Factored MDPs for Detecting Topics of User Sessions

Maryam Tavakol[†]

[†]Technische Universität Darmstadt
Department of Computer Science
Darmstadt, Germany
tavakol@kma.informatik.tu-darmstadt.de

Ulf Brefeld[†‡]

[‡]German Institute for Educational Research
Information Center for Education
Frankfurt, Germany
brefeld@kma.informatik.tu-darmstadt.de

ABSTRACT

Recommender systems aim to capture interests of users to provide tailored recommendations. User interests are however often unique and depend on many unobservable factors including a user's mood and the local weather. We take a contextual session-based approach and propose a sequential framework using factored Markov decision processes (fMDPs) to detect the user's goal (the topic) of a session. We show that an independence assumption on the attributes of items leads to a set of independent models that can be optimised efficiently. Our approach results in interpretable topics that can be effectively turned into recommendations. Empirical results on a real world click log from a large e-commerce company exhibit highly accurate topic prediction rates of about 90%. Translating our approach into a topic-driven recommender system outperforms several baseline competitors.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

MDP; recommender systems; session-based; user intent

1. INTRODUCTION

Recommender systems are designed to satisfy user's information and tangible needs. Guessing the intention of users is not only fundamental for the overall user experience but directly linked to revenue. User intent however is driven by unobservable internal (e.g., mood, spontaneous inspiration) as well as external (e.g., weather, location) processes [15]. Capturing user intent is therefore one of the most challenging problems in many retrieval and recommendation tasks.

The context of a user is often seen as a proxy for the unobserved processes [34]. Context may be provided by previously visited pages [8], viewed items [28], or user profiles [11], and is often studied together with personalisation [20]. There is a broad range of applications using contextual variables of users including query refinement [27], re-ranking for web search [12], market segmentation [14], and latent variable models for spoken language understanding [9]. An alternative approach to capturing the intent of users are topic models. Topic models [5] can be seen as a generative probabilistic semantics that have been proposed for retrieval [31] as well as recommender systems [23, 10]. Topic models for temporal data however often require stationary data distributions [4, 1], an assumption too restrictive for highly dynamic scenarios such as e-commerce.

In this paper, we focus on recommender systems where user feedback is recorded implicitly, for instance through clicks on result pages of search engines or on lists of recommended items. The implicit feedback can be used to train autonomous recommender systems as the noisy and incomplete batch of user responses provides a partial labelling of the data. Note that these partial labels do not suffice for purely supervised approaches as the outcome of recommending alternative items is undefined. On the other hand, the task neither fits a purely unsupervised setting as the valuable (partial) ground-truth would be discarded. The abstract problem setting matches that of reinforcement learning-style approaches where uncertainty about the value of actions (e.g., recommending an item) is minimised by trading off exploration and exploitation [28, 18]. Reinforcement learning-based approaches are naturally sequential models with intrinsic Markov assumptions that allow for capturing the context of a user by explicitly representing sequences of previously clicked items [28, 35].

We study factored Markov decision processes (fMDPs) [6] to detect topics of user sessions. We take a sequential approach and leverage ideas from [35] and [28] to characterise sessions in terms of the history of viewed items. However, straight forwardly solving the resulting fMDPs is infeasible due to exponentially increasing state spaces. Moreover, the structure of the value function does not necessarily retain the structure of the process after factorisation [17]. Hence, many approaches to approximate value functions have been proposed (e.g., [13, 7]).

Commencing with a standard fMDP on the history of viewed items, our main contributions are as follows. We show that an independence assumption on the attributes of items allows to equivalently represent the fMDP by an ensemble of independent fMDPs. Compared to the initial fMDP, the resulting state space is orders of magnitude smaller and the ensemble can be optimised efficiently. In addition, we propose a robust approximation following ideas from Shani et al. [28] to improve the predictive accuracy in the presence of data sparsity and large-scale applications. We show that the learned Q-values can be easily turned into interpretable topics and recommendations.

In extensive experiments on real world data at enterprise-level scales provided by Zalando, we observe highly accurate topic de-

RecSys'14, October 6–10, 2014, Foster City, Silicon Valley, CA, USA.

ACM 978-1-4503-2668-1/14/10 ...$15.00.
http://dx.doi.org/10.1145/2645710.2645739.

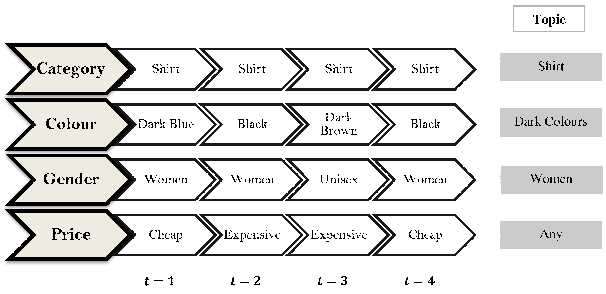

Figure 1: An exemplarily user session of an e-commerce platform. The session is viewed as a (possibly independent) sequences of item attributes. The right hand side shows the topic of the session.

tection rates of about 90%. We show that these topics can effectively be utilised to recommend items of interest with high accuracy. Translating our approach into a topic-driven recommender system outperforms collaborative baseline methods in terms of average rank.

The remainder of our paper is structured as follows. Section 2 motivates the problem setting and presents a running example that we will use throughout the paper. We introduce our technical contribution in Section 3 and present empirical results in Section 4. Section 5 briefly reviews related work on topic detection for sequential prediction problems including recommender systems and Section 6 concludes.

2. PRELIMINARIES

Traditional recommender systems and personalisation approaches are designed to capture long-term preferences of users. Prominent examples are movie [3] and job recommendations [21], conference paper assignments [25], or e-commerce application scenarios [19]. Short-term interests of users provide valuable additional information to improve the quality of recommendations. Short-term interests may rise from unexpected events (got an MP3 player, need headphones) spontaneous moods and ideas (split-up with partner, need action movie for distraction), or external sources (its freezing outside, need winter coat).

Figure 1 visualises the scenario using an e-commerce example. The figure shows an exemplary user session where a user views a series of garments. The first item is a *cheap dark blue shirt* followed by an *expensive black shirt* and so on. Instead of addressing the items and their attributes jointly, we treat their features as independent. We thus focus on sequences *shirt, shirt, shirt, shirt* for the attribute *category* or *dark blue, black, dark brown, black* for *colour*, respectively. Every such sequence gives us an expectation about the value of the next item. For instance, the former (constant) sequence is likely extended by another *shirt* while the latter gives rise to a dark coloured item. The attribute *price* in Figure 1 constitutes a special case as the sequence does not allow to confine the possible values. Hence, *any* of the attribute values is possible or, in other words, the attribute *price* is not important for the user session.

Given the session in Figure 1, the user's goal is to find *dark coloured shirts for women of any price level*. We call the corresponding distribution of attribute values the *topic* of the session. For the attribute *colour*, we expect dark colours to be very likely while light colours are associated with small probabilities close to zero. Formally, a topic is defined as follows.

DEFINITION 1. *Let s be a (possibly ongoing) user session and $\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ be a set of random variables encoding attributes of items. The topic of a session s is defined as the distribution of attributes of the next item given by $P(\mathcal{X}_1 = x_1, \ldots, \mathcal{X}_n = x_n | s)$.*

Once the topic of a session has been detected, a recommender system could leverage the estimate to recommend items that lie in the very topic of the session. Note that the topic of a session is independent of other sessions of that user and its lifetime therefore well suited to adapt to short-term interests of users.

In this paper, we aim to accurately identify the topic of user sessions. We propose factored Markov decision processes (fMDPs) and define user sessions as sequences of viewed (clicked) items. The detection of topics is consequentially addressed as a sequential Markov decision problem. In the next section, we deploy reinforcement learning agents to detect the topic and to translate the estimate into topic-driven recommendations.

3. MDP-BASED TOPIC DETECTION

3.1 A Standard Approach

We are given a set of items $\mathcal{I}$ described by a set of n random variables $\mathcal{I} = (\mathcal{X}_1, \ldots, \mathcal{X}_n)$, where each $\mathcal{X}_j$ encodes an attribute (e.g., colour, category) and takes on values in a discrete and finite set $dom(\mathcal{X}_j)$. Every item $i \in \mathcal{I}$ is therefore defined by a set of n attributes $i = (x_1^i, \ldots, x_n^i)$ with $x_j \in dom(\mathcal{X}_j)$ for $1 \leq j \leq n$. Items are completely characterised by their attributes, that is, the existence/probability of an item is equivalent to the existence/probability of its combination of attributes. Assuming for instance that item i has attributes $i = (x_1^i, \ldots, x_n^i)$ it holds

$$\Pr(\mathcal{I} = i) = \Pr(\mathcal{X}_1 = x_1^i, \ldots, \mathcal{X}_n = x_n^i). \quad (1)$$

To avoid cluttering the notation unnecessarily, we omit the superscript i in the remainder.

An MDP is a four-tuple $(\mathcal{S}, \mathcal{A}, R, P)$ consisting of the set of states $\mathcal{S}$, the set of actions $\mathcal{A}$, a reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and a transition function $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ representing the dynamics of the environment. The goal of MDP is to find the optimal value function $V : \mathcal{S} \to \mathbb{R}$ for the state space [29].

In a straight forward sequential MDP for contextual item recommendation, the set of states $\mathcal{S}$ is defined as the Kleene closure of the set of items, that is $\mathcal{S} = \mathcal{I}^*$. The set $\mathcal{S}$ thus contains all possible sequences of items and every element $s \in \mathcal{S}$ can be identified with a (possibly unfinished) user session in terms of the viewed items. An action $a \in \mathcal{A}$ corresponds to recommending a particular item from $\mathcal{I}$, so that we may identify $\mathcal{A} = \mathcal{I}$ and consequentially for every $i \in \mathcal{I}$ there exists an $a \in \mathcal{A}$ such that $i = a$ and vice versa.

The described MDP is trivially infeasible due to the infinite number of states $\mathcal{S}$. Though in practice not all possible sequences will actually be observed and an additionally incorporated Markov assumption may further reduce the state space, the model remains intractable even for small and medium-sized ranges of items.

3.2 Factorisation

We therefore take a different approach and define the MDP over the set of attributes $\mathcal{X}_1, \ldots, \mathcal{X}_n$ instead of the items $\mathcal{I}$. Due to Equation (1), we obtain an equivalent factored MDP (fMDP) where the set of states is given by the Kleene closure $\mathcal{S} = (\mathcal{X}_1, \ldots, \mathcal{X}_n)^*$. An element $s \in \mathcal{S}_j$ corresponds to a sequence of realisations of the j-th attribute. Consequentially, the factorisation also impacts the set of actions which is now given by $\mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_n$ with $\mathcal{A}_j = dom(\mathcal{X}_j)$ for all j. We use $a_j \in \mathcal{A}_j$ and $x_j \in \mathcal{X}_j$ interchangeably in the remainder for convenience.

The reward after taking action a (i.e., recommending the corresponding item i) in state s is given by the reward function $R(s, a)$.

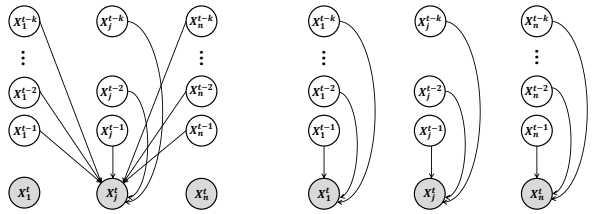

Figure 2: Left: Transition model of a joint factored MDP. Every attribute value depends on the complete history of all previously viewed items and their attributes which leads to an infeasible model. Right: Sequences of attributes form independent components. There are no dependencies between attributes.

Positive rewards indicate a click on a recommended item in which case the recommendation was successful and has been accepted by the user. The transition function $P(s'|s,a)$ estimates the probability of entering state s' after recommending a in state s. Note that s serves as a prefix of s' which is given by $s' = s \circ i'$, where i' is the clicked item by the user and $\circ$ the operator that appends two sequences, e.g., $q \circ p = pq$.

The length of the actual state s is continuously increased by appending clicked items, exactly one at a time. Thus, instead of addressing the complex $P(s'|s,a)$, transition probabilities $P(i'|s,a)$ are often used as an equivalent proxy due to their simpler structure. The quantity $P(i'|s,a)$ is the transition probability of clicking on item i' when in state s and recommending item a. The transition probabilities can be represented as a two-layer acyclic graph that connects the attributes of the previously viewed items in s with the attributes of the item to be clicked denoted by i'. Theoretically, the joint transition probability can be efficiently computed by factorising conditional probabilities, e.g.,

$$\Pr(\mathcal{X}'_1, ..., \mathcal{X}'_n|s,a) = \prod_{j=1}^{n} \Pr(\mathcal{X}'_j|parents(\mathcal{X}'_j), a),$$

where $parents(\mathcal{X}'_j)$ denotes the parents of the node $\mathcal{X}'_j$ in the underlying graphical model. However, the state space of the fMDP grows exponentially and renders practical application infeasible as the exact estimation of the optimal policy is not feasible due to the curse of dimensionality [13]. Thus, we haven't won anything yet in terms of feasibility but successfully rephrased the model over attribute sequences of the viewed items (Figure 2 left).

3.3 Exploiting Independence

Directly addressing the joint $\Pr(\mathcal{X}_1, \ldots, \mathcal{X}_n|s,a)$ requires a state space that is intractable even for small and medium-sized warehouses. We therefore treat the attributes of the items as independent and approximate the intractable joint by a product of independent decisions,

$$\Pr(\mathcal{X}_1, \ldots, \mathcal{X}_n|s,a) \approx \prod_{j=1}^{n} \Pr(\mathcal{X}_j|s_j, a_j).$$

The idea is to split the fMDP into an ensemble of n disjoint and independent fMDPs, one for each attribute. The j-th fMDP focuses on only the j-th attribute and recommends a realisation a_j of $\mathcal{X}_j$ based on the sequence of attributes s_j of the previously viewed items. In Figure 1 for instance, guessing that the next item will be another *shirt* can trivially be done in the absence of all other attributes. A similar argument holds for expecting a *dark colour* or a garment for *women*.

The following theorem shows that an fMDP with independent chains of random variables admits an equivalent representation as an ensemble of n independent fMDPs. In order to propagate single receiving reward through all fMDPs, we consequentially assume additive factorised rewards $R(x,a) = \sum_{j=1}^{n} R_j(x_j, a_j)$.

THEOREM 1. *A factored MDP with a set of n independent components $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ allows an equivalent representation as an ensemble of n independent fMDPs, one for each component $\mathcal{X}_j$ where $1 \leq j \leq n$. Let $V^*(x)$ be the optimal value for state $\mathcal{X} = x$ in the joint fMDP and $V^*(x_j)$ be the optimal value for attribute $\mathcal{X}_j = x_j$ in the j-th fMDP, for all $1 \leq j \leq n$. It holds*

$$V^*(x) = \sum_{j=1}^{n} V^*(x_j).$$

PROOF. The standard update rule of value iteration is given by

$$V^{N+1}(x) = \max_a [R(x,a) + \gamma \sum\nolimits_{x'} P(x'|x,a) V^N(x')] [29].$$

Replacing the maximum operator by a softmax gives

$$V^{N+1}(x) = \frac{1}{\rho} \log \sum_a \exp[\rho R(x,a) + \gamma\rho \sum_{x'} P(x'|x,a) V^N(x')],$$

where ρ controls the degree of the approximation and the exact maximum is recovered for $\rho \to \infty$. We show the claim by induction for value iteration. For $N = 1$, we have

$$V^1(x_j) = \frac{1}{\rho} \log \sum_{a_j} \exp[\rho R_j(x_j, a_j)]$$

for the j-th fMDP of the ensemble and the joint is obtained by

$$\begin{aligned} V^1(x) &= \frac{1}{\rho} \log \sum_a \exp[\rho R(x,a)] \\ &= \frac{1}{\rho} \log \sum_{a_1} [\sum_{a_2} \ldots [\sum_{a_n} \prod_j \exp\{\rho R_j(x_j, a_j)\}]] \end{aligned}$$

The innermost summation can be rewritten as

$$\begin{aligned} &\sum_{a_n} \Big[\exp\{\rho R_1(x_1, a_1)\} \times \cdots \times \exp\{\rho R_n(x_n, a_n)\} \Big] \\ &= \exp\{\rho R_1(x_1, a_1)\} \times \cdots \times \exp\{\rho R_{n-1}(x_{n-1}, a_{n-1})\} \\ &\quad \times \sum_{a_n} \exp\{\rho R_n(x_n, a_n)\} \end{aligned}$$

by drawing unrelated terms out of the sum. Continuing for the other summations gives

$$\begin{aligned} V^1(x) &= \frac{1}{\rho} \log \Big[\prod_j \sum_{a_j} \exp\{\rho R_j(x_j, a_j)\} \Big] \\ &= \frac{1}{\rho} \sum_j \log \sum_{a_j} \exp[\rho R_j(x_j, a_j)] = \sum_j V^1(x_j), \end{aligned}$$

which shows the claim for $N = 1$. Now assume that $V^N(x) = \sum_{j=1}^{n} V^N(x_j)$ holds for all x. For the joint fMDP, the second summand in the exponent is simplified by

$$\begin{aligned} &\sum_{x'} P(x'|x,a) V^N(x') = \sum_{x'} P(x'_1|x_1,a_1) \cdots P(x'_n|x_n,a_n) V^N(x') \\ &= \sum_{x'} P(x'_1|x_1,a_1) \cdots P(x'_n|x_n,a_n) [V^N(x'_1) + \cdots + V^N(x'_n)], \end{aligned}$$

where the latter gives rise to the telescope sum

$$\sum_{x'_1} P(x'_1|x_1,a_1)\Big[\sum_{x'_2} P(x'_2|x_2,a_2)\Big[\times\cdots \\ \cdots\times\Big[\sum_{x'_n} P(x'_n|x_n,a_n)\Big[V^N(x'_1)+...+V^N(x'_n)\Big]\Big]\Big]\Big].$$

The innermost summation over the new state x'_n yields

$$V^N(x'_1)\sum_{x'_n} P(x'_n|x_n,a_n)+\cdots+\sum_{x'_n} P(x'_n|x_n,a_n)V^N(x'_n)$$

and since $\sum_{x'_n} P(x'_n|x_n,a_n)=1$ we obtain

$$V^N(x'_1)+...+V^N(x'_{n-1})+\sum\nolimits_{x'_n} P(x'_n|x_n,a_n)V^N(x'_n).$$

Drawing out the remaining terms from unrelated summations and putting things together gives

$$V^{N+1}(x)=\frac{1}{\rho}\log\sum_{a_1}\Big[\cdots\Big[\sum_{a_n}\prod_j \exp[\rho\{R_j(x_j,a_j) \\ +\gamma\sum_{x'_j} P(x'_j|x_j,a_j)V^N(x'_j)\}]\Big]\Big].$$

Reordering terms shows the claim. □

Theorem 1 shows that any high dimensional fMDP with independent attributes can be equivalently expressed by several independent fMDPs. Exploiting the independence between the attributes, the resulting ensemble consists of an fMDP for every component. The resulting state spaces are independent sequences over a single attribute given by the Kleene closure $\mathcal{S}_j=(dom(\mathcal{X}_j))^*$ for all components j. Note that a result by [17] shows that the value function of fMDPs does in general not retain the structure of the process. Our theorem proves that a structured value function is generally obtainable for fMDPs with independent components.

Still, a major drawback of the model is the dependence on the whole session, that is, every viewed item impacts all subsequent actions. We therefore take a k-th order Markov assumption to represent only the k most recently viewed items explicitly. The set of states of the j-th fMDP is effectively reduced to $\mathcal{S}_j=(dom(\mathcal{X}_j))^k$. The Markov assumption discards long-range dependencies and lead, together with the previous independence assumption, to an efficient and compact representation of the ensemble as shown in Figure 2 (right).

3.4 Optimisation

The resulting independent fMDPs can be optimised independently and in parallel using standard reinforcement learning techniques such as value iteration. Value iteration learns the state-value function, $V(s)$, using the model of the environment; the reward and transition functions $R(s,a)$ and $P(s'|s,a)$, and converges to the optimal solution in a discounted finite MDP [29].

The set of states in the j-th fMDP is described by a k-sequence of realisations of the j-th attribute $\mathcal{X}_j$ given by $s_j=(x_j^{t-k},\ldots,x_j^t)$. The task of the agent is to predict the value of action $a_j\in dom(\mathcal{X}_j)$ in the actual state s_j. The transition function P encodes the probability of observing the subsequent state $s'_j=(x_j^{t-k+1},\ldots,x_j^{t+1})$ and the reward function R_j provides feedback for recommending a_j in s_j. Value iteration uses the following update rule for value determination,

$$V^{N+1}(s_j)=\max_{a_j}\Big[R_j(x_j,a_j)+\gamma\sum\nolimits_{s'_j} P(s'_j|s_j,a_j)V^N(s'_j)\Big].$$

When the value function converges to the optimal V^*, state-action values $Q(s_j,a_j)$ can be derived

$$Q(s_j,a_j)=R(s_j,a_j)+\gamma\sum\nolimits_{s'_j} P(s'_j|s_j,a_j)V^*(s'_j),$$

where $Q(s_j,a_j)$ measures the quality of recommending a_j in state s_j. Realisations with high Q-values are likely to be observed in the next page view while small Q-values indicate very unlikely observations. We use the terms $Q(s_j,a_j)$ and $Q(s_j,x_j)$ interchangeably in the remainder.

Reinforcement learning techniques often perform poorly in large scale problems due to slow convergence rates. Adapting the model to data is therefore performed in two steps; offline and online. First, an initial model is learned by value iteration where transition and reward functions are adapted to historic data by maximum likelihood. The trained model is then deployed in an online scenario where it is gradually updated according to the user feedback to improve estimations. In practice, value iteration can be repeated periodically (e.g., once in a week) to keep the system up to date.

3.5 Approximation

In practical applications, the available data is often too sparse to allow for an accurate estimation of the transition probabilities. In addition, applications on large-scales render keeping the whole set of transition probabilities infeasible due to memory requirements. We thus propose an efficient approximation of our model based on the ideas of Shani et al. [28].

The main idea is to focus on estimating the probability $\Pr(i'|s)$ of item i' to be clicked next, irrespectively of the action. The transition $\Pr(i'|s,a)$ can be approximatively reconstructed from $\Pr(i'|s)$ as follows. Recall that action a is identical to an item $i\in\mathcal{I}$. There are three possible outcomes of taking action $i=x$ when in state s: (i) The user accepts the recommendation i with probability $P(i|s,i)$, (ii) she rejects i and clicks instead on item i' with probability $P(i'|s,i)$, or (iii) the session terminates with probability $P(\emptyset|s,i)$. Consider the former two events. The task is to estimate $P(i|s,i)$ and $P(i'|s,i)$ as a surrogate for the entire transition function. Note that in the latter, a click on i' is independent of the recommended item i.

The assumption is that the probability of clicking on a recommended item is larger than the probability of choosing the item in the absence of a recommendation, that is $P(i|s,i)\geq P(i|s)$ [28]. Analogously, the probability of clicking on item i in the absence of any recommendation is higher than for clicking on i when the recommended item is actually $i'\neq i$, that is $P(i|s,i')\leq P(i|s)$. By choosing appropriate constants $\alpha>1$ and $0<\beta<1$, the desired quantities are approximated by $[P(i|s,i)\approx\alpha P(i|s)$ and $P(i|s,i')\approx\beta P(i|s)$, subject to $P(i|s,i)+\sum_{i'\neq i}P(i'|s,i)+P(\emptyset|s,i)=1$, which is obtained by normalisation.

3.6 Topic Extraction

Once approximate or exact Q-values $Q(s_j,x_j)$ are computed, they can be used to extract the topic of the session as follows. The value $Q(s_j,x_j)$ is proportional to the probability that the user clicks on an item with attribute x_j given the sequence of realisations s_j. In other words, realisations with high Q-values are likely observed next and thus constitute a part of the topic of s_j. For uniformly distributed Q-values, e.g., $Q(s_j,x_j)\approx Q(s_j,x'_j)$ for all x_j,x'_j, the topic contains the whole domain $dom(\mathcal{X}_j)$, indicating that the j-th attribute does not contribute to the topic. As a consequence, any realisation of that attribute may be observed next. Intermediate Q-values are ranked according to their difference to the maximum Q-value, such that the expected realisations of attribute

j are computed by the min-max normalisation

$$q(\mathcal{X}_j = x_j|s_j) = \frac{Q(s_j, x_j) - \min_{x'_j}[Q(s_j, x'_j)]}{\max_{x'_j}[Q(s_j, x'_j)] - \min_{x'_j}[Q(s_j, x'_j)]}, \quad (2)$$

for all $1 \leq j \leq n$. The independent results are then multiplicatively combined to approximate the desired probabilities

$$P(x_1, \ldots, x_n|s) \propto \prod_{j=1}^{n} q(x_j|s_j).$$

3.7 Recommendation

Our approach can also be turned into a recommender system. In contrast to the topic extraction, we use a softmax instead of the min-max normalisation to translate Q-values into probabilities,

$$\Pr(\mathcal{X}_j = x_j|s_j) = \frac{\exp\{Q(s_j, x_j)\}}{\sum_{x'_j} \exp\{Q(s_j, x'_j\}}. \quad (3)$$

The softmax gives us a probability distribution over the state space of every attribute. The use of the exponential function penalises even small differences and thus acts like a probabilistic winner-takes-all. Note that in practice, recommendations have to be computed very efficiently under rigid time constraints. Having a clear set of winners helps to speed-up the computation by continuously filtering out items at early stages that cannot make it into the top-m to save time for more promising candidates.

Given the estimates in Equation (3), the score for item i with attribute combination $x_1, \ldots, x_n$ is simply given by the product of the corresponding probabilities, or alternatively, by the sum of the corresponding log-probabilities, that is,

$$score(i; s) = \prod_{j=1}^{n} P(\mathcal{X}_j = x_j|s_j) \propto \sum_{j=1}^{n} \log P(\mathcal{X}_j = x_j|s_j).$$

The scores impose a ranking on the items and the top-scoring products can be recommended.

4. EMPIRICAL EVALUATION

In this section, we evaluate our approach on an anonymised click log from Zalando[1], a large European online fashion retailer. The data distribution is modified so that no conclusions on customer data or business figures of the company can be drawn. There are $1,721,483$ user sessions consisting of $24,353,852$ clicks in total. Sessions are split after 25 minutes idle time and the average session consists of 14 clicks. Every click is associated with a timestamp, the attributes of the viewed item, user ID, and the recommended items. We focus on attributes colour, gender, category, and price. There are 62 different colours, 16 genders (including types of accessories), 61 categories, and 16 discrete levels of price in the log.

4.1 Small-scale Topic Detection

Measuring the performance of topic detection methods using real world data is difficult as topics are not observed variables but contained only implicitly in the data. We therefore test the topic prediction against the attribute values of the next clicked item. We translate the distribution in Equation (2) into a discrete set of attribute values. A simple thresholding approach discards unlikely realisations and returns a set T_j for every attribute $1 \leq j \leq n$ given a session $s = (s_1, \ldots, s_n)$,

$$T_j(s_j) = \{x_j | x_j \in dom(\mathcal{X}_j) \wedge q(\mathcal{X}_j = x_j|s_j) > c\}$$

where c is a user defined constant. Large values of c thin out the topic and focus on highly probable attribute values. On the other hand, small values of c weaken the interpretability and usability of the resulting topics unnecessarily that may contain many unlikely realisations. In the first set of experiments, we use $c = \frac{1}{2}$ and study variations of the parameter afterwards. The joint topic $T(s)$ is then given as the union over all attributes by $T(s) = \bigcup_{j=1}^{n} T_j(s_j)$.

We evaluate the accuracy of the extracted topics for every attribute as well as for the joint topic using indicator functions $[[z]]$ yielding one if the argument z is true and 0 otherwise. Let $T_j(s_j)$ be the topic of an ongoing session and x'_j the corresponding realisation of the next clicked item. The topic prediction is correct if $[[x'_j \in T_j(s_j)]]$. The joint topic is then evaluated by concatenating the individual results with an and-operator,

$$acc(T, s, i') = \bigwedge_{j=1}^{n} [[x'_j \in T_j(s_j)]].$$

Note that high accuracies in individual attributes do not necessarily indicate a good joint performance as the all attribute values need to be contained in the topic.

We compare the ensemble approach of Section 3.4 (M1) with its approximation in Section 3.5 (M2). As a baseline, we deploy a simple Markov process (MP) that uses estimates $P(i'|s)$ directly instead of $Q(s, i')$ for the computation of the topic. Thus, its probabilities are proportional to the number of times that item i' has been clicked in state s estimated by maximum likelihood. Additionally, we include LDA [5] as another baseline. To this end, every session is treated a document where the attributes of the viewed items are considered the words of the document. The set of words is thus defined by $\bigcup_{j=1}^{n} dom(\mathcal{X}_j)$ and contains 155 distinct words. We apply the method by [5] for both estimation and inference of topic proportions as well as word distribution per topic. At testing time, LDA determines the topic mixture of the ongoing session based and computes the probability distribution of attributes according to the mixture. Thresholding is identical for all methods.

For the first set of experiments, we only use a subset of the data for evaluation as the exact variant cannot be evaluated on all available data due to memory issues. In the corresponding subset, there are $34,343$ user sessions consisting of $722,179$ clicks in total with the average of 21 clicks per session. We split 70% of the resulting sessions for training, 20% as holdout, and 10% as test sessions according to the temporal nature of the data. Optimal parameters for M2 and LDA are found by model selection and are given by $\alpha = 2$, $\beta = 0.001$ for M2 and $\alpha_{LDA} = 0.1$ and 100 topics for LDA, respectively. Rewards are positive for clicks on recommended items as well as adding to cart, and sale actions. Removing items from the cart is penalised with negative rewards, all other actions realise a reward of zero.

Table 1 shows average accuracies of the best models for Markov assumptions of order $k \in \{1, 2, 3, 4\}$ and LDA. The exact ensemble M1 performs poorly for short histories but improves significantly for larger k. We credit this finding to the necessity of taking chains of consecutive clicks into account. Although the individual predictions on attribute levels are promising, the joint topic is not well captured. Further, the high sparsity of small data sample leads to the predictive accuracy of below 70%. By contrast, the approximate M2 performs much better for short histories and detects the correct topic in 94% of the cases for $k = 1$. The performance decreases for longer chains. The observed effect originates from the approximation itself. The data sample is not sufficiently large for reliably approximating longer histories. We will address this issue in the next experiment again.

[1] www.zalando.com

Table 1: Accuracies for the topic detection on a subset.

	k	joint	colour	gender	category	price
MP	4	33.69	49.78	92.24	78.52	63.96
	3	37.70	52.98	92.31	79.50	65.06
	2	37.65	52.15	92.22	79.68	64.24
	1	28.06	44.31	91.85	79.01	56.28
M1	4	67.53	85.61	95.00	90.70	78.68
	3	69.56	93.94	95.21	93.36	72.01
	2	40.62	45.96	95.30	94.90	78.39
	1	16.47	28.37	95.31	95.28	46.55
M2	4	75.33	81.92	94.65	90.05	92.38
	3	89.52	92.95	94.83	92.81	94.48
	2	93.69	95.12	94.97	94.45	95.00
	1	94.14	95.25	94.98	94.82	94.97
LDA	-	1.65	11.76	85.89	52.8	21.14

LDA characterises the next click by dominant attributes of the ongoing session. The results show that users tend to click on items with so far unseen attribute values, particularly for price and colour. However, apart from M2, the joint topics are mostly inaccurate and do not reflect the performance for individual attributes. The outcomes of MP show that simply counting frequencies of subsequent events is not sufficient for achieving state-of-the-art performances.

4.2 Large-scale Topic Detection

In this section, we focus on the approximate ensemble (M2) and repeat the previous experiment on the whole click log. We split all available data into consecutive training (70%), holdout (20%), and test (10%) sets to preserve the temporal nature of the data. Table 2 shows the results for MP and the approximate M2 for histories of size $k \in \{1, 2, 3, 4\}$ as well as LDA. All three methods exploit the abundance of data and improve their performance. However, the overall joint performance of LDA is still far from a real-world deployment and even for MP still stays constantly below 40%. The approximate M2 clearly outperforms the baselines and yields impressive joint accuracies of about 90% for all k. The additional data trades-off the approximation issues observed in the previous experiment for larger k at the expense of smaller k.

4.3 Analysing the Topics

In this experiment, we study the variation of the detected topics in the course of the sessions using the experimental setup of Section 4.2. We measure the difference of subsequent topics $T(s)$ and $T(s')$ by their Jaccard distance. A large distance indicates rapid changes in neighbouring topics and either refers to a badly adapted model or to undetermined users who are just browsing instead of following a specific goal. By contrast, small distances indicate that users are very predictable and only search for very particular items without digressing.

Figure 3 (left) depicts the session on the x-axis and the variation of neighbouring topics in terms of their Jaccard distance on the y-axis averaged over all sessions. Unsurprisingly, for all histories $1 \leq k \leq 4$, the variation decreases rapidly after a few clicks. The more clicks a user performs, the more feedback is provided to the system and can be exploited by the model. Except for histories of length $k = 4$, all models converge quickly to only a few variations. That is, only very few attribute values are replaced between time steps. For longer histories $k = 4$, we observe more variations which is also reflected by lower overall accuracies in Table 2.

Figure 3 (center) shows the impact of the topic threshold c on the average size of the topics for the attribute *category*. Increasing the threshold effectively thins-out the topics on average. However, changing the topic threshold c also impacts the accuracy. Figure 3 (right) shows that tighter topics may fail to capture the user's intent and we observe decreasing accuracies for larger values of c. The actual value of c trades-off the specificity of topics and the accuracy of the topic prediction.

Table 2: Accuracies for the topic detection using all data.

	k	joint	colour	gender	category	price
MP	4	39,56	53,50	89,70	77,93	71,25
	3	39,53	52,83	89,70	78,09	71,04
	2	38,37	50,78	89,57	77,94	71,09
	1	30,82	42,37	89,15	77,29	70,02
M2	4	88,3	91,09	92,61	90,88	92,19
	3	91,13	92,73	92,45	92,04	92,56
	2	91,48	92,82	92,46	92,37	92,49
	1	91,53	92,85	92,4	92,39	92,55
LDA	-	2.84	12.31	81.18	51.22	41.71

4.4 Topic-driven Recommendations

We now demonstrate the effectiveness of the topic detection by translating the detected topics into recommendations according to Section 3.7. We use again the experimental setup from Section 4.2 and compare the approximate ensemble M2 with a collaborative filtering using matrix factorisation (CF) and a combination of topic models and collaborative filtering presented in [31] (TM). Both CF and TM are the same methods as described in [31], however we set $\alpha_{LDA} = 0.1$, number of topics $= 100$, and the number of factors in matrix factorisation $= 200$ by model selection. To evaluate these two baselines, items are ranked according to the previous clicks of the actual user as given by the user-item matrix. Note the conceptual difference of our fMDP and the collaborative filtering approaches. While the former takes a session-based approach and thus aims at short-term interests, the latter two are user-specific and could be considered global models for long-term user interests. Additionally, we incorporate three simple baselines: ranking items randomly (Rnd), ranking items according to their similarity to the previously viewed item so that items with the same attributes are ranked on top (Prev), and ranking items according to their popularity (Pop).

We also wanted to include a sequential MDP-based approach [28] as another competitor. However, a standard MDP approach as described in Section 3.1 is defined in terms of the items and turns out infeasible even for the small sample that we used in Section 4.1. There are more than 80,000 items in the small sample, leading to a minimum memory requirement of about 52GB for maintaining two tables of size $80,000 \times 80,000$, one for transitions and the other for the Q-values (for $k = 1$) using only four byte representations. The data set we are experimenting with in this section contains more than 240,000 items. We therefore leave this comparison for future work.

Figure 4 shows average ranks of the recommendations in the course of the sessions. Note that the average rank is a variant of Average Relative Position (ARP) [22]. As the baselines are static recommender systems that do not exploit the sequential nature of the data, their performance is more or less constant in the length of the session; small fluctuations disappear in the figure due to the log-scaled y-axis. The sequential M2 exploits the temporal nature of the data and adapts quickly to the topic of the session. The best method realises a second-order Markov assumption. The figure could be extended to the right to include longer sessions but the information gain is rather small as the performance of all methods does not change significantly.

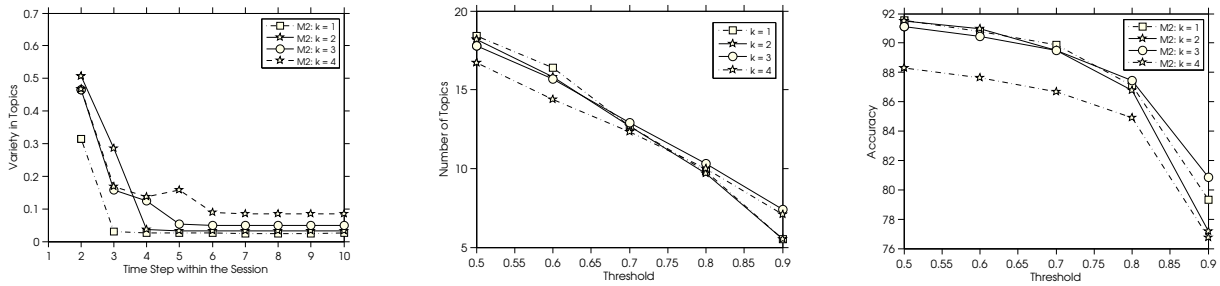

Figure 3: Left: Variance of topics. Center: Size of topics for attribute *category*. Right: Impact of topic threshold.

Table 3 shows aggregated results by averaging the performances over the length of the session. The baselines perform worse than M2, and among them, Prev and Pop outperform Collaborative based methods. The best method for histories of length two realises average ranks of about 15,000. Although the absolute number appears quite high, recall that there are more than 240,000 items in the data set. On average, clicked items are among the top 7% of the ranking for $k = 2$.

Table 3: Aggregated Average Ranks (in hundreds)

M2, $k =$								
1	2	3	4	TM	CF	prev	pop	rnd
174	158	209	270	723	749	272	295	1213

4.5 Discussion

Tables 1 and 2 exhibit differences in the predictability of the attributes. Unsurprisingly, *gender* is always predicted with high accuracy as it is unlikely that users switch often between genders within a session. The same is held for the attribute *category*. In contrast to *gender* and *category*, attributes *colour* and *price* prove more difficult. Apparently, users are somewhat flexible about prices and colours. Nevertheless, we observe highly accurate predictions for these attributes for the approximate ensemble M2.

Note that the choice of k depends on the application at hand. Our results show that the performance of the exact M1 increases with larger k (Tables 1). However, the larger the history, the longer it may take to adapt to a change in the topic; for instance because the user has not found what she was searching for or is distracted by a completely different item that is also displayed on the page. In practice, the fMDPs could be reset after cart or purchase operations by the user. The approximate fMDPs however perform better for short histories although the effect becomes smaller for larger training sets. We credit this finding to difficulties in the approximation caused by sparsity in the data distribution.

Since the internal representation of the factored MDPs is a graphical model, it is straight forward to augment additional variables to capture the context of the user. A promising candidate seems to be the time the user spends on the page before clicking. Very short stays could be an indicator for dissatisfaction, possibly followed by a change in topic while longer stays may give rise to a careful examination of the item at hand and a possible cart operation.

Finally, recall the conceptual differences of the fMDP-based recommender and the collaborative filtering baselines. While the former takes a session-based approach (short-term interests), the latter is user-centric and implements the notion of personalisation (long-term interests). Thus, the two strategies can be considered orthogonal. An interesting open questions is therefore whether it is possible to combine session-based with personalised strategies to obtain the best of the two worlds.

5. RELATED WORK

Topic detection is a broad field in machine learning, particularly for processing text. Topics of static data collections such as text corpora are traditionally identified using Latent Dirichlet Allocation (LDA) [5] and variations thereof. The evolution of topics in data streams is for instance detected by modelling time [33] or by introducing additional dependencies [2]. Other approaches, such as dynamic topic models [4] and online LDA [1], study segmented data streams. The idea is to turn topics of previous segments into priors for the actual time slice. A drawback of these approaches is that the topics remain constant across segments; effectively the same topics are re-identified and there is no mechanism to discard outdated topics or to introduce new ones.

Barbieri et al. [2] extend LDA to a first-order Markov model that determines topics of interest for collaborative recommendations. They propose a personalised recommender system based on user click histories where topics are identified for every user in the system. Wang and Blei [31] study LDA with collaborative filtering and matrix factorisation. They deploy topic models to assess content similarities in the reduced space of topics. Similarly, Chatzis [10] proposes to combine collaborative filtering with indian buffet processes for movie recommendations. The three approaches therefore aim at capturing long-term interests of users and an application to short-term goals of a session is not straight forward. By contrast, Wang and Zhang [32] propose a session aware recommender system that aims to capture the general intention of users in terms of three predefined and abstract categories: repurchase, variety-seeking, and buying new products. Note that the topics in [2, 31, 10, 32] are computed prior to the recommendation and can thus be considered static.

Markov decision processes (MDPs) [29, 24] are frequently used for sequential decision-making under uncertainty. Shani et al. [28] introduce sequential MDPs for recommender systems. Prior to their work, Zimdars et al. [35] propose a sequential recommender system where item recommendations are computed by random forests. Rendle et al. [26] study first-order Markov chains with matrix factorisation for basket recommendations. A reinforcement learning approach to recommender systems based on Q-learning has been presented in [30]. Moreover, Karatzoglou [16] combines temporal and collaborative aspects by minimising regularised loss functions. We design our approach based on fMDPs to take advantages of both, MDPs and factorisations. Factored MDPs are introduced by Boutilier [6].

6. CONCLUSIONS

We presented a sequential session-based approach for detecting the intention of user sessions on the Web. We phrased the problem as a topic detection task in terms of item attributes and proposed to solve the task via factored MDPs. We argued that a straight forward application is infeasible and devised an efficient formulation

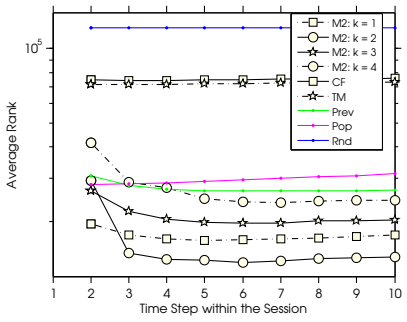

Figure 4: Average ranks for the recommendation task.

by assuming independence of attributes. We showed that factored MDPs with independent components admit an equivalent representation as an ensemble of independent fMDPs with structured value functions. Additionally, we presented an approximation of the ensemble and evaluated both methods on a large click log. Our empirical results showed that our methods were able to accurately detect topics of sessions. Translating our approach into a topic-driven recommender system outperformed collaborative baseline methods simple straw men.

Acknowledgements

We would like to thank Chong Wang and David M. Blei for sharing their code with us on short notice. We are grateful to Zalando for sharing data and actively supporting our work. Maryam Tavakol is supported by a Zalando scholarship.

7. REFERENCES

[1] L. AlSumait, D. Barbará, and C. Domeniconi. Online LDA: adaptive topic models for mining text streams with applications to topic detection and tracking. In *ICDM*, 2008.

[2] N. Barbieri, G. Manco, E. Ritacco, M. Carnuccio, and A. Bevacqua. Probabilistic topic models for sequence data. *MLJ*, 93:5–29, 2013.

[3] R. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *ICDM*, 2007.

[4] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML*, 2006.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[6] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *JAIR*, 11:1–94, 1999.

[7] C. Boutilier, R. Dearden, and M. Goldszmidt. Stochastic dynamic programming with factored representations. *Artificial Intelligence*, 121(1):49–107, 2000.

[8] H. Cao, D. H. Hu, D. Shen, D. Jiang, J.-T. Sun, E. Chen, and Q. Yang. Context-aware query classification. In *Conf. on Res. and Dev. in IR*, 2009.

[9] A. Celikyilmaz, D. Hakkani-Tür, and G. Tür. Leveraging web query logs to learn user intent via bayesian discrete latent variable model. In *ICML*, 2011.

[10] S. P. Chatzis. A coupled indian buffet process model for collaborative filtering. In *ACML*, 2012.

[11] M. Daoud, L. Tamine-Lechani, M. Boughanem, and B. Chebaro. A session based personalized search using an ontological user profile. In *Proceedings of the 2009 ACM Symposium on Applied Computing*, 2009.

[12] G. Giannopoulos, U. Brefeld, T. Dalamagas, and T. Sellis. Learning to rank user intent. In *CIKM*, 2011.

[13] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored MDPs. *JAIR*, 19:399–468, 2003.

[14] P. Haider, L. Chiarandini, and U. Brefeld. Discriminative clustering for market segmentation. In *KDD*, 2012.

[15] A. Hassan, R. Jones, and K. Klinkner. Beyond DCG: user behavior as a predictor of a successful search. In *WSDM*, 2010.

[16] A. Karatzoglou. Collaborative temporal order modeling. In *RecSys*, 2011.

[17] D. Koller and R. Parr. Computing factored value functions for policies in structured MDPs. In *IJCAI*, volume 99, pages 1332–1339, 1999.

[18] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, 2010.

[19] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[20] Z. Ma, G. Pant, and O. R. L. Sheng. Interest-based personalized search. *ACM Trans. Inf. Syst.*, 25(1), 2007.

[21] I. K. Paparrizos, B. B. Cambazoglu, and A. Gionis. Machine learned job recommendation. In *RecSys*, 2011.

[22] I. Pilászy, D. Zibriczky, and D. Tikk. Fast ALS-based matrix factorization for explicit and implicit feedback datasets. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 71–78. ACM, 2010.

[23] S. Purushotham, Y. Liu, and C.-C. J. Kuo. Collaborative topic regression with social matrix factorization for recommendation systems. In *ICML*, 2012.

[24] M. Puterman. *Markov Decision Processes*. Wiley, New York, 1994.

[25] N. Ramakrishnan, D. Conry, and Y. Koren. Recommender systems for the conference paper assignment problem. In *RecSys*, 2009.

[26] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized Markov chains for next-basket recommendation. In *WWW*, 2010.

[27] E. Sadikov, J. Madhavan, L. Wang, and A. Halevy. Clustering query refinements by user intent. In *WWW*, 2010.

[28] G. Shani, D. Heckerman, and R. I. Brafman. An MDP-based recommender system. *JMLR*, 6:1265–1295, 2005.

[29] R. Sutton and A. Barto. *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA, 1998.

[30] N. Taghipour, A. Kardan, and S. S. Ghidary. Usage-based web recommendations: A reinforcement learning approach. In *RecSys*, 2007.

[31] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *KDD*, 2011.

[32] J. Wang and Y. Zhang. Opportunity model for e-commerce recommendation: right product; right time. In *Conf. on Res. and Dev. in IR*, 2013.

[33] X. Wang and A. McCallum. Topics over time: a non-Markov continuous-time model of topical trends. In *CKDM*, 2006.

[34] R. W. White, P. N. Bennett, and S. T. Dumais. Predicting short-term interests using activity-based search context. In *CIKM*, 2010.

[35] A. Zimdars, D. Chickering, and C. Meek. Using temporal data for making recommendations. In *UAI*, 2001.