Conversational Recommender System by Bayesian Methods

Francesca Mangili, Denis Broggini, and Alessandro Antonucci

IDSIA, Lugano (Switzerland) {francesca,denisb,alessandro}@idsia.ch

Abstract. We present a Bayesian approach to conversational recommender systems. After any interaction with the user, a probability mass function over the items is updated by the system. The conversational feature corresponds to a sequential discovery of the user preferences based on questions. Information-theoretic criteria are used to optimally shape the interactions and decide when the conversation ends. Most probable items are consequently recommended. Dedicated elicitation techniques for the prior probabilities of the parameters modelling the interactions are derived from basic structural judgements based on logical compatibility and symmetry assumptions. Such prior knowledge is combined with data for better item discrimination. Our Bayesian approach is validated against matrix factorization techniques for cold-start recommendations based on metadata using the popular benchmark data set *MovieLens*. Results show that the proposed approach allows to considerably reduce the number of interactions while maintaining good ranking performance.

Keywords: Conversational Recommender Systems · Bayesian Methods.

1 Introduction

The task of selecting from a collection of items the one which is in some sense optimal for a specific user is a classic AI problem. Several algorithms have been explored to perform tasks of this kind and automated recommendations are currently provided by all major e-commerce platforms [18, 9]. In standard setups, no explicit interaction with the users is considered, and the recommendation system bases its decision on some background information (also called *metadata*) about the user, historical records of her choices and those of other similar users, and, more recently, contextual information [1, 13].

Modern technologies such as chatbots or personal assistants need instead systems capable of supporting and modelling dynamic and sequential interactions, thus requiring a substantial re-design of traditional approaches. Here we focus on such a newer class of recommendation systems, called here *conversational*, as we term conversation a sequence of dynamically customized interactions between the user and the system, before the latter returns an item recommendation [4]. This class of systems is typically based on knowledge-based recommendation techniques requiring a strong interaction with the user.

Recommendations based on conversations are particularly relevant when the goal is to support the user in purchasing high-involvement products. In such situations, indeed, the user wants to be involved in the decision process, and thus is not bothered by the need to interact with the system [10].

To achieve that, we take inspiration from existing approaches in the field of computer testing [3, 14], whose goal is to determine the skills of a student on the basis of the answers to the questions of a test. In particular we focus on *adaptive* approaches, where the system selects the next question to ask to the student from a given set on the basis of the previous answers by information-theoretic scores, that are also used to decide when the test should be ended. This can be easily obtained by generative probabilistic models such as Bayesian networks [16], to be sequentially updated any time a new answer is collected.

The adaptive concept is reshaped here as a conversational approach which lends itself to the (future) development of a dynamic generation of questions and richer interaction models. The question selection is consequently reduced to an inference in the corresponding generative model. The major issue consists therefore in a reliable and scalable elicitation strategy of the model parameters. This is achieved by combining prior structural judgements and historical data.

The paper is organized as follows. We first review the basics of probabilistic approaches to recommender systems in Section 2. Elicitation strategies based on structural judgements such as logical compatibility and symmetry are in Section 3. A procedure to cope also with non-exclusive answers is in Section 4. Our conversational approach driven is indeed detailed in Section 5. In Section 6, we show how our elicitation techniques can be merged with historical data and how to prevent a fragmentation of these data over the items. Finally, we discuss our experiments in Section 7 and the conclusions in Section 8.

2 Bayesian Recommendations

Consider a recommendation system based on a *catalogue* $\mathcal{I}$ of n items, say $\mathcal{I} := \{i_1, \ldots, i_n\}$. The system is intended to support the user in selecting a single item from the catalogue on the basis of a conversational process. The system outcome can however be used to rank the items. We assume that, after the conversation, the user always picks one item from $\mathcal{I}$. The uncertain quantity I denotes the element of $\mathcal{I}$ to be picked by the user. We consider a Bayesian setup and model the subjective probabilities of picking the different items before the beginning of the conversational process as a *prior* probability mass function $P(I)$.

We call *questions*, the interactions between the system and the user. Let us start with a static approach to elicitation based on a list of m questions $\mathbf{Q} = \{Q_1, \ldots, Q_m\}$, called here a *questionnaire*. A generic question is denoted as an uncertain quantity Q taking its possible values (*answers*) from $\mathcal{Q}$. We assume a finite set of possible answers for Q, say $\mathcal{Q} := \{q^1, \ldots, q^r\}$. The selection of the optimal item to be suggested at the end of the conversation might be based on the conditional probabilities $P(i|\mathbf{q})$ for each item $i \in \mathcal{I}$ given the collected answers, denoted as $\mathbf{q}$.

To improve the quality of the user experience, it seems desirable to minimize the number of questions required to safely identify the optimal suggestion. To this goal, the list of questions in $\mathbf{Q}$ should be built dynamically. Such a customized list of questions is called *conversation.* Before discussing this dynamic approach, we will present the set up that allows for the computation of $P(i|\mathbf{q})$, be $\mathbf{q}$ the output of a conversation or a whole questionnaire.

Let us start from the case of a single question Q. As a model of the relation between Q and I, we might be able to elicit a *conditional probability table* $P(Q|I)$, whose columns are indexed by the answers, and whose rows are associated to the items. After an answer $Q = q \in \mathcal{Q}$ is received from the user, the probability mass function over the items can be updated by Bayes rule, i.e.,

$$P(i|q) = \frac{P(i)\ P(q|i)}{\sum_{i \in \mathcal{I}} P(i)\ P(q|i)}\,. \tag{1}$$

This shows that the impact of an answer $Q = q$ on the choice I only depends on the relative proportions of the values of $P(q|i)$ for the different values of $i \in \mathcal{I}$. In particular, setting an element of the conditional probability table equal to zero implies a logical constraint for which the answer associated with column $Q = q$ makes the choice of the item associated with row $I = i$ impossible. To model a whole conversation made of m questions, we formulate a naive-like assumption about the conditional independence of the questions given the item:

$$P(q_1, \ldots, q_m|i) = \prod_{j=1}^{m} P(q_j|i)\,. \tag{2}$$

Under the assumption in Equation (2), if a conditional probability table $P(Q|I)$ is available for each $Q \in \mathbf{Q}$, the probability $P(i|\mathbf{q})$ can be obtained by recursively applying Equation (1) to update the probability $P(i|q_1, \ldots, q_k)$ after the first k answers by conditioning also on the next answer q_{k+1}. The follwoing example will be used through the paper to illustrate the features of our method.

Example 1. Users of a platform for online booking of entertainers are invited to answer a number of questions about the kind of entertainment they are looking for. The two questions Q_1 and Q_2 asked to the users and a catalogue of three entertainers-items are in Table 1.

Table 1: A catalogue and two questions of a questionnaire

$\mathcal{I}$	Description
i_1	DJ for all events
i_2	Band for weddings and corporate events
i_3	Magician for birthdays and parties

Q_1	Which artist?
q_1^1	DJ
q_1^2	Band
q_1^3	Musician
q_1^4	Entertainer

Q_2	Which event?
q_2^1	Wedding
q_2^2	Corporate event
q_2^3	Birthday
q_2^4	Party for kids

3 Structural Judgements for Prior Elicitation

In this section we describe a general elicitation procedure for the conditional probability table $P(Q|I)$. This is based on logical arguments, discussed in Section 3.1, to be integrated by symmetry considerations, discussed in Section 3.2.

3.1 Logical Compatibility

Consider the natural language item descriptions in Table 1. It is straightforward to deduce from this information a number of *incompatibility* relations between the entertainers and the answers to the two questions in Table 1. E.g., the magician is not available for weddings, this meaning that answer q_2^1 is not compatible with item i_3. Assuming such a compatibility analysis available for all the items and answers, we might define an indicator function $\delta(i,q)$, for each $i \in \mathcal{I}$ and $q \in \mathcal{Q}$, and for each $Q \in \mathbf{Q}$, that returns one if i is compatible with q and zero otherwise. An item i is therefore characterized by its set of compatible answers or *support*. We will denote as $\mathcal{Q}_j(i)$ the support of i associated with Q_j, i.e.,

$$\mathcal{Q}_j(i) := \{q \in \mathcal{Q}_j : \delta(i,q) = 1\}\,. \tag{3}$$

For instance, in Example 1, $\mathcal{Q}_1(i_2) = \{q_1^2\}$ and $\mathcal{Q}_2(i_2) = \{q_2^1, q_2^2\}$. Notation $\mathcal{Q}(i)$ is used instead for the union of all the supports of i associated to the different questions in $\mathbf{Q}$. E.g., in Example 1, $\mathcal{Q}(i_2) = \{q_1^2, q_2^1, q_2^2\}$.

At the probabilistic level, we translate logical incompatibilities into zero-probability statements as follows:

$$q \notin \mathcal{Q}(i) \Rightarrow P(q|i) = 0\,, \tag{4}$$

i.e., if the right item to recommend is not compatible with an answer, that answer is impossible. If $q \notin \mathcal{Q}(i)$, also the joint probability $P(i,q)$ is zero as $P(i,q) = P(q|i)\ P(i)$. On the other hand, if $q \in \mathcal{Q}(i)$, the elicitation of $P(q|i)$ might need additional assumptions. This is not the case of the following example.

Example 2. In the setup of Example 1, for question Q_1, $\mathcal{Q}_1(i_1) = \{q_1^1\}$, $\mathcal{Q}_1(i_2) = \{q_1^2\}$ and $\mathcal{Q}_1(i_3) = \{q_1^4\}$. In other words, each item admits only a single consistent answer. Thus, because of Equation (4), for each i, $P(q|i)$ is zero for any q apart from the one in $\mathcal{Q}_1(i)$. $P(Q_1|I)$ takes therefore the values in Table 2 (left).

Table 2: Elicitation of $P(Q_1|I)$ and $P(Q_2|I)$ (two alternatives)

$P(q_1\|i)$	q_1^1	q_1^2	q_1^3	q_1^4
i_1	1	0	0	0
i_2	0	1	0	0
i_3	0	0	0	1

$P(q_2\|i)$	q_2^1	q_2^2	q_2^3	q_2^4
i_1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
i_2	$\frac{1}{2}$	$\frac{1}{2}$	0	0
i_3	0	0	$\frac{1}{2}$	$\frac{1}{2}$

$P(q_2\|i)$	q_2^1	q_2^2	q_2^3	q_2^4	$\tilde{q}_2$
i_1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	0
i_2	$\frac{1}{4}$	$\frac{1}{4}$	0	0	$\frac{1}{2}$
i_3	0	0	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$

The above notion of logical compatibility is sufficient to elicit the conditional probability table $P(Q|I)$ in the special case where each item is compatible only with a single answer, i.e., $|\mathcal{Q}(i)| = 1$ for each $i \in \mathcal{I}$, where symbol $|\cdot|$ denotes cardinality. If this is the case, $P(q|i) = 1$ if $q \in \mathcal{Q}(i)$ and zero otherwise.

For questions of this kind, after an answer $Q = q \in \mathcal{Q}$, the updating as in Equation (1) corresponds to a uniform *scaling* of the prior probability for the items compatible with q, whereas all the incompatible ones receive zero mass:

$$P(i|q) = \begin{cases} P(i)[\sum_{i':q\in\mathcal{Q}(i')} P(i')]^{-1} & \text{if } q \in \mathcal{Q}(i) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Note that in this setting the Bayesian approach acts as a logical filter, making the items incompatible with the answers impossible.

3.2 Symmetry Statements

Consider a question $Q \in \mathbf{Q}$ such that the assumptions in the previous section are not satisfied, i.e., there are items compatible with more than a single answer. In this case, the elicitation of $P(Q|I)$ requires further assumptions. We adopt a strategy forcing answers to have the same impact on all compatible items. Such a prior assumption will be relaxed when coping with historical data in Section 6). Other strategies are in [15]. Let us introduce the discussion by an example.

Example 3. In the setup of Example 1, for question Q_2, we have $\mathcal{Q}_2(i_1) = \{q_2^1, q_2^2, q_2^3, q_2^4\}$, $\mathcal{Q}_2(i_2) = \{q_2^1, q_2^2\}$ and $\mathcal{Q}_2(i_3) = \{q_2^3, q_2^4\}$. Assuming the same probability for the answers compatible with a given item completely determines the conditional probability table $P(Q_2|I)$, that is depicted in Table 2 (middle).

The procedure considered in Example 3 allows to elicit a conditional probability table in the general case of items compatible with multiple answers. This basically corresponds to set a uniform probability $P(q|i) = |\mathcal{Q}(i)|^{-1}$ for all $q \in \mathcal{Q}(i)$ and zero otherwise. After answer $Q = q$, the updating in Equation (1), differently from Equation (5), is a non-uniform rescaling:

$$P(i|q) = P(i)\frac{|\mathcal{Q}(i)|^{-1}}{\sum_{i':q\in\mathcal{Q}(i')} P(i')|\mathcal{Q}(i')|^{-1}}\,, \tag{6}$$

if $q \in \mathcal{Q}(i)$ and zero otherwise. Yet, when supports have different cardinalities, this approach might lead to questionable results as in the following example.

Example 4. Consider the same setup of Example 2 with a uniform prior over the items, i.e., $P(i_k) = \frac{1}{3}$ for $k = 1, 2, 3$. After the answer $Q_2 = q_2^1$, following Equation (6), we obtain $P(i_1|q_2^1) = \frac{1}{3}$ and $P(i_2|q_2^1) = \frac{2}{3}$. In other words, both the band and the DJ are available for the wedding, but despite the uniform prior, the band has twice the probability of being selected, the reason being the lower cardinality of the support of the band compared to the DJ.

According to Example 4, the strategy of Example 2 leading to Table 2 (middle) has a bias towards items with smaller supports (i.e., lower compatibility). To prevent such bias, we propose an alternative elicitation as in this example.

Example 5. In the setup of Example 2 add to question Q_2 a fifth answer $\tilde{q}_2$ compatible with all the items. Consider the elicitation of $P(I|Q_2)$ in Table 2 (right). With a uniform prior over I as in Example 4, $P(i_1|q_2^1) = P(i_2|q_2^1) = \frac{1}{2}$.

The procedure in Example 5 can be generalized by setting the parameters of the conditional probability table $P(Q|I)$ as follows:

(i) $P(q|i) = \kappa^{-1}$ for each $q \in \mathcal{Q}$ and $i \in \mathcal{I}$,
(ii) $P(\tilde{q}|i) = 1 - \kappa^{-1}|\mathcal{Q}(i)|$.

The value of $\kappa > 0$ is arbitrary, provided that the table is a proper stochastic matrix, i.e., its elements have non-negative values and normalized rows. Setting $\kappa := \max_{i \in \mathcal{I}} |\mathcal{Q}(i)|$ guarantees non-negative probability of the *dummy* state $\tilde{q}$. Non-negativity of other elements follows from (i) and normalization from (ii).

Considering this approach to the elicitation of $P(Q|I)$, after a (non-dummy) answer $Q = q$, the updated probability is given by Equation (5) as all valid $P(i|q)$ have the same value. This proves that posterior inferences are not affected by the choice of κ. Moreover, if i' and i'' are two items consistent with q:

$$\frac{P(i'|q)}{P(i''|q)} = \frac{P(i')}{P(i'')}\,, \tag{7}$$

i.e., even in general, this approach does not suffer from the bias in Example 4.

4 Coping with Non-Exclusive Answers

In some cases it might be reasonable to allow users to select multiple answers to a question, thus violating the original assumption about answers being described as mutually exclusive events. For instance, in the setup of Example 1, if we assume the answers to Q_1 non-exclusive, an user might return a set-valued answer such as $\mathcal{Q}_1^* = \{q_1^1, q_1^2\}$, meaning *(DJ, band)*. Different semantics can be considered to model set-valued observations. Following the random set literature [5], we distinguish between *ontic* (or conjunctive) and *epistemic* (or disjunctive) semantics. From an epistemic perspective, the set reflects an incomplete knowledge about the actual value of the answer (which has a single value). In our example, this assumption implies that the user is eventually looking for a DJ or for a band and certainly not for a musician or an entertainer. Yet, for some reason (e.g., limited capability of introspection or limited amount of time to answer), she is not able to isolate a single option (which should however exists). The ontic view, instead, regards the set of answers as a precise component of a more complex entity, e.g., the combination *(DJ, band)* is exactly the option the user wants to select. The two semantics requires different modelling strategies.

In the ontic case, we replace the set of possible answers $\mathcal{Q}$ with its partition set in order to include all the $2^{|\mathcal{Q}|-1}$ possible set-valued options (remember that the user should always provide at least an answer). Afterwards, the elicitation and updating procedures remain as in the previous section.

In the epistemic case, in terms of elicitation nothing changes as the question Q is assumed to take single elements of $\mathcal{Q}$ as values. Consequently the quantification of $P(Q|I)$ can be achieved as in Section 3.2. The difference arises at the updating level as the procedure in Equation (5) cannot directly cope with set-valued answers. To update item probabilities we should instead intend a set-valued answer $\mathcal{Q}^* \subseteq \mathcal{Q}$ as a *virtual evidence* [17], that is not able to reveal the actual state of Q in a reliable way. A virtual evidence describes an observation in terms of its likelihoods for the different (actual) states of Q, these values being defined up to a non-negative constant. Here, we set to zero the likelihoods of the answers not selected by the user, while giving the same value to the selected ones, i.e., those in $\mathcal{Q}^*$. This leads to the following updating rule:

$$P(i|\mathcal{Q}^*) := \frac{\sum_{q\in\mathcal{Q}^*} P(q|i)\ P(i)}{\sum_{q\in\mathcal{Q}^*} P(q)}\,, \tag{8}$$

where $P(q) = \sum_{i\in\mathcal{I}} P(q|i)\ P(i)$. Let us see how this can be achieved in practice.

Example 6. Consider the same setup of Example 5 but with a set-value answer $\mathcal{Q}_1^* = \{q_1^1, q_1^2\}$, i.e., the user replied DJ *and* band. Following Equation (8):

$$P(i|\mathcal{Q}_2^*) \propto [P(q_2^1|i) + P(q_2^2|i)]P(i) \tag{9}$$

for each $i \in \mathcal{I}$, where the proportionality constant is obtained by normalization.

5 Adaptive Conversations

We detail here the high-level procedure used to create a conversation by selecting sequentially the question and driven by decision theoretic criteria. This is inspired by the similar approaches considered for computer adaptive testing [14].

Question Selection. In classical recommendation systems the assessment of the user preferences with respect to the different items is based on a static block of background user information. Such information can be already available in the system or directly obtained from the user after some kind of reduced interaction, e.g., a predefined questionnaire. However, as discussed in the introduction, in modern setups the process of collecting information about the user preferences with respect to the catalogue should be based on a sequence of dynamic interactions. In this view, the questionnaire approach leaves the place to a conversational process taking the form of a personalized sequence of questions dynamically picked from a larger set of questions. The prior probability mass function $P(I)$ is thus sequentially updated each time a new answer is collected, and the updated probability $P(I|\mathbf{Q})$ is used to select the most informative next question. The choice between a possibly huge set of candidate questions/interactions can be driven by information-theoretic criteria making any sequence potentially different from the other. In particular, taking inspiration from the literature in the field of adaptive testing, we pick the question that minimizes the conditional entropy (and hence maximizes the expected information gain), i.e., the adaptive conversational process selects the question Q_j^* such that:

$$j^* = \arg \min_{j=1,\dots,m} H(I|Q_j)\,, \tag{10}$$

where $H(I|Q) := \sum_{q\in\mathcal{Q}} H(I|q)P(q)$ and $H(I|q)$ is the entropy of the posterior mass function over I after the answer $q \in \mathcal{Q}$ to the question Q, and $\{Q_j\}_{j=1}^m$ is the set of questions the system can choose from.

Stopping Rule. This procedure is iterated after every answer and the conversation ends if the posterior entropy $H(I|q)$ decreases under a fixed threshold. As the entropy of a mass function $P(I)$ is defined as $H(I) := -\sum_{i\in\mathcal{I}} P(i) \log_{|\mathcal{I}|} P(i)$, a natural threshold H_τ^* is the entropy of a mass function over I which is uniform on τ items, while the other ones have zero probability, i.e., $H_\tau^* := -\log_{|\mathcal{I}|} \tau^{-1}$. Setting this value in the stopping rule forces the system to halt when most of the posterior probability mass is concentrated on the τ most probable items.

6 Coping with Data

The elicitation procedure in Section 3 is based on structural judgements about logical compatibility for items and answers merged with symmetry (i.e., indifference) statements. Yet, this might be insufficient to capture actual preferences as outlined by the following example.

Example 7. Consider the setup in Example 5. In Table 2 (right), $P(q_2^1|i_2) = P(q_2^2|i_2)$, i.e., the band is equally likely to perform at weddings and corporate events. Yet, in past activities, the band was more likely to attend weddings.

Data about the users, in the form of their answers to the questions and the items eventually picked, might help to achieve a better quantification of the model parameters. In this section we describe how to combine data with the elicitation procedure described in Section 3. Let us first show how this can be directly achieved in the conditional probability table $P(Q|I)$.

6.1 Bayesian Learning

Bayesian approaches allow to combine the elicited parameters with data about items and questions. The probability tables can be used to parametrize a multinomial Dirichlet prior, to be merged with the likelihood of the observed data. The posterior expected value might increase the discriminative power of the system and the consequent quality of the recommendations.

Let $\alpha_j^k := P(q^k|i_j)$ for each $q^k \in \mathcal{Q}$ and $i_j \in \mathcal{I}$, where $P(Q|I)$ has been elicited as in Section 3.2. Assume that n_j i.i.d. observations about users who picked item $I = i_j$ are available, of which n_j^k reported answer q^k, for $k = 1, \dots, |\mathcal{Q}|$. We update the values of $P(Q|i)$ by integrating the product of a Dirichlet prior with weights $s\ \alpha_j^k$ and the likelihood based on the counts [2]. This gives the posterior:

$$P'(q^k|i_j) = \frac{s\alpha_j^k + n_j^k}{s + n_j} \,, \tag{11}$$

for each $j = 1, \ldots, |\mathcal{Q}|$, where $s > 0$ is the equivalent sample size of the Dirichlet prior. Note that the parameters of the Dirichlet distribution are required to be strictly positive, while the elements of $P(Q|I)$ can be zero. If the latter is the case, we take the limit $\alpha_j^k \to 0$ and consider Equation (11) still valid.

Example 8. In the setup of Example 5 assume data about the answers provided to question Q_2 for the band i_2 available. The counts are $n_2^1 = 5$, $n_2^2 = 3$, $n_2^3 = 1$, $n_2^4 = 0$. For $s = 1$, with the prior parameters as in Table 2 (right), Equation (11) *gives $P(q_2^1|i_2) = 0.525$ and $P(q_2^2|i_2) = 0.325$, which means that higher number of weddings compared to corporate events breaks the prior equal probability of the two options. Note also that, despite its (prior) incompatibility with the band the birthday option (q_2^3) has a non-zero count. In such a* prior-data conflict *[7], the option takes a non-zero probability $P(q_2^3|i_2) = 0.1$, while the incompatible and unobserved answer q_2^4 gets sharp zero probability.*

A separate discussion should be provided for the updated probabilities of the dummy state $\tilde{q}$. This answer has zero counts by definition, but its prior might be non-zero, thus leading to a non-zero posterior (e.g., in the previous example, $P(\tilde{q}_2|i_2) = 0.05$). As discussed in Section 3.2, these non-zero values are not affecting those of the updated mass functions $P(I|q)$, and hence those of the corresponding entropies $H(I|q)$. This is not the case for conditional entropy $H(I|Q)$ because of weighted average based on $P(Q)$. A temporary *revision* of the marginal probability mass function $P(Q)$, that sets $P(\tilde{q}) = 0$ and rescales the other values, only for the choice of the next question, might fix the issue.

6.2 Preventing Data Fragmentation

The above procedure separately processes data corresponding to different items. For large catalogues this induces a data *fragmentation*, thus making some estimator unreliable. To prevent this we introduce the notion of *latent question*. Given question Q, its latent counterpart $\hat{Q}$ is a second variable taking its values (called *latent answers*) from the same set of possible states $\mathcal{Q}$ and sharing the same supports (see Section 3.1). The latent question $\hat{Q}$ is a (stochastic) function of $I = i$ and $Q = q$. Unlike Q, $\hat{Q}$ is always consistent with the support of i, being a random element of it or its unique value, or q in the case also Q is consistent. Therefore, the support of an item i can be defined in terms of the *latent* compatible answers and indicated as $\hat{\mathcal{Q}}(i)$. We assume conditional independence between Q and I given $\hat{Q}$, this meaning that if the answer to the latent question is known, the answer to the non-latent question does not depend on the selected item, i.e., $P(q|\hat{q}, i) = P(q|\hat{q})$, for each $q, \hat{q} \in \mathcal{Q}$ and $i \in \mathcal{I}$. By this statement and total probability, the elements of table $P(Q|I)$ can be expressed as:

$$P(q|i) = \sum_{\hat{q} \in \mathcal{Q}} P(q|\hat{q})\ P(\hat{q}|i)\,. \tag{12}$$

Equation (12) offers a strategy to probability tables elicitation less prone to fragmentation issues. To achieve that, for $P(\hat{Q}|I)$, we adopt the strategy in Section 3.2. The data are used instead for the quantification of $P(Q|\hat{Q})$, for which we adopt a diagonal Dirichlet prior to be combined with the data likelihood. Note that the values of $\hat{Q}$ can be generated from those about Q and I. To reduce variance, we replace sampled outcomes of $\hat{Q}$ with a collection of uniform fractional counts for the different elements of $\hat{\mathcal{Q}}(i)$. Following a Bayesian scheme analogous to that described in Section 6.1, we indeed obtain the posterior estimator for $P(Q|\hat{Q})$. Finally, marginalizing the latent question as in Equation (12) we obtain $P(Q|I)$. Note also that, without data, $P(Q|\hat{Q})$ acts as the identity matrix, and the approach proposed here corresponds to that in Section 3.2.

Example 9. In the setup of Example 1, consider the data about Q_1 and I in Table 3 (left). The DJ and the band exhibit some inconsistency with respect to their support, e.g., once the DJ performed even if the request was for a musician. This is modelled by the latent question $\hat{Q}_1$. The definition of $\hat{Q}_1$ allows to complete the data about the latent question as in Table 3 (middle) and hence $P(\hat{Q}|Q)$. Following Equation (12), we combine this probability table with $P(\hat{Q}_1|I)$ elicited as in Table 2 and obtain the quantification in Table 3 (right).

Table 3: Counts for Q_1, $\hat{Q}_1$ and I (left) and corresponding $P(Q_1|I)$ (right).

$n(Q_1, I)$	q_1^1	q_1^2	q_1^3	q_1^4
i_1	4	0	1	0
i_2	2	6	3	0
i_3	0	0	0	5

$n(\hat{Q}_1, Q_1)$	q_1^1	q_1^2	q_1^3	q_1^4
$\hat{q}_1^1$	4	0	1	0
$\hat{q}_1^2$	2	6	3	0
$\hat{q}_1^3$	0	0	0	0
$\hat{q}_1^4$	0	0	0	5

$P(Q_1\|I)$	q_1^1	q_1^2	q_1^3	q_1^4
i_1	$\frac{5}{6}$	0	$\frac{1}{6}$	0
i_2	$\frac{1}{6}$	$\frac{7}{12}$	$\frac{1}{4}$	0
i_3	0	0	0	1

In the above procedure, the latent question and its operational definition can be intended as a rule for the clustering of data associated to different items, the latent answers indexing the clusters, i.e., all items associated to the data in cluster $\hat{q}$ include the latent answer $\hat{q}$ in their support. As we act in a probabilistic setting, *soft clustering* can be obtained [12], e.g., when the item associated to an observation has $|\mathcal{Q}(i)| > 1$ and the answer $q \notin \mathcal{Q}(i)$. Nothing prevents us from considering other data-based clustering algorithms relaxing the one-to-one correspondence between questions and latent questions.

In this section we have focused on models with a single question. To cope with multiple questions, the naive-like assumption formulated in Equation (2) for (non-latent) questions should be reformulated in terms of the latent ones and we assume conditional independence of the latent questions given the item.

Finally, if a more general notion of latent question is adopted, so that (observable) questions and latent questions may take their values from different sets, we could also consider the case of multiple (*observable*) questions associated to the same latent one. This can be easily achieved by maintaining the assumption of conditional independence between the item and all the questions given their latent question, that corresponds in this case to the conditional independence between the questions of the same latent question given the latent question itself. In practice, if Q_a and Q_b are the two questions and $\hat{Q}$ their common latent question, we have Q_a and Q_b independent given $\hat{Q}$, we separately learn $P(Q_a|\hat{Q})$ and $P(Q_b|\hat{Q})$, and finally obtain:

$$P(q_a, q_b|i) = \sum_{\hat{q} \in \hat{\mathcal{Q}}} P(q_a|\hat{q}) P(q_b|\hat{q}) P(\hat{q}|i)\,. \tag{13}$$

Moreover, with the aim of preserving the naive independence structure defined in Equation (2), which is not necessary but simplifies the implementation of the conversational approach discussed in Section 5, the above model may be approximated by one assuming conditional independence of Q_a and Q_b and having for all $q_a \in \mathcal{Q}_a$

$$P(q_a|i) = \sum_{q_b \in \mathcal{Q}_b} P(q_a, q_b|i)\,. \tag{14}$$

7 Experiments

For validation we consider the classical *MovieLens* database [8]. The dataset contains information about 13‘816 movies. Each record is characterized by a list of genres taken from nineteen pre-defined categories, a period (before 1980, from 1980 to 1999, from 2000 to 2016, after 2016, in our discretization) and a *tag* relevance score for each of the tags included in *MovieLens* database Tag Genome. The tag relevance score is a number between zero and one, describing how much the tag reflects the movie characteristics. E.g., tag *violence* has relevance 0.991 for the movie *Reservoir Dogs* and 0.234 for *Toy Story.* We only consider the 200 tags most used by the users, out of the 1‘128 available.

Tags together with genre and period are regarded as questions and used to simulate a conversation based on a static questionnaire whose goal is to detect the right item/movie. No set-valued answers are allowed for these questions. We instead allow sets for answers about the genre in the learning phase, for which the ontic semantics discussed in Section 4 is adopted. Each element of the set is treated as a fractional observation with weight inversely proportional to the cardinality of set of genres provided. The (unobserved) value assumed by the latent genres is assigned consistently with the support of the corresponding item using the procedure described in Section 6.2. On the other end, to simplify the procedure, and at the same time increasing the flexibility of the adaptive approach by augmenting the number of questions in the pool, each genre is separately asked by nineteen independent questions along the conversation.

For each movie in the database, a complete list of questionnaire answers is generated as follows. For the question *period* we might *corrupt* the actual period by changing it into the previous or the next with probability p_y. For the genre, we take the list of genres assigned to a movie and corrupt each of them (independently) with probability p_g. Corruption consists either in removing the genre from the list, or permuting the genre by replacing it by a similar one. Finally, answers to tag-related questions are simulated as follows. A positive answer to a question about a tag is assigned with probability equal to the tag relevance score of the movie. Subsequently, positive tag answers can be corrupted with probability p_t by setting them to zero. This corruption step, simulates a user who has, in general, a number of desiderata smaller than the total number of tags that could properly describe a movie. In our experiments we use $p_y = p_g = p_t = p$ and consider two noise levels, namely *low* ($p = 10\%$) and *high* ($p = 50\%$).

We select 2‘000 movies, used as the catalogue, from which we have generated the test set for the experiments. The remaining movies are used to generate the training dataset from which the conditional probability tables $P(Q|\hat{Q})$ for the genre, period and tag questions given the actual genre, period and tags of the selected movie (latent question) are estimated as described in Section 6. The tables $P(\hat{Q}|I)$ for the genre and period are defined by compatibility and similarity constraints as in Section 3. Concerning tags, instead, the probability $P(\hat{Q} = 1|i)$ of a tag being compatible with item i is set equal to the tag relevance score for item i. Finally, a uniform prior over the items is assumed. The goal of the system is to detect the right movie on the basis of the answers to the questions. In the conversational case, questions are selected following the adaptive strategy detailed in Section 5. The algorithm in charge of the elicitation and question selection has been implemented in Python.[1]

The performances of our approach are compared against that of traditional algorithms based on measures of similarity between the item and user features, such as the *LightFM* algorithm [11]. Here the item features, i.e., the (eventually probabilistic) supports $\mathcal{Q}_j(i)$ for all questions $Q_j,\ j = 1, \ldots, m$ and the user features, i.e., their (simulated) noisy answers, take values from the same space. Thus, there is no need of learning linear transformations to a common latent space, as one can simply compute the items-user similarity in the original space. In the experiments below, the *cosine similarity* is used to measure the affinity between items and users requests. Note that, as we are considering a situation where each user has only a single conversation with the system, we cannot learn users and items similarities from historical interactions as done in traditional collaborative filtering. The fact that the similarity-based approach does not take advantage of the training dataset to improve its item selection strategy limits its accuracy, especially when the noise is large. However, as no standard solutions could be found in the literature (the field of conversational recommendations is relatively new) and since our focus was the ability of the Bayesian model to drive an adaptive conversation (rather than a possible improvement in the accuracy of the final rank), we have been satisfied with validating its accuracy against

[1] `https://github.com/IDSIA/bayesrecsys`

such simple baseline, yet quite popular in the context of neighbourhood-based filtering [6].

As a first experiment, we evaluate the benefits of the adaptive approach in the question selection process with respect to a random strategy. Figure 1 shows the average probability of the movie selected by the user as a function of the number of questions for the two strategies with the two different noise levels. In both cases increasing the number of questions makes the right movie more probable, but the growth is quicker if the questions are selected on the basis of their expected information gain. with the gap between the two approaches increasing unless half of the questions in the questionnaire has been asked.

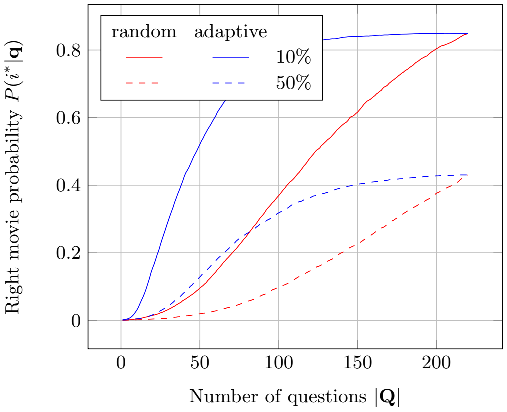

Fig. 1: Adaptive vs. random selection for two noise levels.

For the validation against the similarity-based approach, which does not provide probabilities, we consider the average rank of the right movie as a function of the number of questions. As in the previous experiment we consider our Bayesian approach with adaptive selection of the questions, and the same approach with random selection of the questions. The two methods are compared against the similarity-based approach. As we have no standard method available to select the order of the questions in this case, we test the similarity-based approach using both the order selected by the Bayesian conversation and the random order. Note that, as a consequence, the improvement in the performance of the similarity-based approach in the adaptive setting compared to the random one, are only due to the Bayesian question selection strategy. Results are shown in Figure 2 (top left), with a zoom of the tails in Figure 2 (top right). As in the previous analysis the advantages of the adaptive approach compared to the random one are clear. Moreover, results show that the Bayesian model, by correctly modelling the users behaviours from the historical data, outperforms the similarity approach the more the largest is the noise corrupting the data. In Figure 2 (top left) we also report as a black dotted line the ranks of the Bayesian adaptive approach based on structural judgments without learning from the data. The higher ranks compared to the approach based on data advocates the learning procedure discussed in Section 6.

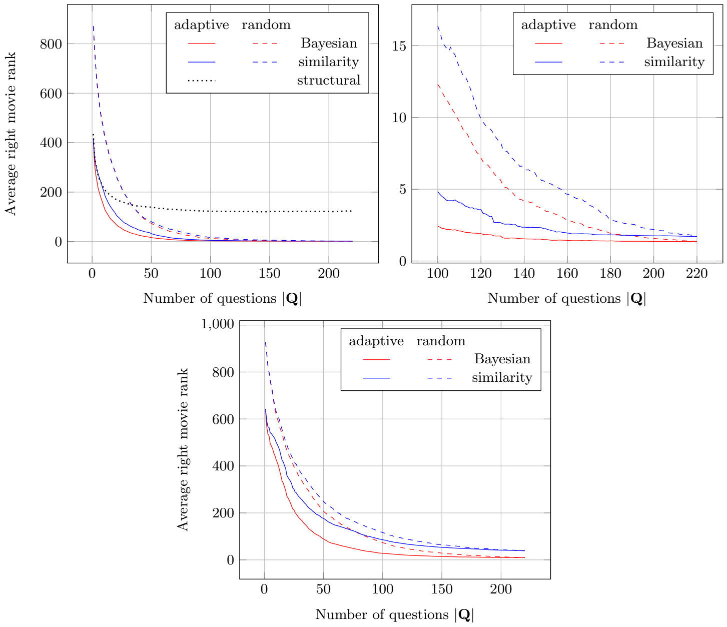

Fig. 2: Average recommendation rank as a function of the number of questions for low (10%, top left and right) and high (50%, bottom) noise levels.

8 Conclusions

A new approach to automatic recommendations which assumes a dynamic interaction between the system and the user to provide customized and self-adaptive recommendations has been developed on the basis of a pure Bayesian approach. The framework introduced in this paper sets the ground to several future developments, among which the dynamic generations of questions in order to improve the conversational nature of the system. This could be based on a natural language generation system interacting with the structured probabilistic description of item properties and elicitation of user needs. Also, this work has focused on a setting where each user interacts with the system a single time. Therefore, collaborative filtering approaches that learn users' taste and items similarities from historical data were not applicable. Such setting looks appropriate for highly involved decisions, such as the selection of an entertainer for a special event.

References

1. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Recommender systems handbook, pp. 217–253. Springer (2011)
2. Bernardo, J.M., Smith, A.F.: Bayesian theory, vol. 405. John Wiley & Sons (2009)
3. Butz, C.J., Hua, S., Maguire, R.B.: A web-based Bayesian intelligent tutoring system for computer programming. Web Intelligence and Agent Systems: An International Journal **4**(1), 77–97 (2006)
4. Christakopoulou, K., Radlinski, F., Hofmann, K.: Towards conversational recommender systems. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 815–824. ACM (2016)
5. Couso, I., Dubois, D.: Statistical reasoning with set-valued information: Ontic vs. epistemic views. International Journal of Approximate Reasoning **55**(7), 1502–1518 (2014)
6. Dacrema, M.F., Cremonesi, P., Jannach, D.: Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In: Proceedings of the 13th ACM Conference on Recommender Systems. pp. 101–109 (2019)
7. Evans, M., Moshonov, H., et al.: Checking for prior-data conflict. Bayesian analysis **1**(4), 893–914 (2006)
8. Harper, F.M., Konstan, J.A.: The Movielens datasets: history and context. Acm transactions on interactive intelligent systems (tiis) **5**(4), 1–19 (2015)
9. He, C., Parra, D., Verbert, K.: Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. Expert Systems with Applications **56**, 9–27 (2016)
10. Jugovac, M., Jannach, D.: Interacting with recommenders - overview and research directions. ACM Transactions on Interactive Intelligent Systems **7**(3), 10 (2017)
11. Kula, M.: Metadata embeddings for user and item cold-start recommendations. arXiv preprint arXiv:1507.08439 (2015)
12. Lewicki, M.S.: Bayesian modeling and classification of neural signals. In: Advances in Neural Information Processing Systems. pp. 590–597 (1994)
13. Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G.: Recommender system application developments: a survey. Decision Support Systems **74**, 12–32 (2015)
14. Mangili, F., Bonesana, C., Antonucci, A.: Reliable knowledge-based adaptive tests by credal networks. In: European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty. pp. 282–291. Springer (2017)
15. Mangili, F., Broggini, D., Antonucci, A., Alberti, M., Cimasoni, L.: A Bayesian approach to conversational recommendation systems. In: AAAI 2020 Workshop on Interactive and Conversational Recommendation Systems (WICRS) (2020)
16. Millán, E., Trella, M., Pérez-de-la Cruz, J.L., Conejo, R.: Using Bayesian networks in computerized adaptive tests. In: Computers and Education in the 21st Century, pp. 217–228. Springer (2000)
17. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Elsevier (1988)
18. Sarwar, B., Karypis, G., Konstan, J., Riedl, J., et al.: Analysis of recommendation algorithms for e-commerce. In: EC. pp. 158–167 (2000)