.
8. L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, February 1989.
9. M. Ramoni, P. Sebastiani, and P. Cohen. Bayesian clustering by dynamics. *Machine learning*, pages 91 – 121, 2002.
10. R. R. Sarukkai. Link prediction and path analysis using markov chains. In *Proceedings of the Ninth International World Wide Web Conference*, Amsterdam, 2000.
11. Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.
12. P. Smyth. Clustering sequences with hidden markov models. In M.C. Mozer, M.I. Jordan, and T. Petsche, editors, *Advances in NIPS 9*, 1997.
13. P. Smyth. Probabilistic model-based clustering of multivariate and sequential data. In *Proc. of 7th Int. Workshop AI and Statistics*, pages 299–304, 1999.
14. J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2), 2000.