CoRE: A Cold-start Resistant and Extensible Recommender System

Mostafa Bayomi
ADAPT Centre
Trinity College Dublin, Ireland
mostafa.bayomi@adaptcentre.ie

Annalina Caputo
ADAPT Centre
Trinity College Dublin, Ireland
annalina.caputo@adaptcentre.ie

Matthew Nicholson
ADAPT Centre
Trinity College Dublin, Ireland
matthew.nicholson@adaptcentre.ie

Anirban Chakraborty
ADAPT Centre
Trinity College Dublin, Ireland
anirban.chakraborty@adaptcentre.ie

Sèamus Lawless
ADAPT Centre
Trinity College Dublin, Ireland
seamus.lawless@adaptcentre.ie

ABSTRACT

In this paper, we propose the Cold-start Resistant and Extensible Recommender (CoRE), a novel recommender system that was developed as part of collaborative research with Ryanair, the world's most visited airline website. CoRE is an algorithmic approach to the recommendation of hotel rooms that can function in extreme cold-start situations. It is a hybrid recommender that blends elements of naïve collaborative filtering, content-based recommendation and contextual suggestion to address the various shortcomings which exist in the underlying user and product data. We evaluated the performance of CoRE in a number of scenarios in order to assess different aspects of the algorithm: personalization, multi-model and the resistance to the extreme cold-start situations. Experimental results on an authentic, real-world dataset show that CoRE effectively overcomes the different problems associated with the underlying data in these scenarios.

CCS CONCEPTS

• **Information systems → Recommender systems**;

KEYWORDS

Contex-aware recommendations, recommendation explanation

ACM Reference Format:
Mostafa Bayomi, Annalina Caputo, Matthew Nicholson, Anirban Chakraborty, and Sèamus Lawless. 2019. CoRE: A Cold-start Resistant and Extensible Recommender System. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, Article 4, 4 pages. https://doi.org/10.1145/3297280.3297601

1 INTRODUCTION

Recommender systems provide their users with the recommendation of items that are relevant to their needs or aligned with their interests. They can be broadly categorised in: (1) Collaborative Filtering [5] whose predictions are based upon the analysis of the behaviour and activity of similar users and (2) Content-based recommenders [4], whose suggestions come from the comparison of the preferences of the individual user with some descriptive features of the items to be recommended. However, both approaches suffer from well known problems. The former has to tackle with the cold-start problem, caused by the lack of knowledge about new users or items, and data sparsity. On the other side, recommending items only on the basis of the individual user's past history gives rise to the phenomena of overspecialisation and to unreliable recommendations with cold-start users. As recommender systems evolved, they started to incorporate a range of different information about their users, in an attempt to provide more personalised recommendations. Context-aware recommender systems make use of contextual information, such as time, location, season, accompaniers of the user in a trip situation, etc., in addition to the typical information on users and items [1]. They are built upon the notion that user preferences change under different contexts.

A problem common to all of the aforementioned techniques is that they are static in two ways: (1) Once the recommendation engine is built, the integration of new sources of information or dimension requires not feasible extension or significant changes to the underlying algorithm; (2) Once the user profile has been established, changing this profile is deemed to be difficult [2].

In this paper, we propose CoRE, a real-time, extensible and cold-start resistant recommender system. CoRE was developed as part of collaborative research with Ryanair, the world's most visited airline website (over 500 million "uniques" annually). CoRE was designed to provide personalized, contextually appropriate hotel room recommendations on the Ryanair Rooms. Due to the historical nature of data sharing with their third-party service providers, Ryanair now finds itself in an extreme cold-start situation from a recommender system perspective, where they have very limited information about their users' behavior with regard to hotel booking. Furthermore, this data includes no ratings or reviews and, therefore, it does not capture the opinions and preferences of users towards their previous bookings. Hence, we developed CoRE, a new approach that is resistant to such cold-start situations. CoRE combines a number of different techniques to address the shortcomings in the user and item data. This paper is organized as follows: Section 2 describes the challenges and limitations in the underlying data, Section 3 describes the design of CoRE and the different models we use to tackle the challenges in data and to overcome the limitations of

SAC '19, April 8–12, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-5933-7/19/04.
https://doi.org/10.1145/3297280.3297601

current approaches. Sections 4 describes the experimental design and the CoRE evaluation.

2 DATA AND CHALLENGES

Ryanair currently uses a set of third-party providers to deliver hotel search and booking to their customers. When a hotel booking is made by a Ryanair customer, all this data is saved on the provider's side and only brief summary information is sent to Ryanair (e.g. hotel ID, user ID, check-in and check-out dates), with no information on user feedbacks and reviews. The data that Ryanair held at the point when this research was conducted has 29, 704 hotel bookings, 11, 683 unique hotels that have at least 1 booking and 20, 223 users who have made at least 1 hotel booking. Since Ryanair relies on several third-party providers for hotel bookings, there is no uniqueness in the dataset. Hence, there is a need to consolidate the hotel data from different providers in order to have a single static inventory from which to generate our recommendations.

Data Consolidation. We saved all data associated with hotel bookings from the different providers in a single database table. Each record has a different sequence of hotel ids according to its provider. To consolidate these records, we used the static hotels repository provided by Expedia[1]. This repository contains approximately 219 thousand hotels from all over the world and contains a range of information about each hotel. We mapped each hotel in our bookings record to the relevant entry in the Expedia static inventory by using a configurable Information Retrieval approach that takes the name and the location of the hotel (from the third-party provider) as an input and searches for it in Expedia's repository. Once the hotel is found, its record is updated in our consolidated inventory by adding a new field that contains the id of that hotel in Expedia. After the mapping, our dataset contained 18, 700 bookings, 9, 279 unique hotels with at least 1 booking, and 17, 384 users who have made at least 1 hotel booking.

Flight Bookings. Since hotel bookings data has inadequate information about each booking, we exploit Ryanair abundance of user data regarding flight bookings. Ryanair clusters their users into ten different segments using a KNN clustering approach based on the commonalities among users' history while booking their flights. Each user segment contains users who are "similar" to each other. Examples of segments are: "Business Traveler" and "Student Backpackers". In addition to this segmentation, Ryanair has ten different trip types that can be assigned to each flight booking. The trip type represents the context in which the booking took place. Examples of trip types are: "Adult Sun Break" and "Family with infants". Using such flight bookings data, we performed a further mapping to enrich hotel bookings with more information. Using the user id, hotel location, flight destination and times of both bookings (flight time and hotel check-in time), we searched around 15 million flight bookings to map each users' hotel booking with their flight booking. This mapping was carried out in order to assign user segment and trip type to each user at the time of each hotel booking. The total number of bookings made by users assigned to a segment is 1, 434, while the total number of bookings whose trip type was known during the room booking is 1, 962.

[1]https://github.com/ExpediaInc/ean-pc-dbscripts

User ratings. The available data does not have any explicit ratings from users, it merely has the act of a Ryanair user making a booking. Hence, we have to rely solely on implicit feedback, where the user's preferences are inferred by observing their actions within the system, such as booking a hotel room.

3 CORE DESIGN

To overcome the challenges associated with the data and to address the drawbacks of current approaches, we built CoRE, a novel approach that blends content-based recommendation, naïve collaborative filtering and contextual information in order to make personalized hotel recommendations. In order to design a content-based recommendation technique, features associated with hotels need to be identified. In the underlying hotels repository (from Expedia) there are 381 features associated with hotels, such as: Free Wi-Fi, Free Breakfast, Restaurant, Hair dryer, etc. We use these features to build the different models that underlie the recommendation process in CoRE, as discussed below.

3.1 Data Modelling

CoRE is capable of recommending items to users regardless of the granularity of information currently held about them. It is designed as a flexible recommender system that incorporates different models that influence the recommendation process based on the availability of data associated with each model. As a user interacts with the system and more data is gathered about individuals, and the community of users, the delivered recommendations continue to improve. It uses a set of discrete features from hotel descriptions to characterize the hotel and build different models. Each model is built based upon a weighted vector of features of the hotels that have been previously booked. The weights denote the importance of each feature to the model. In this research, three models are built: (1) User Model, (2) Segment Model and (3) Contextual Model.

User Model. Hotel features are taken from the underlying static inventory and are used to build a vector of descriptive features which are used to discriminate between hotels in the inventory. The values for each feature are stored for each hotel. Then, for each user who has previously made a booking, the user model is constructed as a weighted feature vector of hotels previously booked by that user. Each feature is represented as a weight of the co-occurrences of that feature in the model.

Segment Model. As mentioned in Section 2, we enriched hotel bookings with the segment that the user, who made the booking, belongs to and each segment contains users who are similar in their travel patterns. In our approach, we build a model for each segment. A segment model is constructed as a weighted feature vector of hotels that have been booked by all users who belong to that specific segment. Each feature is represented as a weight of the co-occurrences of that feature in the model. We exploit this model to influence the recommendation process for the user based upon the preferences of people in the same segment. This is a form of collaborative filtering using "the wisdom of the crowd". If the user's segment is not known at the hotel booking time, the segment model is omitted from the recommendation process.

Contextual Model. Contextual information is an important aspect when recommending hotels to users. With continuous change in users' preferences and fluctuations in their needs, relying on user's historical data or even collaborative preferences will produce unreliable recommendations that do not reflect the user's current needs. Contextual information associated with the user can represent a range of different dimensions. Since CoRE is flexible in incorporating information, a variety of models can be used based on their availability, to influence the recommendation process and thus reflect the different contexts associated with the user's trip. In this research, we use the trip type model as it is the only available contextual information in the data at hand. A trip type model is constructed as a weighted feature vector of the hotels that have been booked by all users who were in that trip type. The trip type associated with this user can be captured in two ways: if the user has booked a flight, then we use the trip type associated with this flight booking. If the user is just booking a hotel without a flight or if the user is anonymous to the system, inferring such context is relatively easy by using the user's input to the system.

Hybrid Recommender. All generated models are exploited, when they exist for a particular user, to generate recommendations: (1) The user model built from previous bookings; (2) The segment model built from all bookings made by users in the same segment; (3) The context model(s) built from all bookings made by users while in that same context. Combining these aspects delivers an algorithm capable of recommending hotels to users even with limited or no previous information about them. When a user makes a request to CoRE, the recommender retrieves the feature vectors related to that user as follows: If the user has previous hotel bookings in the dataset, the algorithm retrieves the feature vector (user model) that has been built based upon these previous bookings. If the user belongs to a segment based on his/her flight booking history, the algorithm retrieves the feature vector for that segment. After identifying or inferring the type of the user's trip, the algorithm retrieves the feature vector for that trip type model. After that, the centroid vector of the retrieved models is calculated. This vector is considered a central representation of the features in the retrieved models. The algorithm then starts to assign a score to each hotel within the target city where the user is travelling to and that matches the criteria the user has specified[2]. The score is the cosine similarity between each hotel vector and the centroid vector. This similarity score can be seen as the blended, weighted input of the different models. Hence, this score is considered a balance between the preferences of the user, the segment that he/she belongs to and people who have experienced the same contexts as the user is currently in. The hotels are then ranked based upon their similarity scores and returned to the user as a ranked list in order to choose a hotel to book. As the centroid vector in CoRE is considered a central representation of the prominent features of models that influence the recommendation process, these prominent features are considered a balance between the preferences of these models. Hence, using these prominent features can provide a sensible explanation to the user of why a hotel is recommended for her. A simple example: "As you are travelling with infants (trip type), this hotel provides free infant beds and free babysitting". Such an explanation would give the user a sense of inclusion and ultimately trust in the recommendations generated. In addition to the models currently used to recommend hotels, other models, such as user location, seasonality, and price sensitivity, can be easily added to impact the recommendation process.

[2]Criteria such as price range or star rating.

3.1.1 Real-Time Recommendation. Each built model (User, Segment, Contextual) is saved in the database to be easily updated, and timely retrieved and used in the recommendation process. CoRE enables real-time model updates to ensure that the recommendations provided are timely and appropriate. With each interaction from the user with the system (i.e. booking), the system updates the various models at run-time using the feature vector of the booked hotel. The features in question are located in each model and updated to reflect the new information. When a request is sent to our recommender, the feature vectors are retrieved for each available model and their centroid can be quickly generated at run-time.

3.1.2 Model Weighting and Feature Selection. As described above, CoRE relies on different models to recommend hotels to the target user. Each model provides different clues related to the potential interest or end goal of the user. Certain models can carry variable influence, depending upon the data situation. The weight assigned to each model indicates the weight its features vector is given while building the centroid vector. Applying model weighting in building the centroid vector and thus in the recommendation process is considered a strength of CoRE over traditional systems. Moreover, CoRE gives users control over the recommended items by allowing them to assign the weight to each model. This can be assigned by the user through a set of sliding bars that indicate the percentage of the influence each model has on the recommended hotels. However, to alleviate the burden of manually setting model weights, we experimented with different values for these weights and found that setting the user model to 0.8 and the other two models to 0.1 produced the best results. This finding shows that the contribution of the user model should be set, by default, more than any other model and then the user can modify these values according to her needs. This finding is not surprising since user model represents the personal preferences of the user while the other models represent other people's preferences.

In addition to the model weighting, also the hotel features can have a different impact on the overall recommendation process, since some features are deemed more relevant than others when booking a hotel. There are a range of feature weighting and selection methods which can be used. In this research, we use the filter-based feature weighting algorithm ReliefF [3]. ReliefF copes with multi-class datasets. The simple intuition behind the ReliefF algorithm is that a good feature has little within class variance and generous amounts of between-class variance. A bad feature is characterized by within-class and between-class variances of similar magnitude. We trained ReliefF over the hotel categories generated by the star rating classification used in the hotel repository. We set the threshold to zero and select features that have a weight higher than zero. After applying ReliefF, the total number of features used in the recommendation was reduced from 381 to 233.

4 EVALUATION

We carried out various offline simulated user experiments in order to evaluate the efficacy of CoRE in recommending hotel rooms. We adopted the "leave-one-out" cross validation approach. In each run, we generated a list of hotels in the target city of the test booking where hotels are sorted in descending order by their predicted value. For the evaluation, we adopted the Mean Percentile Rank (MPR), which is used to measure the user satisfaction of items in an ordered list. We compared CoRE accuracy against the ranking approach that is currently used by Ryanair in their rooms booking website. This approach sorts all hotels in the target city based on Expedia sequence number, which reflects the transactional data from the last 30 days. This value ranks hotels with 1 indicating the best-performing hotel and others following in ascending numerical order. We feel that this is a strong baseline, as the rankings are based upon the best performing hotels for each city from the entire Expedia inventory across all its websites. We carried out different experiments in order to assess the performance of CoRE in different scenarios. Table 1 reports the results of this evaluation. The first scenario (*User*) aims to evaluate the personalisation aspect of CoRE, regardless of the user segment or trip type. We selected users who have more than two bookings in their profiles in order to build the user model from at least two bookings. There are 65 users in the dataset who meet this criterion. Form the results we can see that CoRE outperforms the competing baseline. Running a significance test with 95% confidence level showed that the difference between the accuracy of CoRE and the baseline is statistically significant. We also evaluated the impact of feature weighting on the recommendation process (CoRE_{FW}). From the results, although this impact is not statistically significant we can see that feature weighting enhances the performance of CoRE. In the second scenario (*User + Segment + Contextual*) we evaluate CoRE on users who have previous bookings (User), are assigned to a segment (Segment) and their trip type was known at the time of booking (Contextual). In this scenario, we evaluate the personalization, collaboration and the contextual aspects of the recommender. There are only 22 users in the dataset who have all these elements in their profiles. Although our approach allows users to control the relative contribution of each model, in this offline evaluation we set the user model to 0.8 and the other two models to 0.1 as experimenting with these values produced the best results. It is worth noting that in this experiment, when we remove a user's booking from the user's model (leave-one-out) we also remove this booking from the other two models so that this test booking would have no influence on the recommendation process. The results show that CoRE significantly outperforms the baseline system and using feature weighting enhances its performance. In the last scenario (*Segment + Contextual*) we assess the performance of CoRE in the extreme cold-start situation where the user is new to the system and has no previous bookings in her profile. The recommendations in this scenario are based solely on the information from the other two models (segment and trip type). We selected users who only have one booking in their profile, belong to a segment and their trip type is known. We use this booking as the test booking. We applied equal weighting to the two models. CoRE performs well in the extreme cold-start situations and its accuracy is statistically significantly higher than the baseline. However, we note that, in this scenario, the performance of CoRE is slightly improved when feature weighting is not used. An analysis of the two models highlighted a very scattered distribution within-model of the segments and trip types. We argue that with more bookings and a better fitted hotel classification method for the feature selection, feature weighting would be more reliable in the cold-start scenario.

Table 1: System performance using different model configurations.

Used Models	# Users	Approach	MPR (%)
User	65	Sequence	59.56
		CoRE	28.55
		CoRE_{FW}	22.67
User+Segment+Contextual	22	Sequence	63.50
		CoRE	22.04
		CoRE_{FW}	18.27
Segment+Contextual	1,385	Sequence	58.70
		CoRE	38.66
		CoRE_{FW}	39.23

5 CONCLUSIONS

In this paper we presented CoRE, a real-time hybrid recommender system that blends content-based recommendation, naïve collaborative filtering and contextual information in order to make personalized hotel recommendations. The recommender was developed as part of collaborative research with Ryanair. CoRE uses a set of discrete features from hotel descriptions to characterize hotels and build user, segment and contextual models. We evaluated the performance of CoRE against the recommendation approach currently used by Ryanair on an authentic, real-world hotel booking dataset from Ryanair. The results showed that CoRE significantly outperformed the baseline system and performed well in the extreme cold-start situations where the user is new to the system or has no previous bookings.

ACKNOWLEDGMENTS

This work was supported by the European Unions Horizon 2020 research and innovation programmender the Marie Skodowska-Curie grant agreement No.: 713567 and by the ADAPT Centre for Digital Content Technology, which is funded under the Science Foundation Ireland Research Centres ProgrammeGrant 13/RC/2106) and is co-funded under the European Regional Development Fund.

REFERENCES

[1] Mostafa Bayomi and Séamus Lawless. 2016. ADAPT_TCD: An Ontology-Based Context Aware Approach for Contextual Suggestion.. In *TREC 2016, Contextual Suggestion Track*.
[2] Robin Burke. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction* 12, 4 (01 Nov 2002), 331–370.
[3] Kenji Kira and Larry A Rendell. 1992. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, Vol. 2. 129–134.
[4] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2011. *Content-based Recommender Systems: State of the Art and Trends.* Springer US, 73–105.
[5] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. *Collaborative Filtering Recommender Systems.* Springer Berlin Heidelberg, Berlin, Heidelberg, 291–324.