Towards a theoretical formalization of conversational recommendation

Tommaso Di Noia[1], Francesco Maria Donini[2], Dietmar Jannach[3], Fedelucio Narducci[1] and Claudio Pomo[1]

[1]Politecnico di Bari, via Orabona, 4, 70125 Bari, Italy
[2]Università degli Studi della Tuscia, via Santa Maria in Gradi, 4, 01100 Viterbo, Italy
[3]University of Klagenfurt, Universitätsstraße, 65-67, 9020 Klagenfurt am Wörthersee, Austria

Abstract

Tools that interact vocally with users are becoming increasingly popular in the market, boosting industry and academia interest in them. In such environments, conversational recommender systems succeed in guiding users in situations of information overload. Through multiple interactions with users, such systems ask questions, filter the catalog in a personalized manner, and suggest items that are of potential interest to the consumer. In this context, conversational efficiency in terms of the number of required interactions often plays a fundamental role. This work introduces a theoretical and domain independent approach to support the efficiency analysis of a conversational recommendation engine. Observations from an empirical analysis align with our theoretical findings.

1. Introduction and Motivation

System-generated recommendations have become a common feature of modern online services such as e-commerce sites, media streaming platforms, and social networks. In many cases, the suggestions made by the underlying recommender systems are personalized according to the user's tastes, needs, and preferences. In the most prominent applications of recommender systems, user preferences are estimated based on past user behaviors. However, there are several application domains where no past interaction logs are available or where the user's needs and preferences might differ each time the user interacts with the service (*e.g.,* restaurant recommendation for a party or a romantic dinner). In such application settings, a multi-turn, interactive recommendation process is required, where the system's goal is to learn about the user preferences to the extent that appropriate recommendations can be made. *Conversational Recommender Systems* (CRS) support such processes and these systems received increased attention in recent years [1, 2, 3, 4, 5, 6, 7].

The preference elicitation process in such settings can be implemented in different ways, ranging from predefined fill-out forms to natural language interfaces—see Jannach et al. [5] for an overview. In that context, a specific goal when designing a CRS is to minimize the effort for users by asking as few questions as possible, *i.e.,* to increase the *efficiency* of the dialog [8, 9, 10, 11, 12, 13, 14].

Today, research in the general area of recommender systems, and specifically area of CRS, is almost entirely empirical [15, 16, 17]. Such empirical studies are certainly important and insightful. However, little is known about the theoretical aspects of the underlying interactive recommendation processes. Unfortunately, theoretical questions regarding, *e.g.,* the computational complexity of determining a good or the best interaction strategy, can not be answered without a formal characterization of the overall problem.

With this work, we address this research gap and provide a theoretical model of conversational recommendation. The model is designed in a domain-independent way and aims to cover a wide range of realistic application scenarios. A conversational recommendation process is modeled as a sequence of states, where state transitions correspond to common *user intents* and *conversational moves* [18, 19, 20] that can be found in the literature.

Since our model is agnostic about the application domain and the algorithm that is used to select and rank the objects for recommendation (*i.e.,* the recommendation algorithm) it serves as a basis to analyze important theoretical properties of CRS.

The main contribution of this work[1] is the study of the computational complexity for finding an efficient conversational strategy in terms of number of dialog turns. In particular, we demonstrate that: *(i)* the problem of finding an efficient conversational strategy in terms of number of dialog turns is NP-hard, but in PSPACE; *(ii)* some specific factors of the item catalog influence the complexity of the problem; *(iii)* for a special class of catalogs, the upper bound lowers to POLYLOGSPACE. From a prac-

[1]An extended version of this work is available in Di Noia et al. [21].

MICROS@CIKM'22: Proceedings of CIKM Workshop on Mixed-Initiative ConveRsatiOnal Systems, October, 21, 2022, Atlanta, USA
tommaso.dinoia@poliba.it (T. Di Noia); tommaso.dinoia@poliba.it (F. M. Donini); tommaso.dinoia@poliba.it (D. Jannach); tommaso.dinoia@poliba.it (F. Narducci); tommaso.dinoia@poliba.it (C. Pomo)

tical perspective, our analysis leads to the observation that the efficiency of a conversation strategy is tied to the characteristics of the catalog. Observations from an empirical analysis on datasets based on MovieLens-1M support these theoretical considerations.

2. Model Description

In our theoretical framework we assume a retrieval-based item filtering approach, which is commonly used in critiquing-based and constraint-based approaches to recommendation [22, 23]. In critiquing approaches, users are presented with a recommendation soon in the dialogue and can then apply pre-defined critiques on the recommendations, e.g., ("less $$") [5]. In analogy to database-oriented approaches, we therefore use the term "query" when referring to positive user's preferences. Negative preferences are modeled as constraints on disliked item features. The retrieved items are then ranked according to any type of information, *e.g.,* the popularity of certain items. In order to carry a general analysis, in our approach we abstract from the details of this ranking.

To model the conversational recommendation process, we rely on the notion of *state* of a conversation, and what transformations this state can be subject to, depending on the interaction. For example, each preference expressed by the user leads to a change of the state of the conversation and may also imply a change in the set of recommendable items. This formalization through conversation states ultimately serves as a basis to study the efficiency of conversational strategies. The most efficient conversational strategies will minimize the number of states which must pass through to reach an end.

In our model, we mainly deal with the system-driven part of a conversation, where a conversation consists of a sequence of interactions.[2] The system can perform one of the following actions:[3]

1. ask the user to *fill in* (provide) a value for a particular feature under-specified so far (*e.g.,* the item color);
2. ask the user to enlarge a too narrow choice for a feature value (*e.g.,* to change the price limit);
3. ask for changing a feature value (*e.g.,* from green to red for the color feature);
4. upon rejection of one or more items, ask the user whether if there is a specific feature value of these items she dislikes (*e.g.,* a specific color) .

The user can react to the above system prompts with one of the following interactions:

(a) given one or more recommendations, the user can accept one of them, or reject them;

(b) the user can state that she dislikes every item where a features is filled with a particular value (*e.g.,* "*I don't like green cellphones*")

The formalization of the interaction process described by Di Noia et al. [21] allow us to establish the results mentioned above.

3. Empirical Analysis

3.1. Experimental Design

One main theoretical result is that the chosen conversation strategy (*protocol*) not only impacts the efficiency of a CRS, but that the efficiency also depends on the characteristics of the item catalog, *e.g.,* in terms of the number of available item features and the number of distinct values. We devised an *in-vitro* (offline) experiment using two protocols to empirically validate this result.

Protocols. A CRS may support two different ways (protocols) of how users can reject a recommendation made the system:

P1 - the user rejects the recommendation and the CRS does not ask the user to provide a specific reason, *i.e.,* a reason that refers to a disliked feature value. Examples of such more unspecific feedback—if any feedback is given at all—could be, "*I don't want to go to the Green Smoke restaurant*" or "*I don't want to see the movie American Beauty*" (for some reason, but I cannot explain this to a system);

P2 - the user rejects the recommendation and the CRS asks for a specific item characteristic (*i.e.,* feature value) she does not like at all. For example, green color for cellphones, sea-view restaurants, a particular movie director, etc. We assume that a user will truthfully answer such questions.

Hypotheses. Based on our theoretical results, we formulate two hypotheses, where the difference lies in the characteristics of the item catalog.

H1 We do *not* expect a strong difference in terms of efficiency between P1 and P2 when the items in the catalog have few features with a large number of distinct values.

[2] In constraint-based and critiquing-based systems the recommender system usually drives the conversation in an initial preference elicitation phase. In typical implementations of such systems, the user can however also take the initiative and, for example, request recommendations at any time or proactively revise their preferences.

[3] We note that in the area of Knowledge Representation, slot unfilling (*e.g.,* retract any requirement about colors) could be considered a special case of knowledge *contraction* [24], while slot change (*e.g.,* from green to red) is a form of *revision* [25].

H2 We *do* expect a strong difference in terms of efficiency between P1 and P2 when the items in the catalog have several features with few distinct values.

Experiment Specifics. In our experiment, we simulate the above-mentioned protocols and vary the underlying item catalog as an independent variable.

Efficiency Metric. As commonly done in the literature [14, 17], we use the number of questions (*NQ)* the CRS asks before the user accepts a recommendation as an efficiency measure. Fewer questions indicate higher interaction efficiency.

Dataset and Catalog Description. We rely on the widely used MovieLens-1M (ML-1M) dataset for our experiment, which we enrich with item features using DBpedia [26]. The resulting dataset comprises 3,308 items with 279 unique features. From this dataset, we create two versions to test our hypotheses.

- *Itemset1 (IS1)* has only a few features but with a larger number of distinct values. It is designed to support H1 (we do not expect a strong difference in terms of efficiency between P1 and P2 *when the items in the catalog have few features with a large number of distinct values*).
- *Itemset2 (IS2)*, in contrast, has a larger number of features, but each of them only has a few distinct values and is designed to support H2 (we expect a strong difference in terms of efficiency between P1 and P2 *when the items in the catalog have several features with few distinct values*).

Specifically, to make the datasets sufficiently different and such that they reflect the characteristics described in our research hypotheses, IS1 and IS2 have 4 and 10 features respectively for each item. For each feature of IS1 there are from 1,500 to 2,500 distinct values, whereas there are about 100 values onn average for IS2. To better focus on the main goals of our experiment, we make the simplifying assumption that all features can have only one value. Accordingly, we replaced set-valued features (*e.g.,* the movie cast) with a single value randomly chosen among them.

Simulation Procedure. We simulate the part of a conversation between a user and a CRS where the system drives the interaction by asking the user about preferred item features and making recommendations for items[4]. We assume that the simulated user has certain pre-existing preferences regarding item features and truthfully responds to system questions about these preferences. When provided with a recommendation, the user either rejects it, which means that the dialog continues, or accepts it, and the dialog ends. The CRS in our simulation implements one of the described conversation strategies, P1 or P2.

Note that in our experiment we simulate a user cold-start situation, *i.e.,* we are not taking any long-term user profile into account during the dialog. In order to simulate the response of a user, we first select a set of positively-rated items (PRI) for each user. This set consists of those items in the dataset that the user has rated with a value that is greater or equal to their average rating in the MovieLens dataset. We use this set PRI for two purposes. First, we simulate a dialog for each element I of PRI as an "ideal" item (that the user will accept). Second, we use the items in PRI to determine the pre-existing preferences of a user and simulate their answers to the questions posed by the system. Therefore, if the user previously liked *action* and *romantic* movies, the set of pre-existing preferences contains only these values.

When the simulated dialog with a defined ideal item $\hat{I}$ starts, the system will ask a question on a feature *e.g.,* "*What is your favorite genre?*". The simulated user will then respond by choosing a value from the set of values for that feature occurring in PRI. In our simulation, the user cannot answer with a value that is not present in any recommendable object. After each user answer, the set of recommendable items $\mathcal{S}$ is updated by the CRS according to her answer. A recommendation is shown when the system has no more questions to ask. This situation may occur when: *(i)* preferences on all the features have been expressed, *(ii)* only one item on the catalog is consistent with the user preferences. The user rejects the recommendation when I is not present in the list of recommended items. If the recommendation is rejected, the recommended items are removed from the catalog and the system starts again posing questions to the user. It is noteworthy that, since we may have more than one item in the catalog described by the feature values selected by the user during the dialog, the final recommendation may contain also a set of items.

In case of rejection, protocols P1 and P2 lead to different system reactions. In case of P1, the system starts querying the user again, beginning with a first randomly selected feature. The values selected during the previous interactions are discared. In protocol P2, in contrast, the user declares one of the feature values as *disliked* for the recommended items.

When the recommendation succeeds—*i.e.,* when the ideal item I is in the list of recommendations—the dialog is successfully ended and the simulation continues with a new dialog for another user and/or target item. The simulation ends when a dialog was simulated for each

[4]Other parts of the conversations may include greetings or chit-chat. For a catalog of common user intents in CRS, see [18].

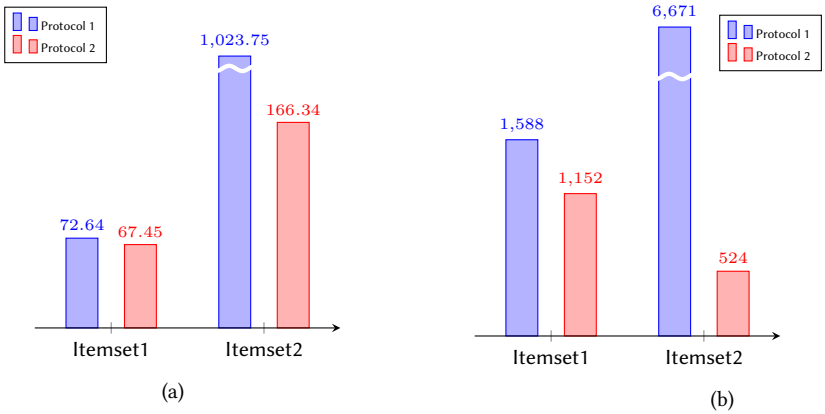

Figure 1: (a) the average number of Questions per Configuration; (b) the maximum number of Questions per Configuration

element in PRI for every user.

3.2. Results and Discussion

We applied protocols *P1* and *P2* both for itemset *IS1* and *IS2*, and we counted the Number of Questions (NQ) required to reach the test item in each configuration. Figure 1 summarizes the results. As expected from the theoretical analysis, with IS1 we observe minor differences between the two protocols in terms of number of requird questions made by the system. More specifically, P1 needs 72.64 as the average number of questions with IS1, whereas P2 needs 67.45. The difference is however huge for IS2 where P1 needs more than 1,000 questions on average to reach the test item, while P2 requires around 166 questions (Fig. 1a). Hence, we can confirm that when the items in the catalog have many features with a smaller number of distinct values, the efficiency of P2 grows drastically compared to P1. Also the *maximum* number of questions confirms this different efficiency for IS1 and IS2 (Fig. 1b). We note that NQ in absolute terms is very large—even in the best combination (P2 with IS1), the number of questions is close to 70, which sounds too high for practical applications. Recall, however, that in this experiment we implemented a worst-case scenario and our experiment used an unrealistic setting on purpose. In our scenario the recommendation task is deliberately difficult:

- for each dialog, there is only one test item (true positive);
- the CRS works in cold-start condition without any user profile;
- the CRS does not implement a cut-off on the number of questions to ask the user.

In conclusion, our experimental evaluation results align with our theoretical findings, thus providing support for our research hypotheses H1 and H2. In other words, our simulation confirmed what was foreseen by the theoretical analysis: the difference between protocol P1 and protocol P2 shows up clearly only in a dataset with many features with a small set of different values.

4. Summary and Outlook

With this work, we contribute to a better understanding of theoretical properties of conversational recommendation problems and we specifically address questions related to the computational complexity of finding efficient dialog strategies. One main insight of our theoretical analysis—which was also confirmed by an *in-vitro* experiment—is that when designing an efficient conversation strategy, we must always consider the characteristics of the item catalog. More specifically, we demonstrated that when a few features characterize the items in the catalog with a large number of distinct values, the critiquing strategy based on asking the user about a disliked characteristic of the recommended item does not give any significant advantage in terms of user effort. Conversely, when the catalog is composed of items with several features with a few distinct values, a critique strategy based on item features can drastically reduce the user effort for reaching a liked recommendation. On a more general level, we hope that our work might help to stimulate more theory-oriented research in this area, leading us to a better understanding of the foundational properties of this essential class of interactive AI-based systems. In future research, we will investigate the explicit consideration of individual long-term user preferences in the interactive recommendation process.

Acknowledgement

The authors acknowledge partial support from the projects PASSEPARTOUT, ServiziLocali2.0, Smart Rights Management Platform, BIO-D, and ERP4.0.

References

[1] K. Christakopoulou, F. Radlinski, K. Hofmann, Towards conversational recommender systems, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 815–824.

[2] T. Shen, Z. Mai, G. Wu, S. Sanner, Distributional contrastive embedding for clarification-based conversational critiquing, in: Proceedings WWW '22, 2022, pp. 2422–2432.

[3] K. Zhou, W. X. Zhao, S. Bian, Y. Zhou, J.-R. Wen, J. Yu, Improving conversational recommender systems via knowledge graph based semantic fusion, in: Proceedings KDD '20, 2020, pp. 1006–1014.

[4] C. Gao, W. Lei, X. He, M. de Rijke, T.-S. Chua, Advances and challenges in conversational recommender systems: A survey, AI Open 2 (2021) 100–126.

[5] D. Jannach, A. Manzoor, W. Cai, L. Chen, A survey on conversational recommender systems, ACM Comput. Surv. 54 (2021) 1–26.

[6] W. Cai, Y. Jin, L. Chen, Impacts of personal characteristics on user trust in conversational recommender systems, in: Proceedings CHI '22, 2022, pp. 489:1–489:14.

[7] F. Radlinski, K. Balog, F. Diaz, L. Dixon, B. Wedin, On natural language user profiles for transparent and scrutable recommendation, in: Proceedings SIGIR '22, 2022, pp. 2863–2874.

[8] G. Adomavicius, Y. Kwon, New Recommendation Techniques for Multicriteria Rating Systems, IEEE Intelligent Systems 22 (2007) 48–55. doi:10.1109/MIS.2007.58.

[9] P. Viappiani, C. Boutilier, Regret-based optimal recommendation sets in conversational recommender systems, in: RecSys '11, 2009, pp. 101–108.

[10] T. Mahmood, F. Ricci, Improving recommender systems with adaptive conversational strategies, in: Proceedings of the 20th ACM conference on Hypertext and hypermedia, 2009, pp. 73–82.

[11] J. Reilly, J. Zhang, L. McGinty, P. Pu, B. Smyth, A comparison of two compound critiquing systems, in: IUI '07, 2007, pp. 317–320.

[12] P. Grasch, A. Felfernig, F. Reinfrank, Recomment: Towards critiquing-based recommendation with speech interaction, in: Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, 2013, pp. 157–164.

[13] F. Narducci, M. de Gemmis, P. Lops, G. Semeraro, Improving the user experience with a conversational recommender system, in: International Conference of the Italian Association for Artificial Intelligence, Springer, 2018, pp. 528–538.

[14] A. Iovine, F. Narducci, G. Semeraro, Conversational recommender systems and natural language: A study through the converse framework, Decision Support Systems 131 (2020) 113250.

[15] A. Iovine, P. Lops, F. Narducci, M. de Gemmis, G. Semeraro, An empirical evaluation of active learning strategies for profile elicitation in a conversational recommender system, Journal of Intelligent Information Systems (2021) 1–26.

[16] V. Bellini, G. M. Biancofiore, T. D. Noia, E. D. Sciascio, F. Narducci, C. Pomo, Guapp: A conversational agent for job recommendation for the italian public administration, in: EAIS, 2020, pp. 1–7.

[17] D. Jannach, Evaluating conversational recommender systems, Artificial Intelligence Review (2022). doi:https://doi.org/10.1007/s10462-022-10229-x.

[18] W. Cai, L. Chen, Predicting user intents and satisfaction with dialogue-based conversational recommendations, in: UMAP '20, 2020, p. 33–42.

[19] R. Reichman, Getting computers to talk like you and me, MIT Press, Cambridge, Massachusetts, 1985.

[20] D. Jannach, Finding preferred query relaxations in content-based recommenders, in: Intelligent techniques and tools for novel system architectures, Springer, 2008, pp. 81–97.

[21] T. Di Noia, F. M. Donini, D. Jannach, F. Narducci, C. Pomo, Conversational recommendation: Theoretical model and complexity analysis, Information Sciences —in press— (2022). doi:https://doi.org/10.1016/j.ins.2022.07.169.

[22] A. Felfernig, G. Friedrich, D. Jannach, M. Zanker, Constraint-based recommender systems, in: Recommender systems handbook, Springer, 2015, pp. 161–190.

[23] D. Jannach, Advisor Suite – A knowledge-based sales advisory system, in: ECAI '04, 2004, pp. 720–724.

[24] S. Colucci, T. Di Noia, E. Di Sciascio, F. M. Donini, M. Mongiello, Concept abduction and contraction for semantic-based discovery of matches and negotiation spaces in an e-marketplace, Electron. Commer. Res. Appl. 4 (2005) 345–361.

[25] P. Gärdenfors, Belief revision and knowledge representation, in: Proceedings of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge, 1996, p. 117.

[26] T. D. Noia, V. C. Ostuni, P. Tomeo, E. D. Sciascio, Sprank: Semantic path-based ranking for top-N recommendations using linked open data, ACM Trans. Intell. Syst. Technol. 8 (2016) 9:1–9:34.