Predicting Current User Intent with Contextual Markov Models

Julia Kiseleva, Hoang Thanh Lam, Mykola Pechenizkiy
Department of Computer Science
Eindhoven University of Technology
P.O. Box 513, NL-5600MB, the Netherlands
{t.l.hoang, j.kiseleva, m.pechenizkiy}@tue.nl

Toon Calders
Computer & Decision Engineering Department
Université Libre de Bruxelles
Avenue F.D. Roosevelt 50, B-1050 Bruxelles, Belgium
Toon.Calders@ulb.ac.be

***Abstract*—In many web information systems like e-shops and information portals predictive modeling is used to understand user intentions based on their browsing behavior. User behavior is inherently sensitive to various contexts. Identifying such relevant contexts can help to improve the prediction performance. In this work, we propose a formal approach in which the context discovery process is defined as an optimization problem. For simplicity we assume a concrete yet generic scenario in which context is considered to be a secondary label of an instance that is either known from the available contextual attribute (e.g. user location) or can be induced from the training data (e.g. novice vs. expert user). In an ideal case, the objective function of the optimization problem has an analytical form enabling us to design a context discovery algorithm solving the optimization problem directly. An example with Markov models, a typical approach for modeling user browsing behavior, shows that the derived analytical form of the optimization problem provides us with useful mathematical insights of the problem. Experiments with a real-world use-case show that we can discover useful contexts allowing us to significantly improve the prediction of user intentions with contextual Markov models.**

***Keywords*—intent prediction; context-awareness;**

I. Introduction

In many web applications, contextual information is available along with the data. This information is very useful for predictive analytics. For example, in a weblog of users' activities on a website, the contextual information such as 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, e.g. training and employing a local model for each of the contexts recognized to be useful.

There are several important groups of research questions in context-aware predictive analytics related to 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* contexts from data. In many cases, contextual information is provided explicitly in the form of additional features describing e.g. user current location. These *explicit* contexts are different from *implicit* contexts that can be only inferred from data. For instance, we may have no explicit information whether a user is very well familiar with a particular website's functionality or falls into the category of novice users. However, a context discovery approach may be able to infer such information automatically based on the number of visits or access patterns of the user. For simplicity, but without loosing the generality of our study we assume that all the contexts are non-overlapping and that a user is at any moment is associated only with one contextual category.

If we want to focus predictive modeling on a subset of the most promising contexts, we can perform data exploration and use domain expertise for choosing an appropriate subset of contexts. However, for complicated and large-scale datasets, deep understanding of data by exploration can be rather limited. In addition to that, in many cases, domain expertise may be not always available (yet). Therefore, an alternative straightforward context selection approach is direct evaluation of every subset of the set of targeted contexts through predictive accuracy testing. This solution is computationally demanding when the set of targeted contexts has a high cardinality. Moreover, in several cases, evaluation must be done in an online settings with a real information system in operation that is also very expensive and time demanding.

In this work, we formulate (useful) context discovery as an optimization problem. Even when domain experts are not available and data exploration gives only partial knowledge about the data and if direct context evaluation and testing are expensive, it is still possible to select a good set of contexts if we know the closed form of the objective function of the optimization problem. On one hand, an analytical form of the objective function provides us with useful mathematical insights of the problem. It may give us a good hint for context discovery even in the case that the optimization problem is hard. On the other hand, it enables us to evaluate the contexts in an off-line settings before performing an online testing with a small set of selective contexts.

We focus on Markov models that are commonly used for modeling web user behavior. Our analysis shows that the objective function calculated as the expected accuracy of prediction using the Markov model has a closed form. Analyzing the analytical form of the objective function helps us to find interesting properties of adopting contextual information for prediction with Markov model. Namely, *if the data are generated by a Markov model, any context preserving the Markovian property in each contextual category is useful in the sense that the accuracy of prediction using Markov model*

built for each category of the context is at least as large as the expected accuracy of the Markov model built for the whole data. This property is a theoretical judgement of context-ware method for prediction using Markov models.

We conducted experiments on a real dataset to illustrate that 1) if useful contexts are discovered, the local Markov models predict user intentions statistically significantly better than the global model, and 2) local Markov models perform well, i.e. not significantly worse than a global model, even when the contexts are absolutely not useful and have substantially smaller number of instances to induce local models.

The rest of the paper is organised as follows. In Section II we introduced related work on context-awareness in supervised learning applications. In Section III we introduce definitions of context-aware predictive analytics. In Section IV we introduce a specific case of using contextual information for improving prediction ability of Markov models. In Section V we propose the context discovery method based on navigation graph clustering. We present our experimental study using a real Web portal data sets in Sections VI and VII. Section VIII concludes.

II. Related Work

Many studies have demonstrated that integrating context-awareness into predictive modeling helps to better understand user information needs and improves the effectiveness of ranking [18], query classification [6] and recommendations [12].

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, or using it for learning local models and hybrid models, for correction of model outputs or performing contextual selection of instances to learn a model from [15]. Examples of contextual information include current date, season, weather [5], users' location [16] and emotional status.

In machine learning research the term usually characterizes the features that do not determine or influence the class of an object directly, but improve predictive performance when used together with other predictive features [15]. Recent approaches consider how to derive such features. E.g. in [19] feature transformations requiring the resulting contexts to be independent from the class labels have being explored.

In this work we consider both cases, when contextual information is given and when we need to derive it based on some assumptions on what kind of useful hidden information may be present in the data. User modelling for personalization is a fundamental and challenging problem. For modeling user behavior (navigation) on the Web, the use of Markov models is a reasonable choice as they are compact, simple and based on well-establish theory. Several Markov models were proposed for modelling user Web data: first-order Markov model, hybrid-order tree-like Markov model [10], prediction by partial match forest [7], kth-order Markov models [9], variable order Markov model (VOMM) [4] that provide the mean to capture both large and small order Markov dependencies. Recently, it was shown in [8] on large data set that it is better to use the variable order Markov models for this purpose. Other, perhaps the most commonly used techniques, are based on Hidden Markov Models (HMM). However, work with HMMs typically requires understanding of the domain and very large training samples [2]. In [11] a hierarchical clustering approach was proposed for decomposing users' web sessions into non-overlapping temporal segments. In the experimental study it was shown that such temporal context can be identified and used for more accurate next user action prediction with Markov models.

In this work we also study context-awareness with the class of Markov models.

III. Contextual Prediction

This section discusses generalized definitions of contextual predictive analytics. Let D be the set of all possible data instances. As a running example, we consider a website containing five different activities with categorical labels a, b, c, d and e. Every user visiting the website produces a sequence of transition activities corresponding to the categories that the user has visited. In this example, D is the set of all possible sequences of activities from the categories a, b, c, d and e.

Let $\Theta = C_1 \times C_2 \times C_3 \times \cdots \times C_N$ be the space of all possible contextual features associated with every data instance, where each C_i is a context. Denote $\theta_s \in \Theta$ as the contextual feature vector associated with sequence s.

Let $M : \Theta \times D \mapsto V$ be a predictive model that maps each test sequence $s \in D$ associated with the contextual information θ_s to the decision space V. Let $F(s, M(\theta_s, s)) : kD \times V \mapsto R$ 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. $V \equiv kD$. An example of the evaluation function is the number true predictions made by M over the test instance s. For instance, assume that the model M predicts $s = ababc$ as $M(\theta_s, s) = \underline{abedc}$ then it makes three true predictions corresponding to the underlined activities, i.e. $F(s, M(\theta_s, s)) = 3$.

Let $T \subseteq kD$ be a set of test instances and denote $Pr(s)$ as the probability that $s \in T$. The expectation of the evaluation function $F(s, M(\theta_s, s))$ over the test set is defined as $E[T, M] = \sum_{s \in T} Pr(s).F(s, M(\theta_s, s))$. The value of the expectation $E[T, M]$ can be considered as an objective that we need to optimize and assume that M^* is the optimal model, i.e. $M^* \arg\max_M E[T, M]$.

Let C be a context with n categories: $C = \{c_1, c_2, \ldots, c_n\}$ associated with each data instance $s \in kD$. A context may have different categories, e.g. the *region* context can be divided into four categories such as Europe, Africa, American, or Asia. For simplifying the discussion, we consider contexts that have only two categories. The discussion of the general cases with more than two categories is very similar.

Assume that we have a context C with two categories c_1 and c_2 dividing the test set into two disjoint subsets T_1 and T_2 such that $T = T_1 \cup T_2$. Denote M_1 and M_2 as two predictive models built for the category c_1 and c_2 respectively. Let $P(c_1)$ and $P(c_2)$ are probabilities that a test instance belonging to the category c_1 and c_2 respectively.

Theorem 1 (Contextual Principle): Let M^* be an optimal model on T then it is a combination of M_1^* and M_2^*. Where

M_1^* is an optimal model for T_1 and M_2^* is an optimal model for T_2.

Proof: Because $M_1^* = \arg\max_{M_1} E[T_1, M_1]$ and $M_2^* = \arg\max_{M_2} E[T_2, M_2]$ we must have $E[T_1, M_1^*] \geq E[T_1, M^*]$ and $E[T_2, M_2^*] \geq E[T_2, M^*]$. We further derive:

$$P(c_1)E[T_1, M_1^*] \geq P(c_1)E[T_1, M^*] \quad (1)$$
$$P(c_2)E[T_2, M_2^*] \geq P(c_2)E[T_2, M^*] \quad (2)$$

$$P(c_1)E[T_1, M_1^*] + P(c_2)E[T_2, M_2^*] \geq E[T, M^*] \quad (3)$$

On the other hand, since $M^* = \arg\max_M E[T, M]$, we have:

$$E[T, M^*] \geq P(c_1)E[T_1, M_1^*] + P(c_2)E[T_2, M_2^*] \quad (4)$$

From two inequalities 3 and 4 we imply that: $E[T, M^*] = P(c_1)E[T_1, M_1^*] + P(c_2)E[T_2, M_2^*]$. In other words, M^* is a combination of M_1^* and M_2^*. ■

Theorem 1 shows that the problem of finding the best model for every test instance can be solved by considering the subproblems of finding optimal models for test instances in each individual contextual category. This result provides us with theoretical judgement for personalization and exploitation of contextual information in predictive analytics.

Nevertheless, in practice finding an optimal model for each contextual category is usually as hard as finding an optimal model for the whole data. Indeed, it is usually the case that the type of model is chosen in advance, e.g. Markov models. Model's parameters are estimated from training data D. Under this circumstance, contextual predictive analytics seeks for a context such that it divides the training data into two subsets D_1 and D_2 and the predictive models trained on D_1 and D_2 improve the predictive performance in comparison to the model trained on the whole training data. To this end, we define useful contexts as follows:

Definition 1 (Useful Context): Given a model M built based upon the whole training data D and M_1, M_2 are two models built based upon D_1 and D_2 corresponding to each contextual category of a context C respectively. The context C is useful if an only if: $E[T_1, M_1] \geq E[T_1, M]$ and $E[T_2, M_2] \geq E[T_2, M]$

IV. Contextual Markov Models

This section discusses a specific case of using contextual information for improving prediction ability of Markov models. In particular, we are given log of sequences of activities performed by users in a web application. The task is to predict the next activity in a sequence. Markov model is chosen as a predictive model for this problem. We are interested in finding useful context such that Markov models built for each category of the context improve the prediction performance compared to the Markov model built for the whole data. We call this problem as the *contextual Markov model*.

To simplify the discussion, we only consider the special case with the first order Markov model or Markov chain. Generalization of our discussion to Markov models with any order is similar to that special case. Let us denote $kA = \{a_1, a_2, \cdots, a_n\}$ as the set of all possible activity. A Markov chain M is associated with a transition probability matrix $[P(a_j|a_i)]$, where $P(a_j|a_i)$ is the probability of transition from the activity a_i to the activity a_j.

For any activity $a \in kA$, we denote $m(a)$ as the activity with highest transition probability from the activity a, i.e. $m(a) = \arg\max_{b \in kA} Pr(b|a)$. Given that the current state is the activity a, if the data follows Markovian property then $m(a)$ is always the best prediction of the next state. Therefore, we consider a predictor which always chooses the most probable transition for the next state. If the test sequences in T are random samples from the Markov model M, the expected accuracy of the predictor, i.e. the expectation of true prediction rate can be calculated as follows:

$$E[T, M] = \Sigma_{a \in kA} P(a)P(m(a)|a) \quad (5)$$

Let $C = \{c_1, c_2\}$ be any context and M_1, M_2 are two Markov chains built for each categories c_1 and c_2 respectively. Consider a new predictive model that uses M_1 to predict test sequences belonging to T_1 corresponding to the first category c_1 and uses M_2 to predict test sequences belonging to T_2 corresponding to the second category c_2. We also denote $[P_1(a_j|a_i)]$ as the transition matrix of the Markov model M_1 and $[P_2(a_j|a_i)]$ as the transition matrix of the Markov model M_2. If two test sets T_1 and T_2 contain randomly sampled sequences from two Markov models M_1 and M_2 then the expected accuracy of this prediction can be calculated as follows:

$$E[T, M_1, M_2] = P(c_1)E[T_1, M_1] + P(c_2)E[T_2, M_2] \quad (6)$$

where $P(c_1)$ and $P(c_2)$ stand for the probability of the test sequence belonging to the first and the second category respectively and:

$$E[T_1, M_1] = \sum_{a \in kA} P_1(a)P_1(m_1(a)|a) \quad (7)$$
$$E[T_2, M_2] = \sum_{a \in kA} P_2(a)P_2(m_2(a)|a) \quad (8)$$

Theorem 2: Assume that the test data possess the Markovian property and this property holds for every category of a context C. Moreover, the training data D together with D_1 and D_2 are large enough such that we can learn accurate Markov models M, M_1 and M_2 then that context is useful, i.e.: $E(T_1, M_1) \geq E(T_1, M)$ and $E(T_2, M_2) \geq E(T_2, M)$

Proof: Under the category c_1, let $P(m(a), a|c_1)$ be the probability of the event indicating that the current activity is a and the next activity is $m(a)$. We have:

$$E[T_1, M] = \sum_{a \in kA} P(m(a), a|c_1) \quad (9)$$
$$= \sum_{a \in kA} P(m(a)|a, c_1).P(a|c_1) \quad (10)$$
$$= \sum_{a \in kA} P(m(a)|a, c_1).P_1(a) \quad (11)$$
$$\leq \sum_{a \in kA} P_1(m_1(a)|a, c_1).P_1(a) \quad (12)$$
$$\leq E[T_1, M_1] \quad (13)$$

The inequality $E[T_2, M] \leq E[T_2, M_2]$ can be derived in a similar way from which the theorem is proved. ■

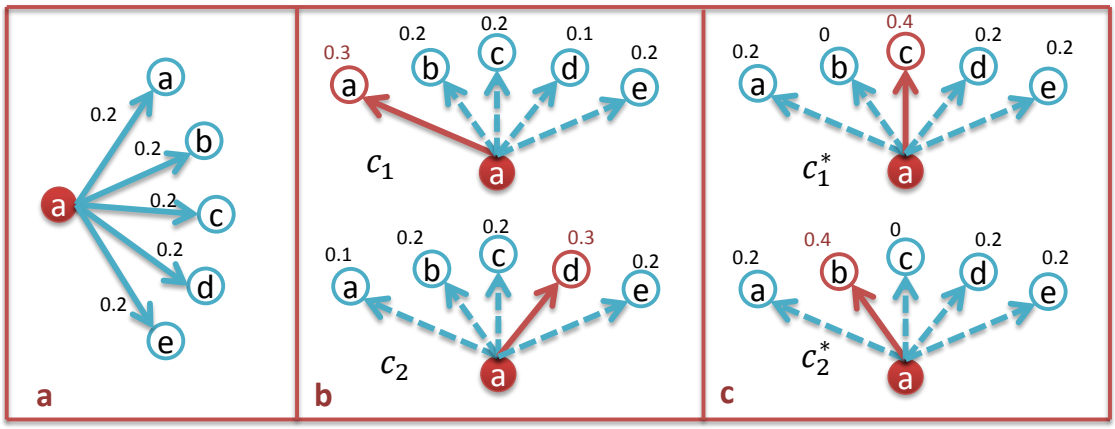

Fig. 1. An example of transition distributions from a to the other states. Two contexts $C = \{c_1, c_2\}$ and $C^* = \{c_1^*, c_2^*\}$ have different transition distributions. The most probable transition paths are highlighted with red-purple color.

Theorem 2 shows that if the data possess the Markovian property then exploiting any context preserving the Markovian property is always beneficial. This theorem can be considered as a theoretical judgement for using contexts to improve Markov model. In practice, Markov models are usually learnt from training data. The accuracy of model's parameters estimation is highly dependent on the amount of available data. When we exploit a context, the training data is split into smaller portions by the context which may cause the decline in the accuracy of parameter estimation.

Finally, the contextual Markov problem is defined as an optimization problem as follows:

Definition 2 (Contextual Markov): Given training data D, find the context $C = \{c_1, c_2\}$ splitting D into D_1 and D_2 such that the Markov models M_1 and M_2 learnt from D_1 and D_2 respectively maximize the evaluation function on the test set T: $E[T, M_1, M_2]$.

V. Context Discovery Techniques

A. Clustering-Based Approach

In order to illustrate the key idea behind our proposed algorithm, consider an example in Figure 1 where transition probabilities $P(x|a)$ ($x \in \{a, b, c, d, e\}$) from the current state a to the other states are shown. Figures 1.b and 1.c show the transition probability $P(x|a, C)$ ($x \in \{a, b, c, d, e\}$) in two different contexts $C = \{c_1, c_2\}$ and $C^* = \{c_1^*, c_2^*\}$.

In Figure 1.a, all transitions are equally probable. Therefore, the transition probability distribution from a has very high entropy making prediction ineffective. If we use the predictor always predicting the most probable transition, the expected true prediction rate is 0.2.

The situation is changed when we consider two contexts C and C^*. In particular, in Figure 1.b, the distributions $P(x|a, c_1)$ and $P(x|a, c_2)$ both have lower entropy than the transition distribution $P(x|a)$. Under the context C, the true prediction is $P(m(a)|a, c_1) = 0.3$ in the category c_1 and $P(m(a)|a, c_2) = 0.3$ in the second category. Similarly, under the context C^* the true prediction rate is $P(m(a)|a, c_1^*) = 0.4$ in the category c_1^* and $P(m(a)|a, c_2^*) = 0.4$ in the second category. Therefore, by exploiting the context C^* we may increase the prediction accuracy from 0.2 to 0.4.

Common sense tells us that the prediction is easier if the context splits the data into homogeneous groups. In doing so, users with similar behavior are grouped together which may result in low-entropy transition distribution. A possible clustering algorithm to group users is an agglomerative hierarchical clustering algorithm which uses the objective function $E[T, M_1, M_2]$ as a principle for merging clusters. Overall, our approach consists of two important components: (1) a *clustering algorithm* which groups training sequence into groups with similar sequences and (2) an *alignment procedure*which assigns new test sequence to clusters given partial content of the test sequence being seen so far.

B. Clustering by Community Detection

A general representation of users' historical behavior is given as a log of web sessions $kD = \{S_1, S_2, \ldots, S_n\}$ where each web session is a sequence of states $S_i = (a_1, a_2, \cdots, a_m)$ corresponding to historical browsing activities of a user. In our case the users' actions are categorized by the type of the users' actions: *searches*, *clicks on ads* or *homepage visits*. A complete set of used categories is presented in Figure 2 as graph nodes. However the set of all possible activity states depends on needs of a particular service e.g. a visit of the home page can be considered as an activity. Thus, activities and their possible orderings within user web sessions can be summarized as a *user 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 kA 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.

Depending on user experience they may perform different activities by visiting different states in the navigation graph. Therefore, we propose a user action clustering method based on community detection in the navigation graph. We want to understand if there are any groups of nodes in the navigation graph and then use this knowledge to characterize the users'

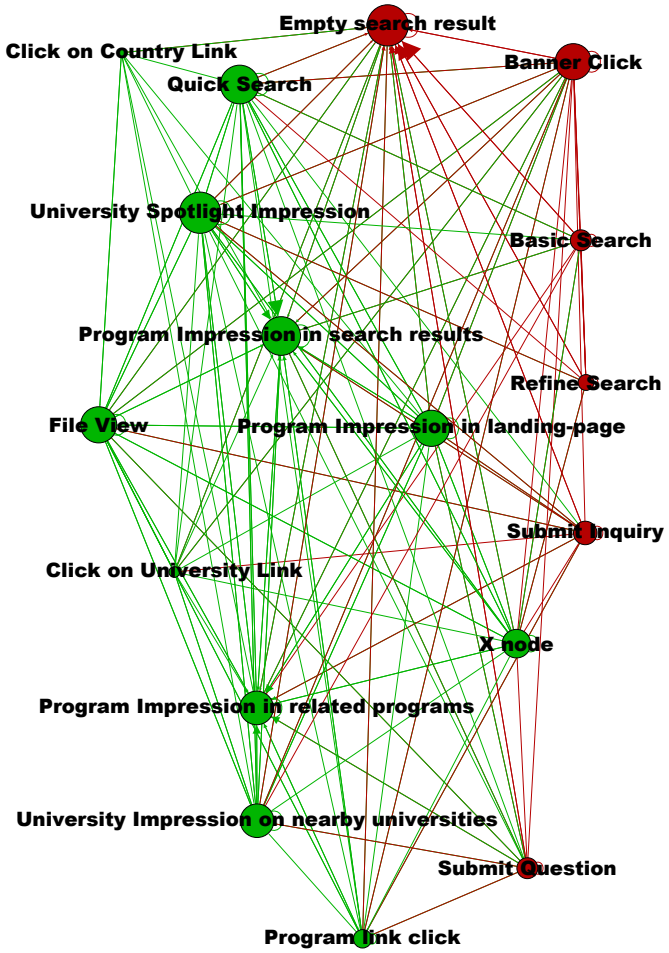

Fig. 2. A user navigation graph. The meaning of nodes is described in Section VI-A in details. A graph partitioning algorithm is used to detect two communities in the graph: the red states are associated with "expert" users and the green states are associated with "novice" users.

behaviour. Intuitively there are two types of user's behaviour on a site: (1) "expert" users, who is experienced with website interface or searches extensively to find required information, and (2) "novice" user, who needs more time to learn about a website or is not interested much in content

Assume that we have n graph partitions by using a communities detection method. Discovered groups of states may be interpret after analysis e.g. V_i corresponds to a "novice" user's behaviour. However, if n is too big to analyze then we have clusters: $\{V_i\}_{i=1}^{n}$. To simplify the discussion, we consider two clusters: Let V_{exp} and V_{nov}. V_{exp} corresponds to states which are visited by a "expert" users (red states in Figure 2) and V_{nov} consists of activates related to a "novice" user (green sector in Figure 2). Having clusters of states, we can align each web session with corresponding clusters. If a web session contains both red and green states, the alignment is performed by examining the sequence from the left to the right: if expert state encounters in a sequence then method assigns the sequence to V_{exp}, otherwise, a session is assigned to V_{nov} so far. Let us call this procedure *sequential user alignment*. For example, let $S_i = ceabbbaa$,$V_{exp} = \{a, b\}$ and $V_{nov} = \{c, d, e\}$. The alignment function A aligns the sequence S_i to clusters as follows. Fist, A sees the state c and S_i is temporarily assigned to V_{nov}. Then, A again sees another "novice" state e and S_i remains in V_{nov}. In the third step, A encounters "expert" state a and S_i is moved to V_{exp}.

In this case, type of user's behaviour is a context. The states in the navigation graph which are visited by user during session S_j is contextual features θ_{S_j}. $\{V_i\}_{i=1}^{n}$ represent contextual categories. The proposed alignment approach allows us to effectively align training sequences to clusters. More importantly, the alignment is very convenient for sequentially aligning test sequences to clusters. In the experiments, we show that this approach works well with a specific real-world use-case. Moreover, the proposed approach can easily be generalized to any sequence data. Vincent et al. [3] introduce an algorithm that finds high modularity partitions of networks in a short time and that unfolds a complete hierarchical community structure for the network, thereby giving access to different resolutions of community detection.

C. Clustering by Geographical Position

Our dataset contains a user location. In the literature, it was shown that the *users' location* is useful contextual information in many applications [16], [14], [1]. A context based on geographical location can have different levels of granularity like continent, country, city and so on. In our experiments we concentrate on a continent level due to limitations from the evaluation side.

Grouping session. We use user IP addresses as contextual features, then $\theta_s = IP$ is contextual vector associated with session s. We define six contextual categories: $C_{geo} = \{C_1 = Europe, C_2 = Africa, C_3 = North\ America, C_4 = South\ America, C_5 = Asia, C_6 = Oceania\}$.

Geographical alignment function. kD is divided into six disjoint training sets associated with the continent: kD_{Europe}, kD_{Africa}, kD_{Asia}, $kD_{NorthAmerica}$, $kD_{SouthAmerica}$, $kD_{Oceania}$.

VI. Experimental Study

A. Data

The anonymized dataset for our case study comes from *StudyPortals.eu*. The web-portal provides information about various study programmes in Europe. We used data that was collected in May 2012, dataset contained over 350.000 session[1]. Each user's session is recorded as following: *user IP address, the two timestamps indicating the start of the session and the end of the session, sequence of the user's actions*. *StudyPortals.eu* has a categorisation of actions a user can perform on their website to describe users' transitions. This taxonomy is used to describe users' paths on the website which can be transferred into a navigation graph. The users' navigation for the *StudyPortals.eu* is demonstrated in Figure 2, where possible values for users' actions a_i are presented: $A = \{a_1, a_2, \ldots, a_{16}\}$. General type of action on the site is **view** (exp. **view** study file with additional information), **click** (**click** on a banner, or a country information link, or a

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

university link, or a program link), **submission** (user feedback through question or inquiry **submit**), **impression** (referee to the recommendation actions), and **search** (**quick search** - simple search from homepage, **basic search** - when user uses special search page, **refined search** - when additional filters are used). X node is the action which is out of the categorisation scope.

B. Experiment Design

Our sequential dataset of users's actions is randomly split into two parts: test set T - 20% of the and training set (Tr) contains - 80% of the session log.

Training phase. The whole Tr is used to learn a global predictive model M (Glob.). Using the *sequential user alignment* method we divide the whole training dataset into subsets, which apply to each of the contexts. Assume we have context C with k categories $\{c\}_{i=1}^{k}$ dividing the train set into k disjoint subsets $\{Tr_i\}_{i=1}^{k}$ such that $Tr = Tr_1 \cup Tr_2 \dots Tr_k$. Let denote $\{M_i\}_{i=1}^{k}$ as k predictive models built for categories $\{c_i\}_{i=1}^{k}$ respectively.

Testing phase. During the testing stage we calculate accuracy of global model M - $Acc[T, M]$ (Equation 5). k categories $\{c\}_{i=1}^{k}$ divide the test set T into k disjoint subsets $\{T_i\}_{i=1}^{k}$ such that $T = T_1 \cup T_2 \dots T_k$. Let $P(c_i)$ is a probability that the test instance belongs to the category c_i then $Acc[T, \{M\}_{i=1}^{k}] = \sum_{i=1}^{k} P(c_k) Acc[T_i, M_i]$ (Equation 6).

As predictive models we use the following Markov models: FOMM, CTW [17] and PST [13]. To calculate the final metrics we run the described evaluation 10 procedures times to collect average metrics. We run the evaluation cycle for two discussed contexts: users' type ("novice" vs. "expert") and geographical location.

VII. Experimental Results

A. Context by Geographical Position

In this experiment we want to evaluate an impact of context based on geographical location of a user. The method to obtain the context based on geographical position is described in details in Section V-C. We use mapping from contextual feature *IP address* to a continent as alignment method to cluster the session. Therefore, we have six contextual categories: EU - users from continent Europe, AS - users from continent Asia, AF - users from continent Africa, NA - users from continent North America, SA - users from continentSouth America, OC - users from continent Oceania. We derive six separate predictive models for each continent: $\{M_{c_i}\}_{i=1}^{6}$ that is trained for each continent and one global prediction model M that is trained on whole dataset as output of training stage.

The resulted accuracy is shown in Table I. Clearly, the user's geographical location is not an useful context according *Definition* 2. Because the way as the context divides data does not give us any benefits in terms of $Acc[T, \{M_i\}_{i=1}^{6}]$ that is accuracy. The related improvements of the predictive accuracy are almost always negative. Only for the case of PST location context gives slightly improvement. Therefore, the geographical location is not a useful context for our domain in this particular use-case.

TABLE I. Average accuracies (± standard deviation) of user intent prediction with the global Markov and local ("location" context) Markov models. "Glob." - global model accuracy, "W.Sum" - weighted sum of local model accuracies (Equation 5), "RI" - relative improvement compared to the global models.

Cat. c_i	Size c_i	FOMM(%)	CTW(%)	PST(%)
Glob.	1	40.6±0.3	49.2±4.3	45.3±0.2
EU	0.45	45.0±0.4	48.3±4.4	47.3± 3.9
AS	0.27	38.9±0.4	47.4±4.1	44.4± 3.3
AF	0.08	34.4±0.7	48.5±3.2	48.4± 3.2
NA	0.16	35.8±0.8	48.3±5.2	49.1± 4.9
SA	0.02	41.7±1.7	48.1±1.6	50.2±4.1
OC	0.01	46.8±2.8	45.2±6.4	49.4±9.1
W.Sum	1	40.1±0.4	48.3±2.6	46.1±1.4
RI	-	**-1.2**	**-1.8**	**+1.8**

B. Context by Community Detection

In this experiment we want to evaluate an impact of the context based on discovered communities in the user navigation graph. The method to obtain the context based on type of users' behaviour is described in details in Section V-B. By applying this method we obtain two communities in our users' navigation graph with modularity equals to 0.174.

M is a global model that built on whole Tr. We use alignment method and as a result we obtain two clusters: V_{exper} and V_{novice}. These clusters are used to learn two contextual models: M_{expert} and M_{novice}. The resulted accuracy is shown in Table II. The related improvement compared to performance of global model is high, up to 18.9% in terms of PST predictor. Distinctly, the type of users' behaviour is a useful context according to *Definition* 2. Since this context gives improvement in terms of $Acc[T, M_1, M_2]$, for all given predicting models: for FOMM relative improvement is 6.9%, for CTW relative improvement is 10.6%, and for PST relative improvement is 18.9%. According to the predictive accuracy it is important to notice that we have much higher relative improvement for the "advanced" users which are our target group from our business perspectives. This group of users has longer sessions which again indicates their interest to find a suitable program. Therefore the type of user's behaviour is a useful context for our domain in particular use-case of users' trail prediction.

Therefore, the proposed technique to discover *useful contexts* that can be used to improve the prediction models for the user navigation *trails*.

TABLE II. Average accuracies (± standard deviation) of user intent prediction with the global Markov and local ("user type" context) Markov models. Relative improvement compared to the global model ("Glob.") is given in bold in the round brackets. "W.Sum" is weighted sum of the local model accuracies (Equation 5).

Cat. c_i	Size c_i	FOMM(%)	CTW (%)	PST (%)
Glob.	1	40.6±0.3	49.2±4.3	45.3±0.2
"expert"	0.11	55.3±0.9 **(+36.2)**	59.3±3.1 **(+20.5)**	60.7±1.8 **(+34.0)**
"novice"	0.89	43.4±0.3 **(+6.9)**	53.2±1.9 **(+8.3)**	53.1±2.9 **(+17.2)**
W.Sum	1	43.4±0.28 **(+6.9)**	54.4±1.7 **(+10.6)**	53.9±2.7 **(+18.9)**

C. Random Context

We introduce a random context R in order to provide a support evidence for presented theory about contextual Markov

models. In particular, we aim to provide an experimental argument that local Markov models are not worse than global Models.

We select randomly training samples of different size. Assume that random context have two categories, so $k = 2$. Therefore, we divide randomly Tr into two samples $(Tr/2)_1$ and $(Tr/2)_2$ and build local models $M_{Tr/2_1}$ and $M_{Tr/2_2}$ respectively. An alignment function randomly selects model M_{Tr/n_i} for a test instance. Then the expected accuracy $Acc[T, M_{Tr/2_1}, M_{Tr/2_2}]$ is calculated (Equation 6). We continue the experiment recursively splitting the training data until the size of Tr/n becomes less than 100 sessions. We run the experiment 10 times and compute averages and standard deviations of generalization accuracies.

The results are presented in Figure 3. Blue plot "Weighed random context" shows accuracy of local models of different size. Figure 3 (B) presents results for PTS predictor. We can clearly see when the training size becomes less than 4k, the standard error increases substantially and the accuracy declines. The same situation happens with CTW predictor in Figure 3 (A) - accuracy drops when training size is less than 4k instances. Both predictors show the same tendency - the accuracy decreases when the size of the sampled training subset is less than 10-20% of the whole set, and an increase of the standard error testifies about future reduction of accuracy (or future unexpected behaviour). Figure 3 also depicts accuracies and the corresponding standard errors of the global model and considered contexts: geographical location and users' behaviour type. Based on the observations we can hypothesize that if standard error is low then the discovered cluster is strong.

VIII. Conclusion

In practice, domain experts can have many ideas about possible context for the domain, based on their intuition; they can apply explorative analysis and data mining techniques to identify the contextual features.

In this paper, we introduced a formal definition of *useful context* and the problem of learning *contextual Markov models*. We formulated the context discovery as an optimization problem. We provided intuitive proofs showing that an optimal global model corresponds to optional contextual models and for Markov models, the contextual models are expected to be at least as good as global. We performed experiment with random contextual Markov models and it shows that with some constraints they are almost as good as global. Therefore, this fact gives us experimental evidence about safety of testing local models. Thus, at least for this class of models we have a sound justification and motivation for context-aware predictive analytics.

We introduced the method to context discovery which consists of two important components: a clustering algorithm which divides training sequences into k groups and an alignment method which assigns new test sequence to clusters. We presented the experiments with real-world dataset for two specific examples of our method: (1) explicit contexts of users' location which is widely used in many applications and (2) implicit context that is inferred from the user navigation graph with our approach.

The experimental case study on the real dataset that we performed can be regarded as an illustration of contextual Markov models learning. This case study shows that if we can identify useful contexts the local Markov models outperform the single global Markov, and if context is not useful, local models will still perform as good as the global model.

Acknowledgements

This research has been partly supported by STW CAPA and NWO COMPASS projects. The experimentation was partly carried out on the National e-infrastructure with the support of SURF Foundation. We would like to thank Thijs Putman from *StudyPortals.eu* for providing the anonymized dataset.

References

[1] A. O. Alves and F. C.Pereira. Making sense of location context. 2012.

[2] R. Begleiter, R. El-Yaniv, and G. Yona. On prediction using variable order markov models. *Journal of Artificial Intelligence Research (JAIR)*, 22:385–421, 2004.

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 10, 2008.

[4] 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.

[5] P. Brown, J. Bovey, and X. Chen. Context-aware applications: From the laboratory to the marketplace. *IEEE Personal Comm*, 4:58–64, 1997.

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

[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] J. Kiseleva, H. T. Lam, M. Pechenizkiy, and T. Calders. Discovering temporal hidden contexts in web sessions for user trail prediction. In *Proceedings of the 22nd international conference on World Wide Web, (Companion Volume, TempWeb@WWW'2013)*, pages 1067–1074. ACM, 2013.

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

[13] D. Ron, Y. Singer, and N. Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning (ML)*, 25(2-3):117–149, 1996.

[14] A. Schmidt, M. Beigl, and H.-W. Gellersen. There is more to context than location. *Computers & Graphics*, 23(6):893–901, 1999.

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

[16] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Trans. Inf. Syst. (TOIS)*, 10(1):91–202, 1992.

[17] F. M. J. Willems. The context-tree weighting method : Extensions. *IEEE Transactions on Information Theory (TIT)*, 44(2):792–798, 1998.

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

[19] I. Zliobaite, J. Bakker, and M. Pechenizkiy:. Towards context aware food sales prediction. In *ICDM Workshops*, pages 94–99, 2009.

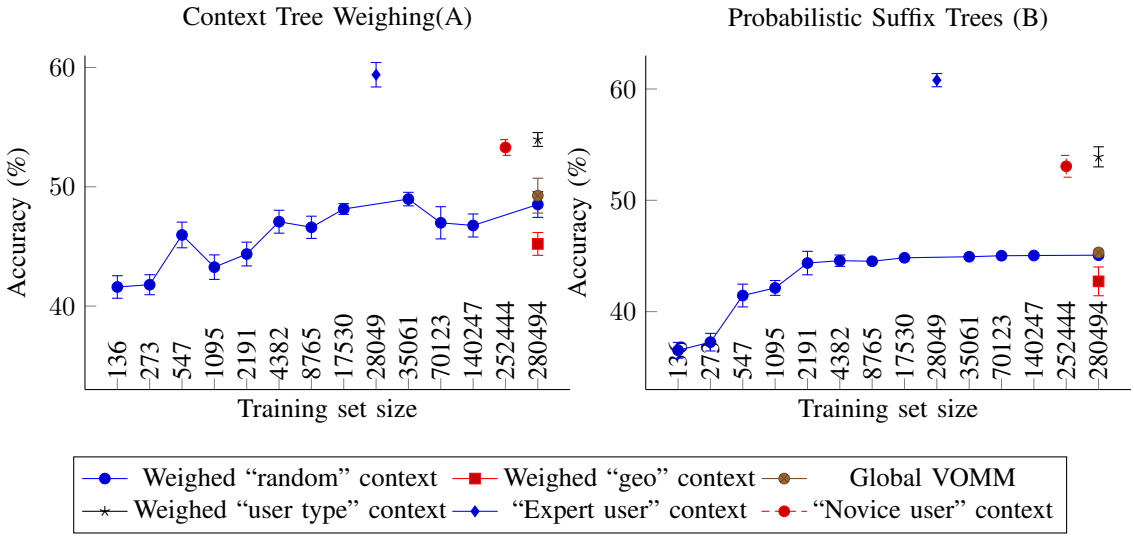

Fig. 3. Mean of accuracy for 10 iterations with standard error (SE). Plot (A) represents results for CTW algorithm. Plot (B) represents results for PST algorithm.