Contextual Models for User Interaction on the Web

Peter Haider[1⋆], Luca Chiarandini[2⋆⋆], Ulf Brefeld[3], and Alejandro Jaimes[4]

[1] Department of Computer Science, University of Potsdam, Germany
haider@cs.uni-potsdam.de
[2] Web Research Group, Universitat Pompeu Fabra, Barcelona, Spain
chiarluc@yahoo-inc.com
[3] Zalando GmbH, Berlin, Germany
ulf.brefeld@zalando.de
[4] Yahoo! Research, Barcelona, Spain
ajaimes@yahoo-inc.com

Abstract. Accurately modeling user sessions on the web is important because such models can be used, on one hand to predict a user's actions, and on the other hand to inform design and content decisions. This includes predicting what links a user will click on, deciding where webpage components should be placed, and what content to provide. Often it is either undesirable or not possible to build personalized models, and even when available, such models suffer from the cold start problem, or are unable to deal with context-dependent variations in user behavior. In this paper, we present a probabilistic framework for session modeling that creates clusters of similar sessions and uses contextual session information (time, referrer domain, link locations). Sessions are probabilistically assigned to the clusters by conditioning on the context. The framework addresses wide variations in user behavior that are due to context by explicitly incorporating it in the model, while specifically leveraging periodicity (weekly and daily behavioral regularities). We evaluate the framework on a set of logs from Yahoo! News.

Keywords: user modeling, context dependency, graphical models

1 Introduction

Modeling user behavior has become critical on the web, but particularly for large-scale web sites that openly offer content or services without requiring user registration. Such websites often rely on repeated user visits, so their success depends highly on how well they are able to anticipate a user's information needs by providing the right content, at the right time, in the right places. Yet, it is not unusual for the "owners" and editors of these sites to rely on simple click-through rate heuristics to make important decisions that clearly impact whether visitors to the site return or not. In the particular case of news, this includes deciding the different layouts of news sections (e.g., should the business section display a link to a technology article on the *top part* of an article page or on the *right or left* panel?), the links to include (e.g., should the sports section have a link to *entertainment*?), and the type of content to promote.

⋆ This work was performed during an internship at Yahoo! Research, Barcelona, Spain.
⋆⋆ Also Yahoo! Research, Barcelona, Spain.

Such decisions, however, are often complex because all of the variables that determine the look and feel of a page and the content provided, must also take into account user behavioral patterns which often depend on context. News consumption patterns differ depending on how the user arrives at the site, whether by clicking on links shared through social media, e-mail, or through comments on the news sites themselves (see [1]). In addition, users search for news, subscribe to RSS feeds, and visit news pages directly. Added to this is the fact that users don't consume news the same way at different times of the day or different days of the week. Given this complexity, there are important needs for news content providers in at least two areas: (1) gaining insights into how users behave when they visit the site depending on the context; (2) using models that can be leveraged to predict behavior and automatically link content or set layout parameters.

In this paper, we address these two areas. In particular, we present a probabilistic framework for session modeling that creates clusters of similar sessions, and uses contextual session information (time, referrer domain, link locations, page categories) to assign a session probabilistically to multiple clusters. We use a generative probabilistic model whose core is formed by a Markov process to capture the sequential nature of augmented sessions, and which naturally extends to a clustering of the data that can be computed by means of a nested Expectation Maximization algorithm. Moreover, the fully probabilistic nature of our approach allows us to turn the model into a predictor by marginalizing out latent variables and conditioning on the desired input observables. Exploiting the flexibility of the inference machinery allows us for instance to compute predictions for the next category, for the location of the next click, or for identifying keywords in link texts given a category, respectively. Visualizing the posterior estimates of the respective parameters provides insights on where to place links and which words to use for the anchor texts.

Our main technical contribution is the extension of Markov process-based clustering models to dynamically include context. We develop a nested mixture model for distributions over session timestamps that is able to capture periodic behavior and derive a nested EM-algorithm that *simultaneously* infers the mixture weights of the time distribution and the cluster parameters for the distributions over categories and other context. Our framework does not limit the type or number of context variables, but we validate our approach using timestamps, referrers, and click metadata as contextual variables.

We empirically evaluate our approach using a large data sample from Yahoo! News and observe that the session-based clustering model outperforms usage-based and personalized models by a large margin. We provide exemplary interpretations of the produced clusters along various dimensions and discuss their impact on user understanding.

The rest of this paper is organized as follows. Section 2 reviews related work, and Section 3 presents our framework. In Section 4 we report on empirical results and provide a discussion of our findings. Section 5 concludes.

2 Related Work

In general, techniques to model web user navigation patterns usually operate on either a per-session or a per-user basis, and usually the deployed models are intertwined with

clustering techniques to identify and group similar users or navigation patterns. Some proposed approaches are based on Markov processes [2, 3], hidden Markov models [4, 5], or relational hidden Markov models [6]. Others focus on user intent [7], behavior [8, 9, 3, 4], implicit feedback [10, 11], or modeling usability and interaction [12–14]. Some work has focused on visualizing [2], discovering [15], and in gaining insights from navigation patterns [16], while some research has focused on modeling the behavior of users pursuing specific known information seeking tasks [17, 14]. Finally, several techniques have been developed in the context of news [18, 12, 19]. Although our framework can be applied for behavioral analysis, visualization, and for gaining insights on user behavior, it is in general closest to approaches based on clustering. Therefore, we discuss those in further detail.

Often, approaches focus on deriving user-based models and estimating personalized stochastic processes from historic user data (e.g, [9, 15], Markov processes such as those mentioned above, and sequence alignment-based methods [20]). Other methods include relational models [6], association rule mining [12, 13], and higher-order Markov models [5]. Billsus and Pazzani [18] model short-term changes in the behavior of users using a hybrid user model composed of two parts: a short-term component based on k-nearest-neighbor, aimed at understanding user interest in stories similar to the ones she has already read, and a Naive Bayes classifier that builds a model of the user based on the words and features that guide her interests. Hoebel and Zicari [9], on the other hand, cluster website-visitors using a combination of hierarchical clustering with a heuristic centroid-based criterion, aiming at discovering groups of users with similar interests in several topics, while Gündüz and Özsu [21] define a similarity measure among navigation sessions and cluster them using a graph-based approach. For every cluster, a click-stream-tree is constructed and used for recommendation.

Hassan and Karim [8] evaluate the impact of clusterings on the performance of predicting pageviews. Using a heuristic-based clustering method instead of a model-based one, they arrive at the conclusion that multiple clusters do not benefit accuracy. Other researchers studied methods to evaluate the quality of clustered user model and model-based recommendation. Li et al. [19] investigate offline evaluation of contextual-bandit-based news article recommendation algorithms. Pallis et al. [16] develop a statistical test to measure the difference between clusters, obtained by clustering according to Markov process parameters, which is then also used to visualize the model. In this way, clusterings can be validated, however without regard to the behavior's context. In contrast to the results of Hassan and Karim [8], the fully probabilistic model we present in this paper proves to be able to take significant advantage of multiple clusters.

Our work differs from previous model-based clustering approaches [6, 2–4] that rely solely on the order in which web pages are requested. Our model extends Markov process-based clustering models by dynamically including context, and explicitly captures periodic behavior by using a time distribution that is a mixture of periodic Gaussians.

3 Contextual Models for User Interaction and Navigation

We define a session as a sequence of click and pageview events $\boldsymbol{e} = v_1, s_1, v_2, s_2, \ldots, v_M$. While clicks realize transitions between web pages, pageviews encode intermediate events such as displaying an article or a picture. More specifically, a session x of length M is formalized as a 5-tuple $x = (t, r, \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{w})$, where t is the timestamp of the session, r is the referrer domain, $\boldsymbol{v} = v_1, \ldots, v_M$ and $\boldsymbol{s} = s_1, \ldots, s_{M-1}$ are sequences of pageview categories and click locations, and $\boldsymbol{w} = w_1, \ldots, w_{M-1}$ are the clicked anchor texts in bag-of-words representation, respectively. Since we only consider navigation clicks within the website, there is no click s_M associated to the last pageview v_M. In addition:

- The location of a clicked link is s, which for simplicity is a discrete identifier that encodes either the clicked component (e.g., widget, module, etc) or area (e.g., North, NorthWest). Other representations, such as relative/absolute (x, y) coordinates could also be used with appropriate distributions.
- The link anchor text is represented by a bag-of-words w. If no anchor text is associated with the link, then $w_i = \emptyset$.
- Every pageview has a category $v_m \in \mathcal{C}$ where $\mathcal{C}$ contains a finite set of categories.

3.1 A Generative Session Model

The basic idea behind our model is as follows. In the first step, a cluster k is drawn according to a multinomial distribution parameterized by π. Then the session is drawn according to the parameterrs θ_k of the selected cluster by drawing timestamp t, referrer r, and the first pageview v_1 and using the Markov process to generate subsequent clicks with pageview v_j location s_j and word distribution w_j until the $exit$ state is reached which terminates the generation process. The probability of a session x can be factorized as follows:

$$P(x|\theta_k) = P(t|\beta_k)P(r|\rho_k)P(\boldsymbol{v}|r, \tau_k)P(\boldsymbol{s}|\boldsymbol{v}, \sigma_k)P(\boldsymbol{w}|\mu_k),$$

where $\theta_k = \{\beta_k, \rho_k, \tau_k, \sigma_k, \mu_k\}$ denotes the set of parameters of the k-th component so that

$$P(x|\Theta) = \sum_{k=1}^{K} \pi_k P(x|\theta_k)$$

with $\Theta = \{(\theta_k, \pi_k)\}_{k=1}^{K}$ denotes the complete generative model.

Figure 1 shows plate models visualizing the generative process. Observed variables are shaded while unshaded nodes correspond to model parameters; arrows denote dependencies and boxes indicate repetitive draws. The node labeled $\boldsymbol{e}$ denotes the sequence of events, whose generating model is detailed in the right hand diagram of Figure 1. The remainder of this section explains the model as well as the inference and parameter optimization processes in greater detail.

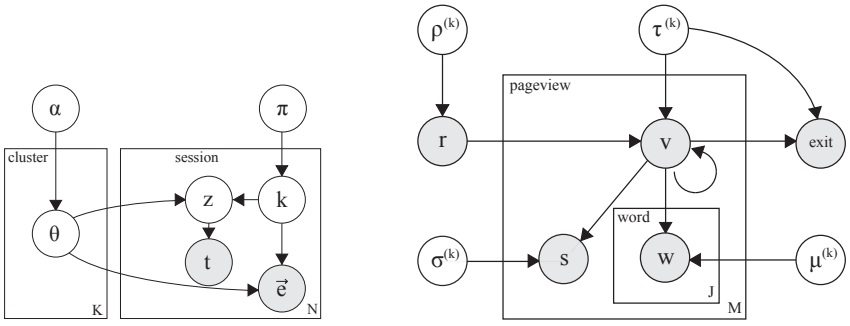

Fig. 1. Graphical model for the generative process. Shaded nodes encode observables, and unshaded nodes model parameters. The navigation sequence is subsumed in node $\boldsymbol{e}$ in the left hand diagram, and detailed in the right hand diagram.

Timestamp $P(t|\beta)$ The distribution for the timestamps is designed to capture regular behavior across days of the week: a week is modeled as a mixture model of periodic Gauss-like distributions whose peaks are repeated in one week intervals. In addition to these weekly repeating patterns, we capture regularities within workdays by including components whose peaks repeat from Monday to Friday.[5].

In general, mixtures using an infinite number of components do not scale well at large scales. Thus, we restrict our mixture to only a finite number of components that can be estimated efficiently; that is, instead of introducing components centered at every possible point within a week, we use components spaced in 10 and 30 minute intervals, respectively. We end up with 1,536 components organized in four groups:

- The first group consists of 48 working-day periodic components spaced 30 minutes apart, with a standard deviation of four hours.
- The 144 components in the second group are also periodic over the working days; their time lag is 10 minutes, and their standard deviation is one hour.
- The components of the third group are non-periodic (apart from repeating weekly) to capture patterns that differ between days of the week. We deploy 336 density functions centered in 30 minute intervals with a standard deviation of four hours.
- The fourth group contains 1,008 non-periodic components spaced in 10 minute intervals with a standard deviation of one hour.

Each element in these groups is referred to as a mixture component g_j. For every cluster k, the influence of each component is parameterized by a 1536-dimensional vector β_k with $\sum_{j=1}^{1536} \beta_{k,j} = 1$. Every session has a latent indicator variable z that selects one

[5] Note that repeating components alone does not favor periodic patterns since a repeating component is itself only a mixture of non-repeating components and does not change the space of overall mixture distributions. We therefore introduce a bias towards periodic and smooth distributions by interpolating the components with a uniform distribution to various amounts. Smoother and more periodic components are interpolated less than peaked components.

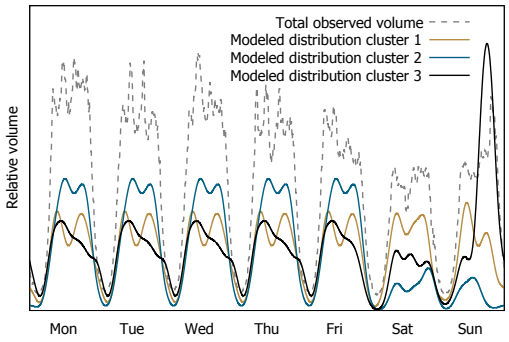

Fig. 2. Observed and modeled time distributions.

of the mixture components, such that the overall distribution over timestamps can be written as $P(t|\beta) = \sum_j P(z = j|\beta)P(t|g_j)$. Figure 2 shows an exemplary time distribution for a solution with three clusters together with the actual observed distribution in the training set.[6] The described mixture model produces smooth and periodic distributions without overfitting the data, that is without reproducing the noise of the actual observed distribution.

Referrer Domain $P(r|\rho)$ and Pageviews $P(\boldsymbol{v}|r,\tau)$ As shown in Figure 1, the referrer domain r and the pageviews $v_1, \ldots, v_M$ form a Markov chain together with a distinguished $exit$-symbol. We use a first-order Markov assumption which reflects the intuition that clicks only depend on the viewed page and are thus independent of previous page views and/or clicks. The resulting Markov process consists of two components, a multinomial distribution parameterized by a vector ρ over the set of all referrer domains $P(r|\rho)$ and transition probability parameters τ for the sequence of pageviews. The latter decomposes into the matrices $\tau = \{\tau^0, \tau^+\}$ where τ^0 specifies the distribution of the topic of the first pageview given the referrer, and τ^+ specifies the probability of transitioning between the topic v_m and topic v_{m+1} or the end of the session, respectively. Hence, the probability of the Markov chain is given by $P(r, \boldsymbol{v}|\rho, \tau) = P(r|\rho)P(v_1, \ldots, v_n|r, \tau)$, where $P(r|\rho) = \rho_r$ and matrices τ^0 and τ^+ such that

$$
\begin{aligned}
P(\boldsymbol{v}|r,\tau) &= P(v_1|r,\tau^0)\left[\prod_{m=1}^{M-1} P(v_{m+1}|v_m,\tau^+)\right] P(exit|v_M,\tau^+) \\
&= \tau^0_{r,v_1}\left[\prod_{m=1}^{M-1} \tau^+_{v_{m-1},v_m}\right]\tau^+_{v_M,exit}.
\end{aligned}
$$

Anchor Texts $P(w|\mu)$ and Location of Clicks $P(s|v,\sigma)$ The distribution of the anchor texts of the clicked links could give insights into the static information needs of

[6] The data is described in Section 4.

the users. The words of the anchor texts are drawn from multinomial distributions over a dictionary with cluster-specific parameter vector μ. Similarly, the location of the clicked links is also modeled by a multinomial distribution which is however conditioned on the category of the following pageview. The latter multinomial is governed by parameter matrix σ. Using the independence of link text and location leads to $P(\boldsymbol{w}, \boldsymbol{s}|\boldsymbol{v}, \sigma, \mu) = P(\boldsymbol{w}|\mu)P(\boldsymbol{s}|\boldsymbol{v}, \sigma)$ with

$$P(\boldsymbol{w}|\mu) = \prod_{m=1}^{M-1} P(w_m|\mu) = \prod_{m=1}^{M-1} \prod_{i=1}^{|w_m|} \mu_{w_{m,i}}$$

where $|w_m|$ denotes the number of anchor text words of the m-th clicked link, and

$$P(\boldsymbol{s}|\boldsymbol{v}, \sigma) = \prod_{m=1}^{M-1} P(s_m|\sigma, v_{m+1}) = \prod_{m=1}^{M-1} \sigma_{v_{m+1}, s_m}.$$

3.2 Parameter Estimation

Given a set of N sessions $X = \{x_1, \ldots, x_N\}$ and the number of clusters K, the task is to estimate the parameters $\Theta = \{(\theta_k, \pi_k)\}_{k=1}^{K}$ of the generative model. We aim at finding the *maximum-a-posteriori* (MAP) solution by solving

$$\operatorname*{argmax}_{\Theta} P(\Theta|X) = \operatorname*{argmax}_{\Theta} P(\Theta) \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k P(x_i|\theta_k),$$

where $P(\Theta)$ is modeled by a symmetric Dirichlet prior with concentration factor α.

The main difficulty in the optimization is the presence of two different types of latent variables, the first is encoding the cluster memberships of the sessions k and the second encodes the distribution over time components z that generate the timestamp. Since the latter is required for inferring the former, we now present a nested Expectation Maximization strategy to optimize both simultaneously.

Let us assume for a moment that the time component indicator variables z_i were known. In that case we could use a standard Expectation-Maximization (EM) clustering algorithm [22] for the parameter estimation. The EM algorithm computes, in every E-step, estimates $\gamma_{i,k}$ of the cluster membership variables, with $\sum_k \gamma_{i,k} = 1$, which indicate the posterior probabilities of an example x_i belonging to cluster k. In the M-step, the MAP-estimates for every set of the cluster parameters θ_k are computed as follows:

$$\hat{\theta}_k = \arg\max_{\theta_k} P(\theta_k|X, y) = \arg\max_{\theta_k} \log P(\theta_k) + \sum_i \gamma_{i,k} \log P(x_i|\theta_k).$$

Due to the conjugacy of the Dirichlet prior to the multinomial distribution, the maximization simplifies to counting the occurrences of a particular component transition. For example the time distribution component weights β_k are computed as

$$\hat{\beta}_{k,\ell} = \frac{\alpha - 1 + \sum_i \gamma_{i,k} [\![z_i = \ell]\!]}{\sum_{\ell'} \alpha - 1 + \sum_i \gamma_{i,k} [\![z_i = \ell']\!]},$$

with the indicator function $[\![z_i = \ell]\!] = 1$ if $z_i = \ell$ is true and 0 otherwise. All other parameters are calculated analogously.

However, since the z_i are actually unknown, we have to marginalize over them. Thus the optimal parameter vector β for a cluster k, given the current cluster membership estimates γ, is optimized using

$$\hat{\beta} = \arg\max_{\beta} (\alpha - 1) \sum_j \log \beta_j + \sum_i \log \left[\gamma_{i,k} \sum_{\ell=1}^{1536} \beta_\ell P(t_i | g_\ell) \right],$$

under the constraint $\sum_j \beta_j = 1$ where g_ℓ denotes the generating components of the timestamp. This is a concave optimization problem under the condition that $\alpha \geq 1$, since the terms $P(t_i|g_\ell)$ are constant. Having no closed-form solution, a straightforward approach would be to solve it using gradient descent or a variant of Newton's method. However the estimates γ change in every iteration of the EM-algorithm, and thus a costly optimization would have to be performed in every iteration.

A more efficient method is to *intertwine* the optimization of β with the EM-algorithm, performing only one closed-form update of β in every M-step. We derive this update analogously to the M-step update for the cluster prior π (cf. [23]), by introducing additional variables $\zeta_{k,i,\ell}$ with $\sum_l \zeta_{k,i,\ell} = 1$ which encode our posterior belief that the timestamp of session x_i is generated by component ℓ, conditioned on x_i belonging to cluster k. These can be computed in the E-step as

$$\zeta_{k,i,\ell} = \frac{\beta_{k,\ell} P(t_i|g_\ell)}{\sum_{\ell'} \beta_{k,\ell'} P(t_i|g_{\ell'})}. \tag{1}$$

Using these estimates, we can compute the component weights of each cluster in the M-step as

$$\hat{\beta}_{k,\ell} = \frac{\alpha - 1 + \sum_i \gamma_{i,k} \zeta_{k,i,\ell}}{\sum_{\ell'} \alpha - 1 + \sum_i \gamma_{i,k} \zeta_{k,i,\ell'}}.$$

This *nested* EM-algorithm is guaranteed to increase the data likelihood in every iteration until convergence to a local optimum, analogously to the standard EM-algorithm.

3.3 Inference

Our generative model $P(x|\Theta)$ can be easily turned into a prediction model by marginalizing out latent variables and conditioning on the desired input observables. Recall that, at the m-th pageview of a session, we already observed the previously visited categories $v_1, \ldots, v_m$ and the previously clicked locations $s_1, \ldots, s_{m-1}$ and link texts $w_1, \ldots, w_{m-1}$, as well as the session's timestamp t and referrer domain r.

For instance, we can predict the category of the next pageview a user will navigate to by conditioning on the context and history while marginalizing over the latent cluster variable. Conditioning again on this prediction, we can furthermore predict which location within the page she will click on next. Let $\boldsymbol{e}_{[m]}$ denote the events of a session up to the m-th pageview, that is $\boldsymbol{e}_{[m]} = \{(v_1, \ldots, v_m), (s_1, \ldots, s_{m-1}), (w_1, \ldots, w_{m-1})\}$,

then the predictive distribution for the next category (including the end of the session) is given by $P(v_{m+1}|\boldsymbol{e}_{[m]}, t, r)$ and can be computed by marginalizing over the cluster variables,

$$P(v_{m+1}|\boldsymbol{e}_{[m]}, t, r) \propto \sum_k P(v_{m+1}|v_m, \theta_k) P(\boldsymbol{e}_{[m]}, t, r|\theta_k) P(k). \tag{2}$$

However, our model also contains traditional models as special cases that are solely based on the observed sequence of categories [2] by an additional marginalization over the context variables,

$$\begin{aligned} P(v_{m+1}|v_1, \ldots, v_m) &\propto \sum_k \sum_{\boldsymbol{s},\boldsymbol{w},t,r} P(v_{m+1}, k|\boldsymbol{e}_{[m]}, t, r) \\ &\propto \sum_k P(v_{m+1}|v_m, \theta_k) P(v_1, \ldots, v_m|\theta_k) P(k). \end{aligned} \tag{3}$$

Comparing Equations (2) and (3) shows that the context variables provide additional information on how to weight the influences of the different clusters. In the following section, we evaluate the context variables in terms of their contribution to the predictive performance.

Our model can contribute to optimize the layout of web pages by providing insights on where to place links and likely-clicked word distributions. We therefore infer the location of the next click by conditioning on the linked category

$$P(s_m|v_{m+1}, \boldsymbol{e}_{[m]}, t, r) \propto \sum_k P(s_m|v_{m+1}, \theta^{(k)}) P(\boldsymbol{e}_{[m]}, t, r, v_{m+1}|\theta^{(k)}) P(k).$$

The predictive distribution for clicking on a link with anchor text w, $P(w|\boldsymbol{e}_{[m]}, t, r)$, can be computed similarly and is proportional to $\sum_k P(w|\theta^{(k)}) P(\boldsymbol{e}_{[m]}, t, r|\theta^{(k)}) P(k)$.

3.4 Incremental and Distributed Parameter Estimation

For practical applications, the batch style of the nested EM-algorithm hinders deployment because every retraining needs to be performed on all data. In this section, we briefly sketch the parameter estimation in realtime using incremental updates, similar to the algorithm proposed in [24].

Once the clusters are determined by running the nested EM-algorithm until convergence, new sessions can be incorporated by performing a single partial iteration. For every new session, we have to compute the estimates γ, then update the counters of all components and normalize the cluster parameters using $\mathcal{O}(K)$ operations. Let $count_T(\cdot)$ denote the counts of all weighted entity occurrences after having processed T examples, e.g. $count_T(\rho^{(k)}, \ell) = \alpha - 1 + \sum_{i=1}^{T} \gamma_{i,k} [\![r_i = \ell]\!]$ and $count_T(\rho^{(k)}) = \sum_\ell count_T(\rho^{(k)}, \ell)$. Then a new example x_* can be incorporated into the model by first estimating its cluster membership using the current model parameters $\tilde{\Theta}, \tilde{\pi}$ as

$$\gamma_{*,k} = \frac{\tilde{\pi}_k P(x_*|\tilde{\theta}^{(k)})}{\sum_{k'} \tilde{\pi}_k P(x_*|\tilde{\theta}^{(k')})}.$$

The counts are updated according to $count_{T+1}(\rho^{(k)}, \ell) = count_T(\rho^{(k)}, \ell) + \gamma_{*,k}[\![r_* = \ell]\!]$ and $count_{T+1}(\rho^{(k)}) = count_T(\rho^{(k)}) + \gamma_{*,k}$. The new MAP-parameters are

$$\hat{\rho}_\ell^{(k)} = \frac{count_{T+1}(\rho^{(k)}, \ell)}{count_{T+1}(\rho^{(k)})}.$$

The remaining parameters, π, β, τ, σ, and μ, are updated analogously.

That way, an up-to-date, approximate model can be maintained efficiently and full retraining is only necessary occasionally. The benefit of an online variant is that novel topics can be taken into account and recommended to users faster. Our model already has an advantage over user-centric, personalized models, because every user benefits from the information gained about sessions in the cluster she is currently in. Having an always up-to-date model entails that estimates for the click probability of a new topic are available as soon as a few peers have clicked on it.

Furthermore, the training of our model can easily be distributed on several machines using the MapReduce framework. EM-like algorithms process training instances one after another and store tables with counts for every instance in the E-step. The counting can be performed on several machines in parallel during the map-phase, independently for every training example, generating cluster membership estimates and fractional counters. In the reduce phase, the fractional counters are aggregated, and finally the M-step is performed, namely computing the MAP-parameters from the total counts. After the M-step, the current model is distributed to all machines for the next iteration of mapping and reducing respectively expectation and maximization (cf. [25]). The distributed computation schema can in principle also be applied to the online variant, for processing multiple new examples in parallel.

4 Empirical Evaluation

In this section we evaluate the generative model under several aspects using a large data sample from Yahoo! News United Kingdom. We use a sample of data from June and July 2011 and use the former month for parameter estimation and the latter for evaluation. Users are disambiguated according to their browser cookies and user sessions are split after 25 minutes of inactivity[7].

The next section reports on the predictive performance of the probabilistic model and compares the outcomes with appropriate baseline methods. Section 4.2 addresses insights gained by applying our model to the news domain and discusses the findings in terms of user understanding.

4.1 Predictive Performance

We measure predictive performance in terms of the predicted log-likelihood of the next pageview and the location of the next click conditioned on the session's history and context. That is, we average $\log P(s_m, v_{m+1} | \boldsymbol{e}_{[m]}, t, r)$ over all events of

[7] All processing is anonymous and aggregated

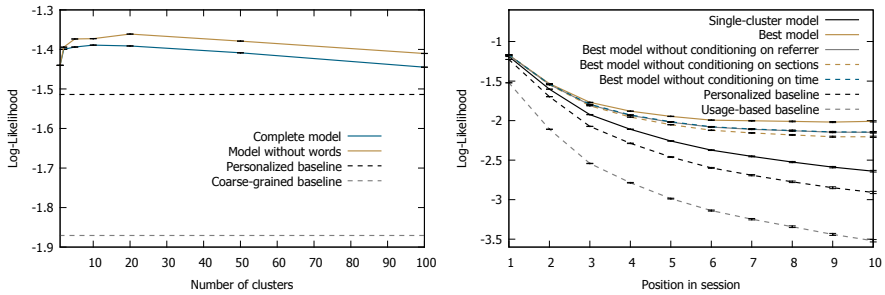

Fig. 3. Prediction performance and standard error depending on number of clusters (left) and length of session history (right)

all test sessions. The higher the session log-likelihood of a model, the better it reflects the characteristics of the data. This is a more natural evaluation measure than for instance measuring the accuracy of the most probable pageview and location, (e.g., $\arg\max_{s_m, v_{m+1}} P(s_m, v_{m+1} | \boldsymbol{e}_{[m]}, t, r)$), since there are no negative examples. Note that if a user clicks on a link ℓ, it does not mean that she is not interested in other articles but that she is at that point *more* interested in ℓ.

Our evaluation comprises several aspects of the probabilistic model. In Section 4.1, we compare the accuracy of the next click with appropriate baseline methods, and Section 4.1 evaluates the impact of the context by marginalizing over the respective variables. The following section introduces the baseline methods.

Baselines We compare our model to two user-centric baselines. Instead of using the nested EM-algorithm, the two baselines use a fixed assignment of user sessions to clusters. They are formally defined as follows.

The *usage-based* baseline simply groups the users into three groups according to their number of pageviews in June. We define the group sizes so that they reflect heuristics used in commercial systems to provide a basic level of personalization and/or monetization. The first group contains *tourists* who rarely visit the site, the second group covers *regular users*, and the third group contains the *power users*. We estimate a probabilistic model for every group.

The *personalized* baseline reflects a personalized approach and estimates a single probabilistic model for every user by assigning her respective sessions in June to a cluster. However, initial experiments showed that the data is too sparse for users with only a few pageviews. We thus split users according to their usage in two groups using a threshold η. For users whose pageviews exceed η in June, a personalized model is estimated as described while users who generate fewer pageviews than η are grouped in a single cluster. If users cannot be disambiguated and uniquely assigned to a cluster in the evaluation data from July we also resort to the model that is estimated on the shared cluster. The trade-off η is adjusted by model selection where the chronologically last 25% of June are used as holdout data.

Predicting Categories Using Context In this section, we evaluate the performance of predicting the next clicked category and link location, conditioned on the session history, using the model in Equation 2. We compare the baselines with the full generative model of Section 3 and a model where we omitted the words of the anchor texts. Preliminary experiments have shown that the latter improves over the full model. By contrast, marginalizing over the other context variables reduces the performance of the full model. We refer to the next section for detailed analysis of the impact of the different context variables.

Figure 3 (left) shows the average log-likelihood of the prediction for different numbers of clusters. The two baselines use a fixed clustering and are therefore independent of the number of clusters. The full probabilistic model and its counterpart without the anchor texts outperform the baselines significantly, even for only a few clusters. Additionally, the models without words consistently outperform the full model, indicating that the distribution over the bag-of-words is too noisy to contribute positively.

The predictive performance initially increases with the number of clusters and then decreases again for more than 20 clusters; generally, solutions with too many clusters tend to overfit the data. In the remaining experiments we therefore focus on models with 20 clusters and always marginalize out the anchor texts of the clicked links.

Evaluating the Impact of Context We now evaluate the importance of the incorporated context. We begin with the best model obtained in the previous section that consist of 20 clusters and does not depend on the anchor texts of the links. Using this model, we selectively discard parts of the remaining context, that is the referrer, the timestamps, and the locations of the clicks, to measure their respective impact. For comparison, we include the *usage-based* and *personalized* baselines and an additional single-cluster solution that has only a single generating component and does not take context into account. Additionally, we break down prediction accuracy by the position of the clicked category and link within the session, in order to gain insight into how accumulating various amounts of context information impacts accuracy.

Figure 3 (right) shows the resulting prediction accuracies for the baselines, the best model, and various sub-models thereof. Accuracies clearly drop as sessions progress. Except for the usage-based baseline, all methods predict the first click after the first pageview equally well. As the number of pageviews increases, the performance of the approaches becomes more distinguishable. A possible explanation for the performance drop is that users are presented a variety of related news articles and may be distracted by interesting news articles while browsing the site, making prediction more difficult as a session progresses.

Interestingly, the single-cluster solution performs significantly better than the other two baselines. Apparently, the fixed clusterings are inappropriate approaches to the data and thus lead to the poor performance. The contextual models always perform significantly better than the baselines which do not take advantage of context. More importantly, instead of deteriorating as the baselines, the context saturates the performance of the contextual models which remain constant for sessions with more than 5 pageviews. Discarding context significantly drops the performance; the differences in performance clearly show the importance of the different types of context.

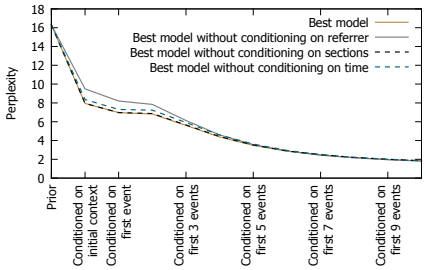

Fig. 4. Perplexity of distribution over clusters.

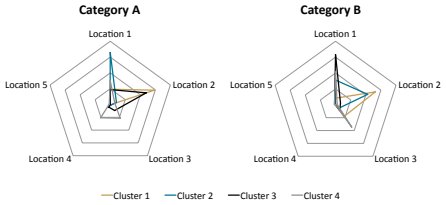

Fig. 5. Distributions over five link locations for four clusters and two exemplary categories.

4.2 Applications of Our Model

In this section we discuss the suitability of the cluster mapping, visualization, and how our model may be applied to modifying page layouts.

Mapping Users to Clusters One of the key questions is how confident the cluster assignments obtained from the probabilistic model are. We measure confidence using the information theoretic measure *perplexity*. In our case, the maximum possible perplexity value for 20 clusters is 20, which indicates a uniform distribution over the 20 clusters, while a perplexity of 1 implies a point distribution for a single cluster.

Figure 4 shows the perplexity of the distribution over the 20 clusters, conditioned on different sets of variables. The leftmost point denotes the *a priori* perplexity that is solely based on the priors π_k of the clusters. The second point is the perplexity of the distribution conditioned on the initial context given by the referrer domain and the timestamp. The figure shows that context significantly reduces the uncertainty by about 50%. Every click of the user further reduces the perplexity which drops rapidly until it reaches 2 which corresponds to the same uncertainty as that of a coin flip.

The figure shows that we obtain significant reductions in perplexity and therefore higher confidence about the cluster membership by conditioning the model on context.

Time-based Visualizations Previous work on clustering user sessions (e.g., [2]), focuses on visualizing the resulting clusters only in terms of the sequences of visited categories. By contrast, one of the main advantages of using dynamic contextual models is that the resulting clusters can be interpreted along the context dimensions. The corresponding visualizations thus highlight context-specific aspects of the data and allow for meaningful projections. For instance, Figure 6 shows the observed category distribution for the four clusters with the highest prior probabilities, projected on the days of the week according to the timestamps of the contained sessions. The rows depict a model with four clusters (left column), the already discussed solution with 20 clusters (middle), and a large model with 50 clusters (right).

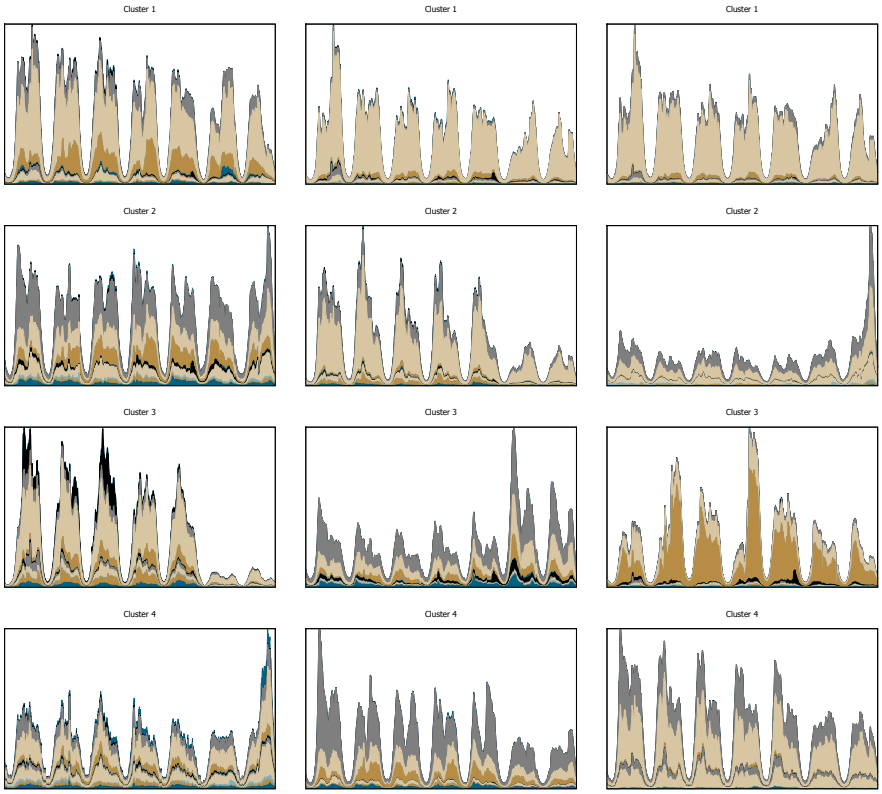

Fig. 6. Distribution of categories over time in the largest 4 clusters for models with 4 (left column), 20 (middle column), and 50 clusters (right column).

The figure shows strong correlations between the relative volume of the categories and time. Some clusters are specialized on reoccurring patterns for business days while others focus on capturing weekends. The respective clusters also possess different topic distributions, indicating that one captures work-related browsing sessions while others cover more recreationally-oriented information needs. Naturally, solutions with more clusters tend to cover fewer categories.

Compared to traditional approaches the contextualization leads to intuitive and interpretable results. Without context, the range of visualization possibilities is limited and more or less restricted to displaying transition matrices or cluster distributions.

Improving Web Page Layout & Content Our model can be used to improve webpage layout (e.g., where to place "modules", sections, or links), and content (e.g., words to use for link anchor texts). For example, Figure 5 shows the five most frequent locations for two categories A and B. The colored lines correspond to the four clusters and show the probability of a click on one of the locations given the category. The visualization

shows that some locations, such as four and five, play only a minor role in the layout and are rarely clicked on. By contrast, locations one and two are received a high number of clicks and exhibit interesting behavior. Sessions in the blue cluster interested in category A mainly use location one, while members of the black cluster focus on location two for performing the same action. Vice versa, location two is preferred by the blue cluster for category B, while the black cluster uses "prefers" location one. Once detected, this behavior can be exploited by cluster-dependent layouts of the page to guide the user through the site, and link locations that are ignored by groups of users could be dynamically replaced by more appropriate pointers.

Discussion The dynamic nature of user behavior in news consumption along with the complexities of the news cycle makes modeling and prediction extremely difficult. Our framework is able to consider context dynamically, and can be applied for prediction, as well as to obtain insights that could be used to make decisions on content and layout. On one hand, the interpretability of the clusters can provide significant insights (e.g., a content provider examining Figure 6 could easily determine the most suitable content categories for weekdays vs. weekends), and on the other hand, its prediction capabilities could be used to automatically adjust content locations and links.

5 Conclusion

We presented a generative model for user navigation on the Web. Our approach models sessions as sequences of contextualized pageviews where context is incorporated in terms of timestamps, click metadata, and referrer domains. The model naturally leads to a clustering of the sessions that can be projected on context variables for interpretable visualizations. Empirically we showed, on a large sample from Yahoo! News, that our probabilistic approach is more accurate than baseline models. We exploited several features and discussed applications of our model in adjusting content locations and links.

Acknowledgements

Part of this work was supported by the German Science Foundation (DFG) under the reference number "GA 1615/1-1"

References

1. Purcell, K., Rainie, L., Mitchell, A., Rosenstiel, T., Olmstead, K.: Understanding the participatory news consumer. Website (June 2010)
2. Cadez, I., Heckerman, D., Meek, C., Smyth, P., White, S.: Visualization of navigation patterns on a web site using model-based clustering. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2000)
3. Manavoglu, E., Pavlov, D., Giles, C.L.: Probabilistic user behavior models. In: Proceedings of the Third IEEE International Conference on Data Mining. (2003)
4. Ypma, A., Heskes, T.: Automatic categorization of web pages and user clustering with mixtures of hidden markov models. In: WEBKDD 2002 - Mining Web Data for Discovering Usage Patterns and Profiles. Springer (2003) 35–49

5. Deshpande, M., Karypis, G.: Selective markov models for predicting web page accesses. ACM Transactions on Internet Technology **4(2)** (2004) 163–184
6. Anderson, C.R., Domingos, P., Weld, D.S.: Relational markov models and their application to adaptive web navigation. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2002)
7. Giannopoulos, G., Brefeld, U., Dalamagas, T., Sellis, T.: Ranking models for user intent. In: Proceedings of the ACM Conference on Information and Knowledge Management. (2011)
8. Hassan, M.T., Karim, A.: Impact of behavior clustering on web surfer behavior prediction. Journal of Information Science and Engineering **27** (2011) 1855–1870
9. Hoebel, N., Zicari, R.: On clustering visitors of a web site by behavior and interests. In: Advances in Intelligent Web Mastering. Volume 43. Springer Berlin / Heidelberg (2007) 160–167
10. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Gay, G.: Accurately interpreting clickthrough data as implicit feedback. In: Proceedings of the Annual International ACM SIGIR Conference. (2005)
11. Kelly, D., Teevan, J.: Implicit feedback for inferring user preference: A bibliography. SIGIR Forum **37(2)** (2003) 18–28
12. Daş, R., Türkoğlu, İ.: Creating meaningful data from web logs for improving the impressiveness of a website by using path analysis method. Expert Systems with Applications **36**(3) (2009) 6635–6644
13. Daş, R., Türkoğlu, İ.: Extraction of interesting patterns through association rule mining for improvement of website usability. Istanbul University-Journal of Electrical & Electronics Engineering **9**(18) (2010)
14. Wilson, M., et al.: Evaluating advanced search interfaces using established information-seeking models. Journal of the American Society for Information Science and Technology **60**(7) (2009) 1407–1422
15. Chevalier, K., Bothorel, C., Corruble, V.: Discovering rich navigation patterns on a web site. In: Proceedings of Discovery Science. (2003)
16. Pallis, G., Angelis, L., Vakali, A.: Validation and interpretation of web users' sessions clusters. Information Processing & Management **43(5)** (2007) 1348–1367
17. Fu, W., Pirolli, P.: SNIF-ACT: A cognitive model of user navigation on the World Wide Web. Human-Computer Interaction **22**(4) (2007) 355–412
18. Billsus, D., Pazzani, M.: User modeling for adaptive news access. User Modeling and User-Adapted Interaction **10(2)** (2000) 147–180
19. Li, L., Chu, W., Langford, J., Wang, X.: Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In: Proceedings of the fourth ACM international conference on Web search and data mining, ACM (2011) 297–306
20. Hay, B., Wets, G., Vanhoof, K.: Mining navigation patterns using a sequence alignment method. Knowledge and Information Systems **6** (2004) 150–163
21. Gündüz, Ş., Özsu, M.T.: A web page prediction model based on click-stream tree representation of user behavior. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge Discovery and Data Mining. (2003)
22. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological) (1977) 1–38
23. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
24. Neal, R., Hinton, G.: A view of the EM algorithm that justifies incremental, sparse, and other variants. Learning in graphical models **89** (1998) 355–368
25. Das, A., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: Proceedings of the 16th international conference on World Wide Web, ACM (2007) 271–280