

Open Access

Dynamic Selection of Recommender Systems: An Application to Tourism

Samuel Arleo and Bruno Silva*

Ubiwhere, Travessa Senhorda Barrocas, Aveiro, Portugal

Abstract

Recommender Systems are software tools and techniques that seek to suggest items that are likely of interest to a particular user. These systems are a key part of most e-commerce applications, as they ease users to find products that meet their needs while improving sales. Several approaches have been created to determine the users' preferences by working with different sources and types of information. Collaborative filtering uses the history of ratings, content and knowledge based recommenders work with the features of items, context aware systems provide suggestions based on the situation parameters or conditions that surround the user, while demographic filtering utilizes user's demographic characteristics. Additionally, there are hybrid approaches that fuse two or more techniques in order to overcome the shortcomings of each method.

In this work an application of dynamic selection to the recommender systems field is studied. This selection strategy, taken from Multiple Classifier Systems, consists of selecting a specific set of classifiers for each test pattern. To adapt this notion to the context of this research, it was proposed a hybrid system that dynamically seeks to select the best recommendation method in each prediction.

After carrying out experiments, the application of dynamic selection did not provide any significant improvement to recommendations. However, the inclusion of demographic and contextual information in a hybrid content-based basis increased the accuracy of the system considerably. The final solution was evaluated using datasets containing reviews of hotels and books. Results showed that the recommender is capable of working in tourism related scenarios and that can also be parameterized to other recommendation problems as long as content, demographic or contextual features are available.

Keywords: Recommendation system; Dynamic classifier selection; Hybrid content-based recommender

Introduction

Recommender systems were initially inspired by a characteristic of the social behavior of human beings where individuals tend to rely on others opinions in making routine decisions, for instance, when buying books, clothing, movies or even when selecting employees for a new job [1].

The study of recommender systems is relatively new compared to search engines and databases [1]. They emerged in the 1990s when the idea of suggesting products based on other users opinions emerged along with the term "collaborative filtering", introduced by Goldberg et al. [2] in the work titled "Using collaborative filtering to weave an information tapestry". Thereafter, several systems were developed using collaborative filtering and knowledge based techniques. The community of people working on personalization systems using approaches from different domains came to the conclusion they were addressing aspects of a larger problem that quickly became known as recommender systems [3].

The field evolved with the contributions of researchers and companies, as these systems emerged into a rapidly expanding Internet business climate [3]. Nevertheless, it was in 2006, when Netflix offered \$1 million prize for a 10 percent improvement of the root mean squared error (RMSE) of their current recommendation algorithm, that the research of this field experienced a significant progress.

Nowadays recommender systems have a well-established community of researchers and are included in conferences and workshops such as the Association of Computing Machinery's (ACM) Conference Series on Recommender Systems (RecSys), established in 2007; User Modeling, Adaptation and Personalization (UMAP); Intelligent User Interfaces (IUI); among others [1]. Recommender systems have also been used in the tourism industry to emulate the services offered by tourist agents, where users seek for advice to select travel destinations under certain time and budget constraints [4]. The tourism industry is one of the most important sectors of Business-to-consumer (B2C) e-commerce. Studies made by the European Travel Commission (ETC) revealed that in the past decade, in developed countries, the Web was the primary source of information for people when searching or booking suitable travel destinations. It is to be expected that recommender applications have an important role in this context supporting tasks like information search, decision making and package assembly [5].

Motivation

Physical stores show the customer a fraction of the products that exist because of space limitations. As a consequence, they might be forced to shorten the stock, offering, for instance, the most popular items making others to remain unnoticed. For this reason, users requirements may not be satisfied, especially when a client looks for new or more specific products. On the other hand, online stores are able to make available the entire universe of alternatives to the user [6].

Physical stores provide only the most popular items, while the online ones are able to make accessible the entire range of items. Then,

*Corresponding author: Bruno Silva, Ubiwhere, Travessa Senhor das Barrocas, Aveiro, Portugal, Tel: +351 234 484 466; E-mail: bsilva@ubiwhere.com

Received December 02, 2017; Accepted December 18, 2017; Published December 25, 2017

Citation: Arleo S, Silva B (2017) Dynamic Selection of Recommender Systems: An Application to Tourism. J Tourism Hospit 6: 325. doi: 10.4172/2167-0269.1000325

Copyright: © 2017 Arleo S, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

the vast amount of alternatives makes recommender systems necessary, as it is impractical for users to explore all the options [6].

Recommender systems represent a way of increasing the number of products sold because the suggested items are likely to suit the user's requirements. This also happens in non-commercial applications, where the selection of an item (e.g., listening to a song, watching a video, etc.) is equivalent to a purchase. Moreover, they can improve the diversity of the acquired products when the user searches for certain items that might be hard to find without the aid of the system, as they go beyond recommending popular ones. Additionally, they can increase the user satisfaction, fidelity and provide a better understanding of what the user wants. Another purpose of recommender systems is to help individuals who lack the sufficient personal experience or competence to make an appropriate choice [1].

In the tourism field, recommender systems usually make suggestions based on contextual information such as the user's location [5]. Although contextual-awareness is ubiquitous in most of the existing tourism recommender systems, other factors may have an impact on users' decisions when selecting a place to spend their vacations. Some are the user's demography and preferences. Thus, considering different types of information, when suggesting a touristic experience might produce truly personalized recommendations.

In this work, different techniques that exploit the previously mentioned information sources are someway applied by using CF, CB, context aware and demographic filtering.

Recommender systems that combine different techniques such as CB and CF are called hybrid recommender systems. The idea is to produce some synergy by taking advantage of the strengths of each component and overcome the shortcomings of the others [3,7].

The systems reviewed in the literature are static approaches: they address all scenarios with the same set of techniques [7]. In Multiple Classifier Systems (MCS), applying the same set of classifiers for all the test samples is called static selection, while choosing a set of experts for each test sample corresponds to a dynamic selection [8].

Bibliographic Review

Recommender systems are software tools and techniques that provide suggestions of items that are likely of interest for a particular user. They emerged in the 1990s when companies and researchers developed the first collaborative filtering and knowledge based recommender systems. Since then, researchers have continuously applied new approaches to take advantage of new information sources such as social data and location [3]. These systems support users in their online decision making and are primarily directed to individuals who lack the knowledge to evaluate the possibly high amount of options [1,3].

The development of recommender systems is inspired on the observation that people usually trust on the opinions of others in making daily decisions [1]. Their main functions are [1]:

- Increase the number of items sold: Suggestions are more likely to tailor users requirements, then an additional amount of products may be sold compared to the number of purchases achieved without any recommendation. This also occurs to non-commercial applications when the user selects the offered content.
- Sell more diverse items: Enable the user to select items that

might be hard to find without the aid of the system. For instance, the system could offer unpopular products that otherwise would probably be overshadowed by the others.

- Increase the user satisfaction: A system able to provide relevant recommendations combined with a properly designed human-computer interaction could increase the likelihood that the suggestions are appreciated.
- Increase user fidelity: A system that when visited recognizes old customers and treats them as valued visitors should keep their interest on the service provided.
- Better understanding of what the user wants: Service providers can adapt the stock, production and ads according to the knowledge obtained by the understanding of users needs.

Collaborative filtering

The main idea of collaborative filtering (CF) recommendation approaches is to utilize the opinions of the community of users (peers) to find similarities of tastes and predict a score that a given user would give to non-purchaseditems [3].

This technique uses as input a matrix of ratings (implicit or explicit) and produces a numerical prediction indicating to what degree the current user would like or dislike a certain item or a list of them. This list should not contain items that the current user has already bought [3].

Collaborative filtering has two types: user-based or user-to-user, and item-based or item-to-item collaborative filtering. The next section describes the fundamentals of each approach. In user-based approach the idea is to offer items to the target user that others with similar tastes also like. Then, for every product p that the active user has not rated yet, a prediction is computed based on the ratings for p made by peers. The underlying assumptions of this method are "if users had similar tastes in the past they will have similar tastes in the future" and "preferences of users remain stable and consistent over time" [3].

Formally, let U={ $u_1, u_2, ..., u_n$ } denote the set of users, I={ $i_1, i_2, ..., i_m$ } the items and $r_{i,j}$ the rating of the user i to the item j in the matrix of ratings or utility matrix with dimensions n x m. The idea is to compute the similarity or weights sim (a, b) between the target user u and others using a metric such as the Pearson correlation. With these values the rating can be predicted using the following aggregation formula (1)

$$pred(u,i) = r_{u} + \frac{\sum_{v \in N} sim(u,v) \times r_{vi} - r_{b}}{\sum_{v \in N} sim(u,v)}$$
(1)

Collaborative filtering can also be item-based. In this case, the system offers to the target user items that have been rated similarly to the products she likes. To predict a missing rating, the similarities between the item and others are computed using, for instance, the cosine similarity. Then the prediction can be obtained using the formula (2)

$$pred(u,i) = \frac{\sum_{j \in J} sim(u,j) \times r_{vi}}{\sum_{j \in I} sim(u,j)}$$
(2)

This technique seeks to overcome the problem faced by the user-based approach when the number of users escalates. Moreover, similarities between items can be obtained offline, allowing the computation of ratings even for very large utility matrices [3]. The prediction step can be simplified utilizing just the nearest neighbors of the target user for the user-based approach and the nearest items to the target item for the item-based [3].

Content-based filtering

These techniques make recommendations based on the user choices made in the past and the characteristics of the items, i.e., the content [7]. The system needs two pieces of information in order to suggest products: a description of items features and a profile that summarizes the past interests of the user. Then, the recommendation task consists of determining the items that best match the preferences of the user [3].

In a content-based system each item has a profile, which is a collection of records representing its characteristics. For example, in movie recommendations the features might be the genre, actors, directors, year, etc. Also relevant features could be extracted from text components of items by using TF-IDF [3,6].

To make recommendations, content-based systems typically work by evaluating how similar an unrated item is to those the active user has liked in the past. This evaluation is usually carried out by comparing the user's profile with the profiles of items applying a similarity measure. Notice that depending on the scales of the features, different transformations such as scaling or normalization may be necessary [6].

Although this approach relies on additional information about items and user preferences, it does not require the existence of a rating history. So recommendations can be generated even if there is only a single user [6].

Context aware filtering

Traditional recommender systems only deal with two types of entities, users and items, and do not put them into a context when providing recommendations. In this scenario, context can be defined as the conditions or circumstances that may have an impact over the decisions of users. Some examples are physical context (time, position, activity of the user, weather, light conditions, temperature), social context (e.g., the user is alone or in group, the presence and role of other people around the user, etc.), interaction media context (device characteristics: phone, tablet, laptop, etc.; media content type: text, audio, video, etc.), modal context (user's state of mind: cognitive capabilities, mood, experience, current goals, etc.) [1]. Adomavicius et al. (cited in Felfernig et al. [5] understand the term "contextual information" in a general way such that it encompasses additional data dimensions, extending the traditional two-dimensional userproduct representation of rating data to a n dimensional data cube. The relevance of contextual information is application dependent. For instance, seasons may have an important impact in a travel recommendation application, but have absolutely no impact in movies recommendations. Thus, the selection of the available context should be based on statistical tests to identify which of the chosen contextual attributes are truly significant [1].

Three main paradigms for incorporating contextual information in recommender systems are contextual pre-filtering, post-filtering and modeling [1]. In the first paradigm information about the current context is used for selecting or constructing the relevant set of data records. Then, ratings can be predicted using any traditional recommender system on the selected data [1]. In contextual postfiltering or contextualization of recommendation output, contextual information is initially ignored, and the ratings are predicted using any traditional 2D recommender system on the entire data. Then, the resulting recommendations are adjusted based on the contextual information.

The last paradigm applies the contextual information directly in the modeling technique as part of rating estimation [1].

Dynamic selection of classifiers

Dynamic selection is a strategy used in Multiple Classifier Systems (MCS) where a specific set of classifiers is selected for each test pattern. The latest research results of the classification methods available in the literature lead to the conclusion that creating a monolithic classifier to cover all the variability inherent to most pattern recognition problems is somewhat unfeasible. Thus, many researchers have focused on providing new solutions for each of the three phases of the MCS: generation, selection and integration [9].

In the integration phase a pool of diverse and accurate classifiers are generated. Dietterich (as cited in Britto et al.) [8] explains that for classifiers to be considered accurates they must have an error rate lower than the random guessing on new samples, while to be diverse they have to make different and perhaps complementary errors. A pool can be homogeneous or heterogeneous depending on whether they use the same base classifiers or not. In the first case, diversity is achieved by varying the information used to construct their elements, such as changing the initial parameters, using different subsets of training data or using different feature subspaces. In the case of heterogeneous the aim is to create experts that differ in terms of the properties and concepts on which they are based [8].

Regarding the selection phase, the idea is to divide the feature space into different partitions, and the most capable classifiers for each of them are determined. The selection can be static or dynamic. In the first case, it is performed during the training phase and remains constant for the successive stages. The dynamic approach can be performed during the training or test phase, although it is commonly applied in the testing one, in which a partitioning scheme based on the NN-rule is used to define the neighborhood of the test sample in the feature space and the competence of each classifier is defined on a local region on the entire feature space represented by the training or validation dataset [8]. Finally, the integration phase fuses the selected experts using ensemble techniques to classify a given testing pattern.

DS solutions may produce a high computational cost, which explains their application to be often criticized. The decision of whether or not to use DS is still an open question [8].

To evaluate a DS approach a minimum requirement is to outperform the single best (SB) classifier, a combination of all classifiers in the pool (CC) and any of the static selection sets in the pool. Also, the concept of oracle is usually present in the evaluation of the proposed methods. The oracle performance is estimated by considering as correctly classified the test samples that can be labeled successfully by at least one classifier in the pool [8].

Tourism and recommender systems

The Web is one of the primary information sources for people when searching or booking suitable travel destinations [3]. Existing recommender systems in e-tourism typically emulate services offered by tourist agents where users seek for advice to select tourist destinations under certain time and budget constraints [8]. Recommender applications can be valuable tools supporting, for example, information search, decision making, and package assembly. The tourism field differs from other application cases as individual travel planning

J Tourism Hospit, an open access journal ISSN: 2167-0269

activities are typically much less frequent. Then, common techniques such as CF may not be enough to provide precise recommendations [3].

Currently, most reported systems use CB, collaborative and demographic-based techniques. These techniques suffer from several problems when applied individually. Hence, a good practice is combining them to overcome the mentioned drawbacks. The most recent approaches follow this trend and propose hybrid recommendation methods, including also contextual information [9]. The particular characteristics of the tourism field offer the possibility to define new mechanisms to learn the user's preferences. In particular, the contextual information is a key in the success of any recommender of tourist activities [9]. It has also been argued that a smart recommender should provide a diversified list of recommendations, e.g., even if the system knows that the user is interested in going to the beach, it might not be convenient to show a list with just beaches and not to suggest other kinds of related activities. The lack of diversity is one of the main issues that CB systems face [9].

The widespread use of mobile devices has increased the application of location aware recommenders. Not only the current location of the user is relevant, but also the current weather and environmental conditions (e.g. temperature, humidity, rainfall degree, wind, season, moment of the day, etc.) to decide if it is more appropriate to recommend indoor or outdoor activities, traffic, facilities (gas station, etc.), tourist attractions (fishing zone, recreation place, seaside, etc.), time needed by the user to reach the place, the opening and closing times, location of the place, social factors (number of users close to the place and number of positive and negative feedbacks), etc.

Issues

Recommendation techniques are the prone to undergo some issues which undermine the suitability of suggestions list:

Sparsity: Many commercial applications handle several users and items. Then, the utility matrix can be extremely sparse and predictions made by CF approaches may be affected. This happens when the number of ratings is too small and users with similar tastes cannot be identified as such because they have not purchased enough items in common [1]. To overcome this problem some systems apply hybrid approaches or dimensionality reduction techniques such as Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) [1]. The first solution is based on the conjecture that the utility matrix is actually the product of two long, thin matrices [6].

Cold start: Also called the new user or new item problem, it happens when a new user or item have just entered the system. The lack of ratings history makes it difficult to find similar peers. This problem usually affects CF techniques. A new item cannot be recommended until some users rate it and it is likely that users do not acquire it due to its lack of purchase history. On the other hand, a poor knowledge of a user's preferences makes it unfeasible the provide accurate recommendations [1].

Scalability: This problem happens when the number of users and items increases substantially, making application of traditional approaches unfeasible due to the high computational resources required. The scalability issue is aggravated when the system needs to perform recommendations immediately as online requirements are coming [1]. The main solutions to deal with scalability when using CF comprises applying SVD to the utility matrix and using item-based, Bayesian or clustering CF [1]. **Gray sheep:** Gray sheep refers to users whose opinions do not consistently agree or disagree with any group of people [1]. White sheeps are those individuals with high correlation values with many other users, while gray sheep have low correlation coefficients with almost all users. In addition to the problem of suggesting items to these users, they can negatively affect the recommendations made to the rest of the community [5]. This problem mainly affects CF systems. For this reason, hybrid techniques that mix CB and CF could address the issue [1]. Ghazanfar and Prügel-Bennett considered the problem as an outlier detection problem, where the similarity between a user and the closest centroid is used to isolate the gray-sheep users [5]. They managed to decrease the error of gray sheep users recommendations.

Shilling attacks: In systems where anyone can provide recommendations, people may give several positive ratings for their own materials and negative ones for their competitors. It is important to introduce precautions that avoid this phenomenon. It has been shown that item-based CF algorithm is much less affected by these attacks than the user-based [5].

Privacy: Users may not want others to know their habits or tastes. Rating databases of CF recommender systems contain detailed and possibly sensitive information about their likes. Sparse factor analysis replaces missing elements in the utility matrix with default voting values (the average of some non-missing elements, either the average by columns, or by rows, or by all). This approach supports computation on encrypted user data (looking at the ratings of a specific user it is not obvious which were given by the actual user). Thus, it provides privacy protection [1].

Over-specialization: Learning-based methods tend to propose items that are similar to the ones the current user has already positively rated. They can produce the effect of giving obvious or repetitive recommendations, for instance, suggesting products too similar to those the user already knows. Then, filtering out not only items that are too far away from the user's requirements but also those that are too close may be a solution Ruotsalo et al. propose a clustering method to group items sharing similar features to then select a representative number of objects from each cluster to increase the diversity. Metrics to measure novelty and redundancy have been proposed. An important concept related to the overspecialization problem is serendipity. Increasing the serendipity means including more unexpected items in the recommendations. It extends the concept of novelty by helping a user find an interesting item that she might not have otherwise discovered [10].

An Hybrid Recommender System

Tourism platforms handle different information sources: demographic information of users, characteristics of tours and also contextual data could be available. For this reason, the developed system is a hybrid between techniques such as CF, CB, demographic and context based filtering. It was decided not to apply pure CF techniques since they depend on finding users with products purchased in common with the target user. This could not be possible in tourism, where user-item interactions are not as frequent as in other situations such as movies stores. On the other hand, CB does not suffer from this issue, since it only needs the characteristics of items instead of the history of purchases to work.

Users and items representation

CB profiles were used to store the characteristics of users and products. Usually in a pure CB system, user profiles contain the features

Page 4 of 9

Citation: Arleo S, Silva B (2017) Dynamic Selection of Recommender Systems: An Application to Tourism. J Tourism Hospit 6: 325. doi: 10.4172/2167-0269.1000325

of the purchased products, i.e., they store the preferences of users in terms of items attributes. On the other hand, an item profile stores the characteristics of that specific product. In the proposed solution users and products have profiles, but besides storing content features, they also contain demographic and contextual characteristics. The subsets of profiles with different information sources will be referred as sub profiles.

Items acquire users demographic characteristics when purchased, while the context at the moment of the purchase is added to both parties involved. Since product features and user's personal attributes do not usually change, demographic sub profiles of users and content sub profiles of items remain constant.

Features have an associated weight that, in the case of user profiles, represents the number of items purchased with that characteristic. In the case of item profiles, the weight of a feature is the number of users with that specific attribute who have bought the product. Contextual features in both user and item profiles also have an associated weight that stores the number of times a specific contextual condition occurred when the user purchased the product (in the case of user profiles) or when the item was bought (for item profiles). Unlike the strategy previously shown to compute the values of features, where they are set to the ratio of the number of items purchased with the attribute, an average of the ratings given by the user to the products with that characteristic was used instead.

After profiles are built, it is possible to find the similarity between users, products or both. Figure 1 shows an example of the profile construction process of a new user after purchasing three products. The demographic subprofile remains constant, while content and context features are taken from the purchased products, reflecting progressively the user's tastes. The resulting profile seems to indicate that the user has a tendency to select tours in Portugal that she prefers art, high-priced tours and traveling alone.

On the other hand, Figure 2 shows an item profile construction after being purchased by three new users. The content subprofile remains unchanged, since characteristics of items do not usually vary. Demographic features are obtained from users who booked the tour and contextual ones from the conditions at the moment of the transaction. At the end, the tour seems to be suitable for women preferably from Spain, who like art, high-cost products, Italy and traveling alone. Notice that from the above examples, the user in Figure 1 has a high similarity with the item shown in Figure 2. Item profiles start only with content features. After the first purchase they acquire the first demographic and contextual characteristics. When comparing profiles using a weighted similarity measure, the weight of each feature pair is obtained by finding the mean between the weights of the feature in both profiles. If this value is high in both profiles, the impact of the feature will be greater in the resulting similarity. Characteristics of items, users and context can have qualitative and quantitative values. Features with a quantitative type were discretized. Figures 1 and 2 show the features "High cost" and "Low cost" obtained from discretizing the price of the tours.

Prediction

The main techniques used to predict a user's rating and subsequently generate recommendations are CB and item-based CF. In CB, the similarity between a user and a tour is used to compute the degree of affinity with a specific product. Then products with a greater similarity are recommended first. Item-based CF calculates the similarity between the target item and the products purchased by the user previously. Then their ratings are added in a weighted aggregation approach using similarities as weights. If the user did not explicitly rate any item, the predicted rating computed by the recommender is used instead.

The difference between the applied strategy and the common itembased CF is that the former uses CB profiles instead of the traditional ones containing the ratings of the items purchased. In case the user does not have purchased items, CB is used since it is normally possible to apply it because even new users have demographic information from which similarities can be computed. Then, the cold start problem is being addressed by this approach.

Dynamic selection of recommender system

The notion of dynamic selection has been applied to obtain the criterion of choice of the recommendation strategy. In dynamic selection of classifiers used in MCS a set of experts is chosen to predict the label of each test sample. Normally the selection criterion is to choose the most locally accurate classifiers by looking at the nearest neighbors of the test sample. In this case, recommenders correspond to the classifiers and the target items to the test samples of the MCS problem. The above strategy was adapted to a ranking problem rather than a classification one, where the predicted continuous values are used later to give an order to the products that will be recommended.







bestrec(t, u) = argmax_{x \in R}
$$\left(\sum_{i \in IAbestrec(i)=r} sim(t, i) \right)$$
 (3)

The selection criterion applied consists of choosing the recommender used to predict the ratings of the items purchased by the user which are more similar to the target product. This performance is measured by adding up the weights of each recommendation method used to predict the ratings of the purchased products. Notice that these methods were chosen using the same reasoning, which makes the dynamic selection an inherently recursive strategy. Eqn. (3) is used to determine the best performing recommender, where:

- t: the target item
- u: the target user
- sim(t, i): the similarity between the target item t and the previously purchased i
- R: set of recommenders selected as the best performing methods in predicting the rating of any purchased item recommender selected as best recommenders for some item bought by u
- I: set of items purchased by user u
- bestrec(i): Recommendation method with the lowest error when predicting the rating of item i.

In the generation phase, a pool of diverse and accurate classifiers (in this case, recommenders) have to be created prior to the dynamic selection. To do so, four similarity measures were applied (weighted and unweighted Euclidean and Manhattan distance-based similarities) and different subsets of features were used (i.e., content, demographic and contextual subprofiles). Distance measures were transformed into similarities. Eqn. (4) presents both Euclidean and Manhattan weighted distance-based similarities.

$$sim(U,V) = \frac{\sqrt[p]{\sum_{i=1}^{n} w_i^p \times |u_i - v_i| p}}{\operatorname{amplitude} \times \sqrt[p]{\sum_{i=1}^{n} w_i^p}}$$
(4)
Where:

- U,V: profiles containing features present in both
- p: the order of the metric

- w: average of the weights of feature i in U and V
- amplitude: distance between the maximum and minimum possible ratings

A value of p=1 corresponds to the Manhattan metric, while p=2 to the Euclidean. Using w=1 results in the unweighted similarities. The reasoning behind the application of distance-based similarity measures is that they are not affected by the values of the features themselves, but only by the difference between them. Only the features present in both profiles are taken into account in the similarity computation. Every time a user buys a new product, more information about her tastes is available, making her profile more complete. Then, the recommendation methods that best predict the rating given to each product purchased are updated using the new information.

Validation

This chapter presents the experiments performed to evaluate the system. The purposes of these tests are to evaluate the dynamic selection approach and the applicability of the recommender to tourism as long as to others cenarios.

Recommender systems are usually evaluated offline with reviews made in the past using evaluation measures that may vary according to the type of the recommendation method. Gunawardana and Shani [10] defines guidelines for the construction of offline experiments in three recommendation modalities: "Recommending good items", "Optimizing utility" and "Predicting ratings". Only the last case was considered since the others are out of the scope of this study.

To evaluate algorithms offline it is necessary to simulate an online situation using previously recorded historical user data, hide some interactions (ratings) and then try to estimate how the user will rate an item. If a pre-filtering of the data is performed, users and items involved in low a low number of transactions should not be excluded. If necessary, randomly sampling may be a preferable method. After the pre-filtering steps, the set of users is partitioned in a test and training sets. Purchases of each test user are also randomly sampled. A common protocol is to select a fixed number of hidden items per user. However, predicting more than one rating for the same test user may affect the independence of results.

To carry out the first evaluation the system was compared against

the single best (SB) recommender, as suggested in Britto et al. [5]. The best performing variant was used in the other experiments. For the tourism experiments a dataset of reviews of hotels was used. A description of these datasets is presented in the beginning of the chapter.

To generate the training and test sets a random selection of users was performed. The size of the latter was set to one fourth of the total number of users. Also, for test users only one test purchases was randomly chosen to ensure the independence of results. For each data set three scenarios were considered.

The first one corresponds to a situation where users already provided a reasonable number of ratings. The second scenario was created to test the performance when dealing with the user cold start problem. In this case one test purchase for each test user was selected and the others were ignored. This to simulate the lack of purchase history of new users. The third scenario represents the item cold start problem. It was emulated by randomly choosing a set of items, selecting only one purchase per item and discarding other transactions that involve them. Thus, the first purchase of the product is simulated.

Results are presented in terms of Root Mean Squared (RMSE), Mean Absolute Error (MAE) and Normalized Mean Absolute Error (NMAE). To guarantee their significance, each experiment was performed a hundred times and results were averaged. Moreover, the execution times of the final experiments are shown alongside the errors.

Dataset

In order to test the applicability of the system in the tourism field, it was used a dataset with online reviews from Trip advisor, a travel website that supports travel planning and allows users to rate products, provide reviews and specify a trip type from among family, friend, couples, solo and business trip. The dataset contains 4669 ratings, 1202 users and 1890 hotels, which indicates a density of 3.88 bookings per user (0.205% of the total number of hotels) and a highly sparse utility matrix with 99.795% of empty entries. No missing or erroneous values were found.

This dataset features cover demographic information (user

information such as user naturality and time zone), content (hotel location, given by city, state and time zone) and contextual information (trip type).

Dynamic selection tests

To test the performance of the DS, this approach was compared against the single best (SB) static approach, as suggested by Britto et al. [8]. Single recommenders were obtained by selecting a main recommender among CB and CF, a similarity measure and a combination of content, demography and context subprofiles. After finding the SB recommender, different variants of DS based systems were generated by applying changes to the components that make up the system described in Chapter 3. Then, the systems were compared using the MAE as the main evaluation criteria.

Each recommender was tested a hundred times, averaging the results. One fourth of the total amount of users (300) was selected as test users, and for each one a test purchase was set as the target of predictions, resulting on 300 test purchases. The rest (4369 purchases) were used to feed the recommender. All tests were made in a common scenario where cold start does not prevail.

Table 1 contains the information about the evaluated single recommenders variants. A different CB and CF system can be created for each of these variants.

The first step to discover the single best recommender was to compute the performance of each system using different similarity measures. For CB systems, the best performance in terms of MAE was achieved with Weighted Manhattan similarity measure. For CF systems, it was the Euclidean similarity that presented the lowest error.

After determining the best similarity measure, the performances of the main subprofiles combinations for CB and CF systems were computed using Weighted Manhattan and Euclidean similarity.

Table 2 ranks the CB and CF single variants by MAE. The SB recommender was the S1 variant of CF which makes use of the Euclidean based similarity measure and all the available features, i.e., the content, demographic and contextual subprofiles.

ID	Euclidean	W. Euclidean	Manhattan	W. Manhattan	Cosine	All subp.	С	CD	CC
S1	1					1			
S2	1						1		
S3	1							1	
S4	1								1
S5		1				1			
S6		1					1		
S7		1						1	
S8		1							1
S9			✓			1			
S10			✓				1		
S11			1					1	
S12			✓						1
S13				1		1			
S14				1			1		
S15				1				1	
S16				1					1
S17					1	1			
S18					1		1		
S19					1			1	
S20					1				1

Page 7 of 9

Citation: Arleo S, Silva B (2017) Dynamic Selection of Recommender Systems: An Application to Tourism. J Tourism Hospit 6: 325. doi: 10.4172/2167-0269.1000325

Page 8 of 9

Main recommender	ID	RMSE	MAE	NMAE	Time
CF	S1	1.248494305	0.950216847	0.237554211	0.051570053
CF	S13	1.252705987	0.955875023	0.238968755	0.095677245
CF	S9	1.254309118	0.957060759	0.239265189	0.088326646
CF	S5	1.256007563	0.959420994	0.239855248	0.092532572
CF	S3	1.263563059	0.960746507	0.240186626	0.082785764
CF	S17	1.262859798	0.961795306	0.240448826	0.096327915
CF	S4	1.277045078	0.968635078	0.242158769	0.082765223
CF	S2	1.319289591	1.000649915	0.250162479	0.075401576
CB	S13	1.402022446	1.059222892	0.264805723	0.065555030
CB	S9	1.407412013	1.062606763	0.265651690	0.063248733
CB	S1	1.414169800	1.073094004	0.268273501	0.061777223
СВ	S16	1.523298897	1.156498513	0.289124628	0.065596200
СВ	S5	1.515938742	1.167157840	0.291789460	0.064529736
CB	S15	1.542301991	1.171286390	0.292821597	0.065273139
СВ	S17	1.730491028	1.355251657	0.338812914	0.065308992
СВ	S14	1.797736900	1.387287648	0.346821912	0.063648922

Table 2: Ranking of single recommenders by MAE.

ID	СВ	CF	Euclidean	W. Euclidean	Manhattan	W. Manhattan	All subp.	С	CD	CC
D1	1	1	1	1	1	1	1	1	1	1
D2	1	1	1				1			
D3	1		1	1	1	1	1	1	1	1
D4	1		1	1	1	1	1			
D5	1					1	1	1	1	1
D6		1	1	1	1	1	1	1	1	1
D7		1	1	1	1	1	1			
D8		1	1				1	1	1	1

Table 3: DS based recommenders.

After having found the SB it was possible to compare it with the DS based recommenders. Different variants of DS were created by selecting a main recommender (or both), a similarity measure (or a combination of all them) and a subprofile (or the whole set). Table 3 shows these recommenders.

In order to evaluate the impact of the application of DS to each recommendation technique, performance of dynamic CB and CF recommenders was compared against the best SB recommender. Table 4 ranks DS variants and best SB recommender by MAE.

According to results shown in that table, is possible to note that the best DS variant was D7. However, the improvement over the SB was not significant, since the latter achieved 0.950216847 of MAE (a difference of 0.00081153). The presence of CB in the DS variants seems to have decreased the performance of recommenders.

As shown in Table 4, the top 4 dynamic systems do not make use of the mentioned technique. This might be a result of the lack of accuracy of CB itself.

According to these results, the SB S1 recommender was chosen as the best performing recommender and was tested under three different scenarios: a normal situation where cold start is not prevalent, a situation where all test users are new and a hypothetical system state where all target items have not been purchased before. Results are displayed in Table 5.

Conclusion

The application of a recommender system in a tourism scenario must take into account that users do not tend to book a high amount of products. This discards pure CF strategies, as they would have to deal with a high sparsity. Instead, a CB strategy was proposed taking

ID	RMSE	MAE	NMAE	Time
D7	1.242218789	0.949405317	0.237351329	0.085027674
SB	1.248494305	0.950216847	0.237554211	0.051570053
D8	1.255639909	0.958357299	0.239589324	0.082163067
D6	1.257268997	0.960446473	0.240111618	0.081858251
D2	1.360617276	1.028100287	0.257025071	0.074728153
D4	1.423978831	1.078379186	0.269594796	0.063522528
D5	1.423978831	1.078379186	0.269594796	0.063522528
D3	1.643002542	1.249151705	0.312287926	0.063781625
D1	1.649245019	1.249372673	0.312343168	0.082290168

Table 4: Ranking of DS based recommenders and SB ((S1)) by	y MAE
--	------	------	-------

Scenario	RMSE	MAE	NMAE	Time	
Normal	1.238364944	0.947538162	0.236884540	0.055648501	
User cold start	1.197942982	0.935927114	0.233981778	0.014631695	
Item cold start	1.380836303	1.045263567	0.261315891	0.048525700	

Table 5: Performances of SB selection in a tourism different scenarios.

into account other information sources that may be available such as the demographic information about users and the context. This was made by creating profiles that represent the tastes and demographic attributes of users, and the features of products and the characteristics of the customers that bought them. Contextual features were also stored in this structure, representing the preferences of users for certain conditions at the moment of visiting a tour and the regular situation parameters that occur when the touristic attraction is booked.

The inclusion of multiple information sources in a CB basis resulted in a substantial improvement of the system's accuracy. A reduction of 5, 04% in the Mean Absolute Error (MAE) was achieved by the single best CF S1 (Table 1) compared to the S2, a system that only uses content features. Similarly, the presence of demographic and contextual features in profiles generated a decrement of 23,64% of the MAE in the CB variant S13 (single best CB) with respect to the S14.

The system proposed in Chapter 3 is a switching hybrid that uses a dynamic selection strategy to choose the most suiting recommender. This selection is performed over a pool of recommenders composed by CB and CF recommendation methods, different similarity measures and subprofiles. The idea behind the DS was to find, for each user, the system with the best performance in predicting the ratings of the acquired items.

Tests were performed using two datasets with online reviews of hotels and books. Results obtained in Chapter 4 showed that the application of DS did not provide any significant improvement.

The most accurate dynamic variant obtained an performance equivalent to the SB recommender. Considering the complexity of this approach it is reasonable to discard its application and to apply the SB instead.

After comparing all the systems that can be derived from the recommender presented in Chapter 3, the single CF based variant which uses the Euclidean similarity and the whole set of features (content, demographic and contextual subprofiles) outperformed the others methods. This recommender belongs to the Monolithic design class of hybrids. More specifically, to the Feature combination hybrid type.

The system is capable of working under the user and item cold start problems. To deal with the former CB was used, which only needs the profiles of the target user and items.

In the Soboto application case, demographic, contextual and content information is available from which predictions can be made.

An interesting result was the poor performance achieved by the variants that made use of the Cosine similarity compared to the Euclidean and Manhattan distance-based similarities. The latter metrics are not affected by the values of the features themselves, but by the difference between them which may be the cause of the improvement.

Recommendation issues addressed

The following are the main issues addressed:

- Cold start: The system deals with the cold start of users since with the majority of booking online platforms, every user has to provide demographic information that can be used to build her profile and then generate recommendations. The cold start of items is attacked since products usually have characteristics associated from which to build their profiles.
- Sparsity: This issue is addressed indirectly since the system does not depend on finding common rating patterns among users in a utility matrix as CF does. The sparsity in the profile matrix

does not represent a problem because usually most items share a set of characteristics such as price and category, easing the comparison between any pair of users or items.

- Scalability: There is no need to maintain all users in memory, but just the online ones. Moreover, a hash table based representation of profiles may be enough to bear an increase in the number of users and items.
- Gray sheep: Due to recommendations which are not collaborative-based (the system does not depend on the opinions of other users to suggest products), but based on the characteristics of the products the user has liked, the system might have a good performance tailoring recommendations to users whose opinions do not consistently agree or disagree with any group of people.

Limitations

The main shortcomings identified were:

- Overspecialization: The system recommend products based on the users purchasing habits, which may tend to specialize suggestions in a small group of items similar to the ones already purchased, which results in allow degree of novelty.
- Privacy: Since users' tastes are present within profiles in the form of items characteristics, additional mechanisms have to be applied to protect the sensitive information.

References

- 1. Ricci F, Rokach L, Shapira B, Kantor PB (2010) Recommender Systems Handbook. Springer-Verlag, New York, NY, USA.
- Goldberg D, Nichols D, Oki BM, Terry D (1992) Using collaborative filtering to weave an information tapestry. Commun. ACM 35: 61-70.
- 3. Jannach D, Zanker M, Felfernig A, Friedrich G (2010) Recommender Systems: An Introduction. Cambridge University Press, NewYork, NY, USA.
- Gavalas D, Konstantopoulos C, Mastakas K, Pantziou G (2014) Review: Mobile recommender systems in tourism. J Netw Comput Appl 39: 319-333.
- Felfernig A, Gordea S, Jannach D, Teppan E, Zanker M (2007) A short survey of recommendation technologies in travel and tourism. ÖGAI Journal 25: 17-22.
- Rajaraman A, Ullman JD (2011) Mining of Massive Datasets. Cambridge University Press, New York, NY, USA.
- Burke R (2007) The adaptive web. Chapter Hybrid Web Recommender Systems. Springer-Verlag, Berlin, Heidelberg.
- Britto AS, Sabourin R, Oliveira LES (2014) Dynamic selection of classifiers-a comprehensive review. Pattern Recogn 47: 3665-3680.
- Borrís J, Moreno A, Valls A (2014) Review: Intelligent tourism recommender systems: A survey. Expert Syst Appl 41: 7370-7389.
- Gunawardana A, Shani G (2009) A survey of accuracy evaluation metrics of recommendation tasks. J Mach Learn Res 10: 2935-2962.