Discovering Temporal Hidden Contexts in Web Sessions for User Trail Prediction

Julia Kiseleva, Hoang Thanh Lam,
Mykola Pechenizkiy
Department of Computer Science
Eindhoven University of Technology
the Netherlands
{j.kiseleva,t.l.hoang,m.pechenizkiy}@tue.nl

Toon Calders
Computer & Decision Engineering
Université Libre de Bruxelles
Belguim
Toon.Calders@ulb.ac.be

ABSTRACT

In many web information systems such as e-shops and information portals, predictive modeling is used to understand user's intentions based on their browsing behaviour. User behavior is inherently sensitive to various hidden contexts. It has been shown in different experimental studies that exploitation of contextual information can help in improving prediction performance significantly. It is reasonable to assume that users may change their intents during one web session and that changes are influenced by some external factors such as switch in temporal context e.g. 'users want to find information about a specific product' and after a while 'they want to buy this product'. A web session can be represented as a sequence of user's actions where actions are ordered by time. The generation of a web session might be influenced by several *hidden temporal* contexts. Each session can be represented as a concatenation of independent segments, each of which is influenced by one corresponding context. We show how to learn how to apply different predictive models for each segment in this work. We define the problem of discovering *temporal hidden* contexts in such way that we optimize directly the accuracy of predictive models (e.g. users' trails prediction) during the process of context acquisition. Our empirical study on a real dataset demonstrates the effectiveness of our method.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; H.3.5 [**Information Storage and Retrieval**]: On-line Information Services - Web-based services

General Terms

Algorithms, Measurement, Experimentation

Keywords

Temporal Web analytics, Context-awareness, Browsing behaviour

1. INTRODUCTION

Exploitation of contextual information for predictive modeling has been emerging as an important research topic hav-

ing an exposure in computational advertising [6, 2], recommender systems [13, 1, 14], web search and other related areas [17]. In many web applications, explicit contextual information is available along with data. However, hidden contexts can be latent in data and discovering them is a challenging problem. It has been shown that contextual information is useful for predictive analytics such as user's trail prediction [4]. For example, in a weblog of users' activities on a website, the contextual information such as the user's (current) location, used device or gender can split users into subgroups sharing similar backgrounds. Users in the same group usually behave in a similar way. Therefore, their intentions are easier to recognize when the predictive models cleverly leverage the available contextual information.

There are several important groups of research questions in context-aware predictive analytics related to hidden context discovery, context management and context integration into predictive modeling. We focus on one of the most general questions from the first group: *"How to discover a set of useful hidden contexts from a web session?"* In particular, we want to show that temporal hidden contexts help to improve users' trail predictions.

Users usually perform actions in a sequential manner on a website and the set of all actions are given. The goal is to predict the next action each user will perform given historical data about user activities. For instance, in Figure 1, we present a web session as a sequence of user's actions (a, b, f, d, c, a, a) (from the alphabet $\{a, b, c, d, e, f\}$). We assume that under one context users perform a specific set of actions and when the context is switched to another one, another set of actions are performed. That means a context is defined as an external factor which is associated with a specific set of user actions. For instance, in a web session, there are sets of actions associated with the context like "search" while there are other sets of actions associated with the context like "buying".

In the general case, sets of actions belonging to different contexts can be intersect but in this work we only consider a simplified case when all contexts are disjoint. In that specific case, a web session will be split into segments each contains only actions belong to a context. If contexts are discovered in an appropriate way, actions in a segment are homogeneous, thus, it is easier to be predicted. Given the set of all possible actions, this work aims at splitting the set into k disjoint subsets each corresponding to a hidden context such that the local prediction models built for each context make prediction better than the global model built for the whole
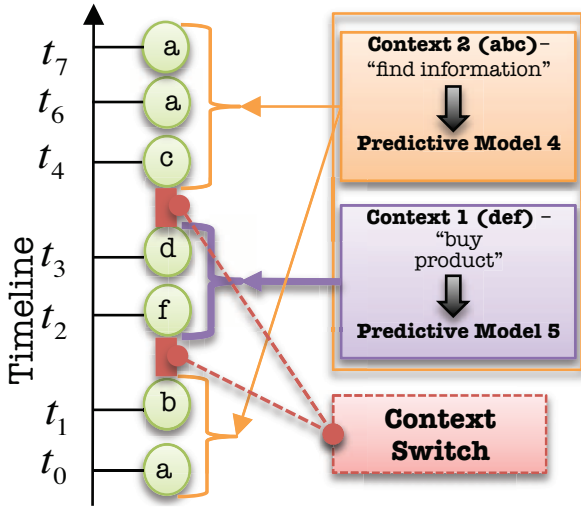
Figure 1: An example of temporal hidden contexts that are reoccurring in user's trail (a, b, f, d, c, a, a)

data. We formulate the context discovery process as an optimization problem. Subsequently, we propose a hierarchical clustering algorithm to solve the problem. The experimental results with a real-word use-case show that our approach is able to discover temporal contexts which allow to improve the prediction performance significantly.

The set of discovered contexts not only helps improve prediction performance but also provides a useful summary about users' behaviour even when domain experts are not available and data exploration gives only partial knowledge about the data. For example, context C_{search} describes a situation when user searches or $C_{recommend}$ depicts a user behaviour when user follows site recommendation. A good user behaviour summary allows us to make an aware decision on what basic hypothesis to create and how to mine the data.

In summary, our major contributions are:

- We formulate the problem of temporal context discovery as an optimisation problem.

- A hierarchical clustering method is proposed to determine the optimal number of hidden contexts and mine temporal contexts.

- We show a real-world use case in which the contexts as we defined them do exist and are useful for prediction.

The roadmap of this paper is as follows. In Section 2 we introduce notation. In Section 3 we formalise the problem of temporal hidden context discovery. We report our experimental results in Section 4. In Section 5 we discuss related work. Finally we conclude our study in Section 6.

2. PRELIMINARIES

Data representation. The general representation of users' historical behavior is given as a log with web sessions: $\mathfrak{D} = \{s_1, s_2, \ldots, s_n\}$. Given a set $\mathfrak{A} = \{a_1, \ldots, a_m\}$ of *event types*, an *event* is a pair (A, t), where $A \in \mathfrak{A}$ is an event type and t is the occurrence time of event. The web session of the user is an ordered sequence of events:

$$s = \langle (A_1, t_1), (A_2, t_2), \ldots, (A_n, t_n) \rangle \qquad (1)$$

such that $A_i \in \mathfrak{A}$ for all $i \in [1, n]$ and $t_i \in [s_s, s_e]$, $t_i < t_{i+1}$ for all $i \in [1, n-1]$, where s_s, s_e are integers denoting the starting and ending time of the session. Note that we do not have $t_i = t_{i+1}$, i.e. several events cannot occur at the same time. Let denote $T = [s_s, s_e]$ as a segment.

In our case the users' behaviour is categorized by the types of events e.g. *searches, clicks on ads* or *homepage visits*. As a running example, we consider a website containing six possible activities with categorical labels a, b, c, d, e and f. Every user visiting the website produces a sequence of transition activities corresponding to the categories that the user has visited. In this example, $\mathfrak{D}$ is the set of possible sequences of activities from the categories a, b, c, d, e and f.

Contextual features and categories. Let $\Theta = \{C_1, \times \cdots C_i \times \cdots \times C_N\}$ be the space of all possible contextual features associated with each segments T of the data instance s (Equation 1). Where each C_i is a contextual feature and θ_s is a vector of the contextual features associated with sequence s at time t_i.

Let $M : \Theta \times \mathfrak{D} \times T \mapsto V$ be a predictive model that maps each test sequence $s \in \mathfrak{D}$ associated with the contextual information θ_s at time $t_i \in T$ to the decision space V. Let $F(s, M(\theta_s, s)) : \mathfrak{D} \times V \mapsto \mathbb{R}$ be the function evaluates how good a model is. For example, in the case which predicts the next activity which the user will perform, the decision space V is the same as the data instance space, i.e. $V \equiv \mathfrak{D}$. An example of the evaluation function is the number of true predictions made by M over the test instance s. For instance, assume that the model M predicts $s = ababc$ as $M(\theta_s, s) = \underline{ab}e\underline{dc}$ then it makes three true predictions corresponding to the underlined activities, i.e. $F(s, M(\theta_s, s)) = 3$.

Given a class of predictive model $\mathfrak{M}$, we are looking for h small models from $\mathfrak{M}$ such that a user's behaviour is explained better by single model compared to global model. Namely the set of individual models give better evaluation then global model. Let $M = \{M_1, M_2, \ldots, M_h\}$ be a set of individual learning procedures from class $\mathfrak{M}$.

Let C be a context with n categories: $Cat = \{Cat_1, Cat_2, \ldots, Cat_n\}$ associated with segment of data instance $s \in \mathfrak{D}$ at time t_i. A function $\mathcal{G}$ maps contextual feature to categories, e.g. the contextual feature 'a user mood' can be mapped into three different categories such as $\{good, neutral, bad\}$. Each contextual category is mapped to a learning model by function $\mathcal{H}$. Each M_i maps each session sequence $s \in \mathfrak{D}$ to decision space V such as $M_i : \mathfrak{D} \to V$.

The ultimate task in this work is to learn a *prediction function* to foretell the next users' actions on the site and $F(s, M_i) : s \times t_i \times V \mapsto \mathbb{R}$ be the evaluation function for model M_i.

The context-awareness is the special way to design the learning process that restricts the search space of M by defining in case of explicit context or discovering in case of hidden context the following functions:

1. The mapping from contextual features to the contextual categories: $\mathbf{G}: \Theta \to Cat$;

2. The mapping between contextual categories and individual learners: $\mathbf{H}: Cat \to M$.

DEFINITION 1 (TEMPORAL CONTEXT-AWARENESS). *The temporal context-awareness is a quadruple $(\theta_s, \boldsymbol{G}, \boldsymbol{H}, t_i)$.*

Web session decomposition. The segmentation Seg_i is defined separately for each web session $s_i \in \mathfrak{D}$ by $(k+1)$ segment boundaries: $1 = b_1 < b_2 < \cdots < b_k < b_{k+1} = (|s_i| + 1)$. The general segmentation for the set of sessions is specified as:

$$\mathfrak{L} = \{(Seg_1, \ldots, Seg_k)_i\}_{i=1}^{|\mathfrak{D}|}, \tag{2}$$

where Seg_i is homogenous segment of the session. The problem is (1) to discover a set of h contexts $\{C_{i=1}^h\}$ and (2) to decompose each $s_i \in \mathfrak{D}$ into k segments and (3) for each segments Seg_j and assignment of the context C_i.

3. MINING TEMPORAL CONTEXTS

3.1 Problem Definition

We assume that the data are generated as follows: the events alphabet $\mathfrak{A}$ is produced by h contexts. Under one specific context, the web session is generated by the events in that context only. Under that assumption, our goal is to decompose the web session (1), which is the sequence of users' actions on the site, into homogeneous pieces, such that the data in each segment can be described accurately by a temporal context C_i and a simple prediction model M_i. Formally, we have to decompose the event space into clusters each corresponds to a hidden context.

DEFINITION 2 (EVENTS CLUSTERING). *A clustering of web-session events $\mathfrak{A}$ is $E = \{E_1, \ldots, E_h\}$, where E_i is a cluster of $\mathfrak{A}$ and each E_i must satisfy the following properties:*

- *each cluster is not empty: $\forall E_i \in E : E_i \neq \emptyset$;*

- *all clusters are pairwise disjoint: $\forall i \neq j E_i \cap E_j = \emptyset$;*

- *every event belongs to one cluster: $\cup_i E_i = \mathfrak{A}$.*

The decomposition of the event space is not arbitrary but aims at maximizing the accuracy of the local models built for every context. Given the original log of sequences $\mathfrak{D}$ that is randomly divided two disjoint $\mathfrak{D}_{train}, \mathfrak{D}_{validation}$ and $\mathfrak{D}_{test}$. A decomposition of the event space into h clusters uniquely splits each sequence in the data into segments. Let $T_{train} = \cup_{j=1}^{Freq(C_i)}(Seg_j \in \mathfrak{D}_{train})$ be the set of segments in the training set that corresponds the context C_i. We learn sets of predictive models $\{M_i\}_{i=1}^h$ based on h sets of sessions segments. We validate our set of models based on $T_{validation} = \cup_{j=1}^{Freq(C_i)}(Seg_j \in \mathfrak{D}_{validation})$. And we test resulted clusters based on $T_{test} = \cup_{j=1}^{Freq(C_i)}(Seg_j \in \mathfrak{D}_{test})$. The effectiveness of one learner can be defined as:

$$F_{c_i}(T_{validation_i}, M_i) = \sum_{a_j \in T_{validation_i}} F(a_j, M_i) \tag{3}$$

For a given decomposition E, at the transition point, we always make wrong prediction. Therefore, the effectiveness of a decomposition E can be defined as:

$$EF(E) = \sum_{i=1}^h F_{c_i}(T_{validation_i}, M_i) \tag{4}$$

The problem of temporal context discovery is defined as an optimization problem where the goal is to find the event space decomposition E such that the prediction performance $EF(E)$ is maximized. We evaluate derived E based on the test set T_{test} and provide obtained accuracy that is calculate as in Equation 4.

3.2 Hierarchical clustering

Based on Equation 4 our objective function for the hierarchical clustering is:

$$EF^* = \arg\max_E EF(T_{validation}, M)), \tag{5}$$

where E^* is optimal clustering that satisfies Definition 2.

Algorithm 1 shows the general framework of *'Temporal Clustering Discovery'*. We start by placing each item in its own cluster (line 1) and building predictive models based on training sets that are related to each cluster (line 2). For example, we have two clusters $E_1 = \{a, b, c\}$ and $E_2 = \{d, f, e\}$ and a training set $Train$. We split $Train$ into two parts that are related to each clusters accordingly:

$$Train = \begin{matrix} \underbrace{abcb}_{\text{covered by } E_1} & / & \underbrace{ef}_{\text{covered by } E_2} & / & \underbrace{aa}_{\text{covered by } E_1} \\ \underbrace{acb}_{\text{covered by } E_1} & / & \underbrace{efde}_{\text{covered by } E_2} & / & \underbrace{ab}_{\text{covered by } E_1} \\ \underbrace{ababba}_{\text{covered by } E_1} & / & \underbrace{efd}_{\text{covered by } E_2} \end{matrix} \tag{6}$$

We obtain the following disjoint sets of sub-sessions:

$$\begin{aligned} Train_{E_1} &= \{abc, ab, ac, ab, ab, ab\} \\ Train_{E_2} &= \{efd, efde, ef, efde, ef\} \end{aligned} \tag{7}$$

Next, we build two prediction models: M_1 and M_2 based on $Train_{E_1}$ and $Train_{E_1}$ respectively. Then, we calculate effectiveness of obtained models by applying them on $Validation$ (line 3). We decompose the validation session using the same way as in 6:

$$Validation = \begin{matrix} \underbrace{abcac}_{\text{predict by } M_1} & / & \underbrace{efde}_{\text{predict by } M_2} \\ \underbrace{acab}_{\text{predict by } M_1} & / & \underbrace{efde}_{\text{predict by } M_2} \\ \underbrace{ab}_{\text{predict by } M_1} & / & \underbrace{ef}_{\text{predict by } M_2} & / & \underbrace{abba}_{\text{predict by } M_1} \end{matrix} \tag{8}$$

Then we iteratively detect two clusters which increase the general effectiveness (line 7) of the predictive system and merge them (line 9). In other words, during each iteration the method tries to increase the general effectiveness (4) of the system. If a merge increases the system effectiveness, the algorithm keeps it (line 8-9). We have resulted set of *clusters E* and set of predictive models $\{M_k\}_{k=1}^h$. We return *best clusters* (line 12).

The additional benefit of the suggested method is that it produces the "best" number of clusters thus discover number of hidden contexts. The general schema of proposed clustering technique is presented also in Figure 2 (Algorithm 1) for our the running example.
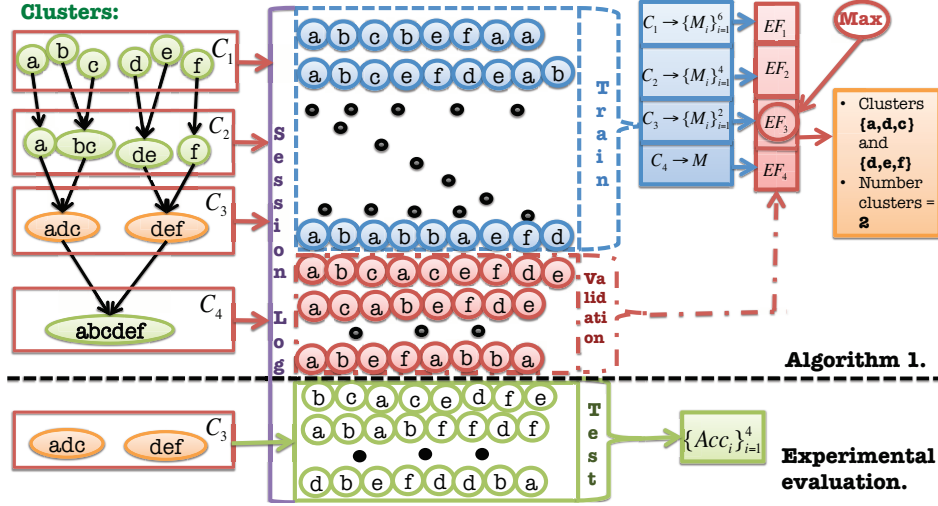
Figure 2: The general schema of proposed hierarchical clustering technique. The process of dividing training, validation and test sets is represented in Equation 6, Equation 8 and Equation 9 respectively.

Algorithm 1 Temporal contexts discovery (TCD)
___
Input: A web-session dataset $\mathfrak{D}$, train sessions $Train$, validation sessions $Validation$, test sessions $Test$, predefined type of predictive model M
Output: A clustering of events $E = \cup_{i=1}^{h} E_i$
1: $E \leftarrow \{\{a\}|a \in \mathfrak{A}\}$
2: $M \leftarrow buildPredictiveModels(Train, E))$
3: $EF_{opt} \leftarrow \text{argmax}_E E(Validation, M)$
4: **while** $|E| > 1$ **do**
5:    E_i, E_j
6:    $M_k \leftarrow buildPredictiveModel(Train \cap (E_i \cup E_j))$
7:    **if** $EF(Validation, M_k) > EF_{opt}$ **then**
8:      $EF_{opt} \leftarrow EF(Validation, M_k)$
9:      $E \leftarrow E_i \cup E_j$
10:    **end if**
11: **end while**
12: **return** E
___

3.3 Evaluation of hierarchical clustering

In order to evaluate obtained clusters we decompose the test set $Test$ according to resulted $E = \{E_1, E_2\}$ as presented in Equation 9. As an evaluation metric we use accuracy which is presented in Equation 10.

$$Test = \begin{array}{c} \underbrace{bcac}_{\text{predict by } M_1} \quad / \quad \underbrace{edfe}_{\text{predict by } M_2} \\ \underbrace{abab}_{\text{predict by } M_1} \quad / \quad \underbrace{ffdf}_{\text{predict by } M_2} \\ \underbrace{ab}_{\text{predict by } M_1} \quad / \quad \underbrace{fdd}_{\text{predict by } M_2} \quad / \quad \underbrace{ba}_{\text{predict by } M_1} \end{array} \quad (9)$$

$$Accuracy(E) = \sum_{i=1}^{h} F_{c_i}(T_{test_i}, M_i) \quad (10)$$

The general schema of the proposed experimental evaluation is described also in Figure 2 (Experimental Evaluation) for either our or the running example.

3.4 Defining Context-Awareness

The main reason to introduce the context-awareness into the predictive system is to improve the quality of forecast. Our hypothesis is that there are some external forces which change users' behaviour during a web session. We called these effects *temporal contexts*. If we can discover these contexts and build local predictive models for each context then we can improve the general effectiveness of the model (Equation 4). Algorithm 1 for *Temporal contexts discovery* directly optimizes the general effectiveness (Equation 5) .

Since we aim to ameliorate the quality of the next users' action prediction and we have web sessions, which are defined as sequences of users' actions, it is logical to apply the Markovian models [8], [3] ,[10], [15]. For simplicity of computation we apply first-order Markov models because we aim to see the difference between global and local models. If we get improvement for the first-order Markov models, then we might have the increase in quality for the variable order Markov models [3].

The identified context-awareness is:

1. The mapping from session events a_i to the clusters E_j that are the contextual features and the contextual categories respectively: $\mathbf{G}$: $a_i \rightarrow E_j$. Therefore the mapping $\mathbf{G}$ is identified by Algorithm 1.

2. The mapping between the clusters E_j and predictive models M_j is $\mathbf{H}$: $E_j \rightarrow M_j$. We train Markov models M_j based on $Train_{E_j}$ (as presented in the Equations 6 - 7).

Therefore, the temporal context at time t_i is quadruple $(a_i, \mathbf{G}, \mathbf{H}, t_i)$.

4. EXPERIMENTAL STUDY

We outline now the experimental methodology we employed to test Temporal Context Discovery algorithm that we presented. The goal of experimental study is to show that introducing of temporal contexts improves predictive accuracy. We first describe the dataset that we used for experiments, and then the design of the experiments.
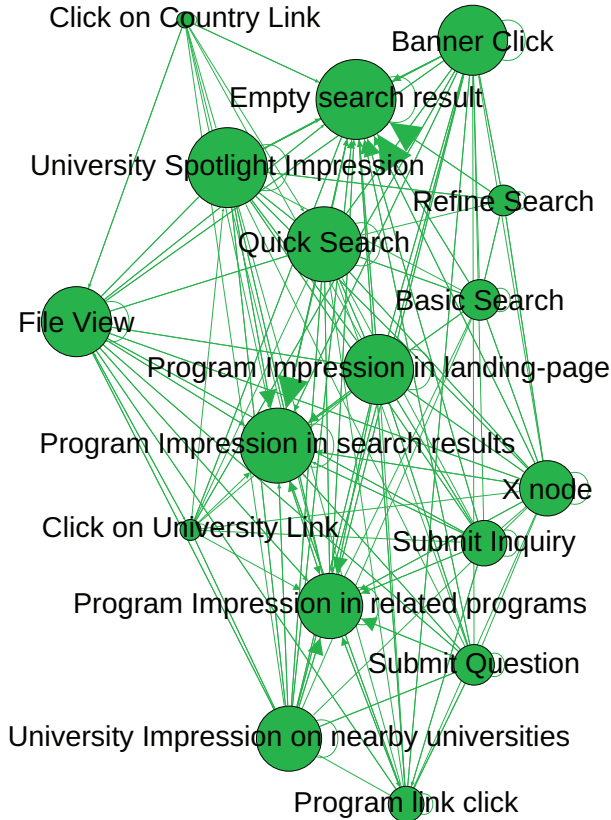
4.1 Data

Figure 3: A user navigation graph

The anonymized dataset for case study comes from *MastersPortal.eu*. The web-portal provides information about various study programmes in Europe. It was launched in 2007. In May 2012, *MastersPortal.eu* received over 1.5 million visits (excluding robots) and contained over 25.000 study programs.

MastersPortal.eu has a categorisation of actions that a user can perform on website. These actions are used to describe users' trails. These users' paths on the website can be transferred into a navigation graph.

DEFINITION 3 (USER NAVIGATION GRAPH). *A user navigation graph is a directed and weighted graph $G = (V, E)$, where V is a set of vertices corresponding to all possible user actions A and E are the set of edges (a_i, a_j). Each edge e of G is associated a weight $w(e)$ indicating the transition probability between two incident vertices of the edges.*

The users' navigation for the *MastersPortal.eu* is demonstrated in Figure 3, where possible values for users' actions a_i are presented. In total we consider sixteen major possible users' actions $A = \{a_1, a_2, \dots, a_{16}\}$ as presented in Figure 3. Different kinds of actions include the following categories:

- general type of actions on the site are **views**, **clicks**, **submissions**, **impressions**, and **searches** (quick - simple search from homepage, basic - when user uses special search page, refined - when additional filters are used);

- users can **view** a study file with additional information;

- users can **click** on a banner, or country information link, or a university link, or a program link;

- users can have feedback from the site through **submit**: question or inquiry;

- **impressions** are refereed to the recommendation actions;

- X node is the action which is out of the categorisation scope.

The data set for the experiment contains all navigation paths of visitors of the website collected during May 2012. The log data contains about 350.000 anonymized sessions [1].
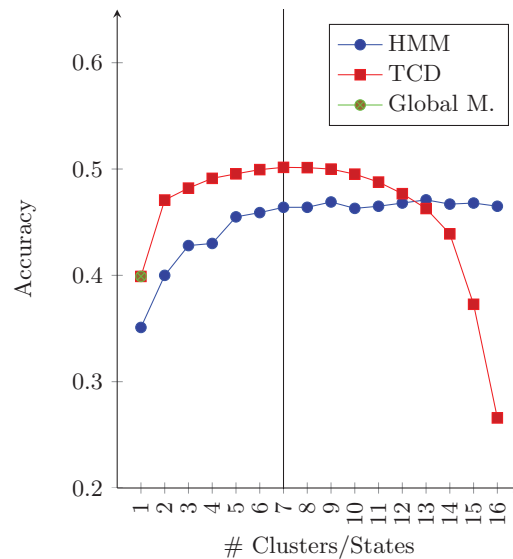
4.2 Results

Figure 4: Resulted effectiveness

Baseline for temporal context discovery. An alternative approach to analyze the sequential data is a Hidden Markov Model (HMM) [11]. This model observes whatever the data is generated by an unknown process that takes several unobserved states, and different states output different observation.

We have defined context-awareness in Section 3.4 which is used in Algorithm 1. We have implemented "Temporal contexts discovery (TCD)" that is presented on he Algorithm 1 in order to obtain clusters of users' activities. Each cluster is associated with a temporal context according to Definition 1.

We have run 10 times both algorithms: HMM and TCD. The method was trained and tested based on randomly splited data. The obtained average metrics are presented Figure 4. It is clear from the plot that TCD outperforms HMM.

---

[1] We made this dataset publicly available as a benchmark. Please refer to the section *Code and Datasets* section at http://www.win.tue.nl/~mpechen/projects/capa/

The accuracy of global model "Global M." is depicted as green dot in Figure 4. The global model equals to TCD when TCD has one cluster. We also show that the set of contextual local models outperforms the global model.

Our objective function for TCD reaches a maximum when number of clusters equals to 7. Therefore, an average number of elements in the cluster is 2.3. It happens because users use repetitive states as shown in Equation 11.

$$s_i = \langle \underbrace{(A_1, t_1), \ldots, (A_i, t_i), \ldots, (A_8, t_8)}_{A_i = \text{'Program impressions in related programs'}}, \\ \ldots, \underbrace{(A_9, t_9)}_{A_9 = \text{'Click on Banner'}} \rangle \quad (11)$$

Table 1: The resulted clusters

Id	Summary	Cluster
1	Intensive Search	Basic Search, Refine Search, Empty Search Result
2	Explore information related to program	Program impression in search result, Banner click, Program click , Click on university link
3	Start of browsing	University Spotlight impression, Quick search
4	Explore country information	File view, Click on country link
5	Explore search result	Program impressions in search results, University impression on nearby universities
6	Explore program	Program in landing page, Submit inquiry
7	-	Submit question, X-node

Table 1 presents seven clusters that we discovered in the dataset. We provide a summary for each cluster in the table. Such summary can be used to provide a users' roadmap on the site. It helps to better understand the user's behaviour on the site and to discover existing problems. Like in our example, the cluster "Intensive Search" shows that users encounter a problem in order to find information. The cluster 7 looks like an outlier because the action 'Submit question' can be made from every page on the site, and we can't decipher 'X-node'. It is a good point that algorithm can filter out the noise.

Figure 5 provides some further insights on the interpretability of the obtained clustering. The figure shows the correspondence of available navigation across the website pages and obtained clusters. We intensity of colors correspond to the values of Jaccard index between the page and the cluster. There is some strong relation between the possible actions that users can perform at certain website pages and corresponding clusters; there are five pages that correspond pretty well to four clusters. We are not interested in cluster 'Outlier', so we do not discuss it:

1. 'University page' corresponds to cluster 'Explore information related to program';
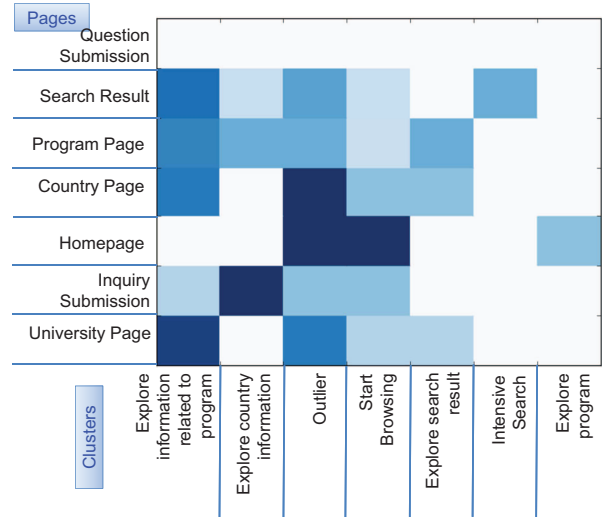

Figure 5: Visualization of the clusters.

2. 'Inquiry submission page' corresponds to cluster 'Explore country information';

3. 'Homepage' corresponds corresponds to cluster 'Start Browsing'.

This type of analysis might help to improve design of a website. But for our study it serves as an additional evidence that the proposed approach for hidden context discovery is reasonable.

5. RELATED WORK

Context often has a significant impact on the way humans (or machines) act and on how they interpret things; furthermore, in context switching causes a transformation in the experience that is going to be lived. Predictive Web Analytics' goal is to foretell users' interests based on the discovered behaviour patterns and it can be considered as a particular case of a context-aware application. Many studies have demonstrated that integrating context-awareness into predictive modeling helps to better understand user information needs and improves the effectiveness of ranking [5], [17], query classification [4] and recommendations [14], [1].

The first approaches for context-aware modeling assumed that contexts were given explicitly and focused on the integration of the this additional data source into the space of predictive features. Later approaches considered algorithmic aspects of using explicit and implicit contextual information for local model selection [12] selection of instances to learn a model from and to perform model output correction [16].

Our approach has some meta-learning flavor, but unlike the work in [12] we consider the mapping from contextual features to contextual categories to be known and deterministic and focus on how to discover contexts. That is defined as an optimization problem unlike in [12] where clustering in performed independently from finding a mapping from contexts to local models.

In machine learning the term 'contextual' usually characterizes the features that do not determine or influence the class of an object directly [16], [19], but improve predictive performance when used together with other predictive fea-

tures. Some recent approaches consider how to derive such features. E.g. in [18] feature transformations requiring the resulting contexts to be independent from the class labels have being explored.

User modelling for personalization is a fundamental and challenging problem. For modeling user behavior (navigation) on the Web, the use of Markov models [15] is a reasonable choice as they are compact, simple and based on well-establish theory. Several types of Markov models were proposed for modelling user Web data: first-order Markov models, hybrid-order tree-like Markov models [10], prediction by partial match forest [7], kth-order Markov models (FOMMs) [9], variable order Markov models (VOMMs) [3] that provide the mean to capture both large and small order Markov dependencies.

6. CONCLUSION AND FUTURE WORK

In this paper, we addressed an important and challenging problem – the discovery of temporal hidden contexts in the web-sessions. We formulated it as an optimisation problem and developed a novel method to discover the temporal contexts by decomposing web sessions into non-overlapping segments, where each segment would relate to one specific context.

We applied hierarchical clustering to merge the activities of the user navigation graph and thus identify temporal contexts.

We demonstrated with an experimental case study on the real-world dataset that the contexts as we defined them do exist in the user navigation graph and are useful for improving the prediction of the next user actions. Indeed, the predictive accuracies of local Markov models that capture the specificity of the users' behaviour were higher than the accuracies of the global models. Interestingly, the discovered contexts were meaningful and we were able to understand and interpret the resulted clustering.

Our study opens an interesting research direction within the area of mining hidden contexts. The future work would cover the following accepts: testing our method on other datasets, introducing a mechanism for detecting context switching within a web-session, considering multidimensional contextual features.

Acknowledgements

7. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. *CARS*, 2010.

[2] M. Aly, A. Hatch, V. Josifovski, and V. K. Narayanan. Web-scale user modeling for targeting. In *WWW*, 2012.

[3] J. Borges and M. Levene. Evaluating variable-length markov chain models for analysis of user web navigation sessions. *IEEE Trans. Knowl. Data Eng. (TKDE)*, 19(4):441–452, 2007.

[4] H. Cao, D. H. Hu, D. Shen, D. Jiang, J.-T. Sun, E. Chen, and Q. Yang. Context-aware query classification. In *SIGIR*, 2009.

[5] H. Cao, D. Jiang, J. Pei, E. Chen, and H. Li. Towards context-aware search by learning a very large variable length hidden markov model from search logs. In *WWW*, pages 191–200, 2009.

[6] D. Chakrabarti, D. Agarwal, and V. Josifovski. Contextual advertising by combining relevance with click feedback. In *WWW*, 2008.

[7] X. Chen and X. Zhang. A popularity-based prediction model for web prefetching. *Computer*, 36(6):63–70, 2003.

[8] F. Chierichetti, R. Kumar, P. Raghavan, and T. Sarlós. Are web users really markovian? In *WWW*, pages 609–618, 2012.

[9] M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. *ACM Trans. Internet Techn. (TOIT)*, 4((2)):163–184, 2004.

[10] X. Dongshan and S. Junyi. A new markov model for web access prediction. *Computing in Science and Engineering*, 4(6):34–39, 2002.

[11] S. Hyvönen, A. Gionis, and H. Mannila. Recurrent predictive models for sequence recurrent predictive models for sequence segmentation. *IDA*, pages 195–206, 2007.

[12] O. Mazhelis, I. Zliobaite, and M. Pechenizkiy. Context-aware personal route recognition. In T. Elomaa, J. Hollmén, and H. Mannila, editors, *Discovery Science*, volume 6926 of *Lecture Notes in Computer Science*, pages 221–235. Springer, 2011.

[13] C. Palmisano, A. Tuzhilin, and M. Gorgoglione. Using context to improve predictive modeling of customers in personalization applications. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 20(11):1535–1549, November 2008.

[14] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *SIGIR*, volume 10, 2011.

[15] R. Sarukkai. Link prediction and path analysis using markov chains. *Computer Networks (CN)*, 33((1-6)):377–386, 2000.

[16] P. Turney. The identification of context-sensitive features: A formal definition of context for concept learning. 2002.

[17] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li. Context-aware ranking in web search. In *SIGIR*, 2010.

[18] I. Zliobaite. Identifying hidden contexts in classification. In *PAKDD (1)*, volume 6634 of *LNCS*, pages 277–288. Springer, 2011.

[19] I. Zliobaite, J. Bakker, and M. Pechenizkiy. Beating the baseline prediction in food sales: How intelligent an intelligent predictor is? *Expert Syst. Appl. (ESWA)*, 39(1):806–815, 2012.