Explanations in Recommender Systems

stu222703
Christian Albrechts Universität
Recommender Systems Seminar
Kiel, Germany
stu222703@mail.uni-kiel.de

ABSTRACT

When there's to choose between a ton of different items (for example movies on Netflix, music on Spotify, products on Amazon or locations for the next vacation), the so called "Recommender Systems" come into play to support the user in order for him or her to make the right or an enjoyable decision. As the amount of choosable items increases, it becomes more and more difficult for the user to make a choice or even search and find the items which he or she would like to use. Recommender Systems are software tools and techniques to determine which items the user would probably like to use in future and to recommend those certain items based on methods I intend to speak about in later sections.

KEYWORDS

recommender systems, explanations, collaborative filtering, case-based

ACM Reference Format:
stu222703. 2018. Explanations in Recommender Systems. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION AND MOTIVATION

Recommender Systems are software tools and techniques, according to Ricci et al. (2022) to determine which items the user would probably like to use in future and to recommend those certain items based on different methods. There are two main possibilities how the user could face new items, the first possibility is just by searching for the item's name or typing in keywords regarding certain items, so basically in this form the user is "walking" straight up to the item. The problem with this possibility is though, in many (or most) cases the user just doesn't know what specific item or even what kind of an item he or she wants to buy or consume. This is where the second possibility is introduced, namely the recommendation of items. In this scenario, the item "walks" to the user or gets a helping hand find the path to the user. On the one hand, with this option the user gets a massive help discovering new items. On the other hand, there's always a certain risk of the user not liking the suggestions and therefore rejecting or just ignoring them. Nevertheless, in this work we will focus on dealing with different techniques of recommending items and explanations on why the user is getting a suggestion for a certain item.

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

Why should we work on selecting the most suitable recommendation for each and every user when there are always options like creating and publishing a **general "Top 10" list of items** (for example most bought products on Amazon for a specific month), a **general most popular list of items** (for example the most watched movies on Netflix) or a **general list of recent uploads** (for example a list of the newest songs on Spotify)?

The issue behind those kind of recommendations is that those are in no manner tailored to an individual user. There's no input from the user's side and therefore it's imaginable hard to lead the user to a decision that will satisfy his or her needs. Nonetheless it must be said that those "Top 10"-lists can portray a justified side-solution as a new, indecisive user would rather follow other people's choices than just start typing in random stuff to find new items. In spite of that fact, the main focus should be spent on getting to know the individual user's preferences.

Once the focus is set on getting to know the user's preferences, explanations can be made on why the user is getting recommended a certain item to for example boost the user's trust into both the recommendation and the company selling or offering the item.

2 RECOMMENDATION TECHNIQUES

There are different methods or techniques to recommend items to a user. Those methods differ regarding the domain, the knowledge and the algorithm behind the recommendation, as described by Ricci et al (2022).

The methods of recommending items we are going to talk about are:

- **Content-based**
- **Collaborative filtering**
- **Case-based**
- **Knowledge-based**
- **Hybrid solutions**

We are going to take a closer look at some different recommendation techniques in particular and we will present an example of how an explanation could look like for each of the recommendation techniques.

2.1 Content-based

The first method we will talk about is the content-based recommendation. Content-based recommendation is based on the availability of (manually created or automatically extracted) item descriptions and a profile that assigns importance to these characteristics (Janach et al. (2010)). The basic idea of content-based recommendation is as follows: Assume there is a user and he rates certain items highly, the recommendation-system will now suggest items similar to the ones the user just rated highly.

A content-based explanation from Amazon.com looked like this: **"Recommended because you said you owned Book A."** (Ricci et al. (2022))

2.2 Collaborative filtering

The collaborative-filtering recommendation technique is based on considering a certain user and finding a set of other users whose ratings are similar to the ratings of the considered user and estimate the considered user's rating based on the ratings of the other users. Those other users are usually called neighbors, so customers with similar taste to the one of the considered user. An explanation based on collaborative-filtering could be like this: **"Customers who bought the guitar also bought this bag to store it in."**

2.3 Case-based

We will discuss the case-based recommendation based on the work of Janach et al. (2010). Case-based recommendation is based on identifying the products which fit best to the query made by the user where a case represents a product. When a user puts on so called constraints on attributes of products, the amount of products which still qualify for a recommendation usually decreases as there are more features to take into consideration.

A simple explanation may look like this: **A pencil is recommended to you because you rated the pen highly.**

2.4 Knowledge-based

In knowledge-based approaches, the recommender system typically makes use of additional, often manually provided, information about both the current user and the available items (Janach et al. (2010)). When there's to buy a TV for example, we don't buy TV's very often so the recommendation system cannot rely on the user to continuously buy TV's over and over again to gain information. This is where knowledge-based recommendation systems come into play by stating out the functions of a certain product.

A knowledge-based explanation might be **"The XYZ TV offers a resolution of 4K."**

2.5 Hybrid recommender systems

In order to deliver a more suitable recommendation, recommendation techniques could well get combined to form a hybrid solution. For example, a considered user got observed and content-based recommendations could be made but instead of delivering the recommendation straight away, the user could get grouped up with neighbors, meaning customers of similar preferences, in order to make an even preciser recommendation or to make more suitable recommendations.

An example of explanation could look like this: **"The stuffed lion is recommended to you because you said you owned a stuffed giraffe. Furthermore, customers who bought the stuffed giraffe also bought a stuffed lion."**

3 PROBLEM FORMULATION

When the user receives a certain recommendation he or she has the right to get to know the reason behind him or her getting that suggestion. **"The movie XYZ is a must-see for you!" "Alright, fine, but why exactly should I watch that movie now?"** Making bold statements like "must-have", "must-see", "must-listen" without backing them up properly will may result in the user questioning the quality and even justification of a recommendation.

While receiving and pursuing a recommendation to try a new ice-cream flavour is not really coupled with a high risk (of losing money and time) for the user, receiving and pursuing a recommendation for the next vacation is indeed linked to a higher risk as it costs more money and valuable time for the user. So the importance of an explanation of a suggestion depends on the domain and the risk which comes along with it.

There are different types of explanations as pointed out by Janach et al. (2010).

- **Functional**: Functional explanations offer clarification about the function of the item you are just getting recommended. **Example:** "The Bugatti Veyron would be well suited for you because the race you want to attend only allows cars with more than 1000HP to take part in."
- **Causal**: Causal explanations offer clarification about why a certain event occurred. **Example:** "The plate broke because you dropped it."
- **Intentional**: Intentional explanations offer clarification about behaviour of human beings. **Example:** "You have to clean the floor because your boss requested it."
- **Scientific**: Scientific explanations offer clarification about relations concepts in science.

To increase the quality of communication and comprehension between user and company or selling agent and buying agent, explanations should be made use of. While the selling agent might be interested in boosting the amount of sells of a certain product by recommending it all the time, the buying agent is just afraid of making the right decision (Janach et al. (2010)) or is afraid of making a decision he or she may will somehow regret and therefore wants to play it safe.

This leaves us with the question **"What are the aims of explanations in detail and how should explanations be made?"**.

4 APPROACHES AND RESPECTIVE RESULTS

In this section we are going to look at the different aims of explanations in recommender systems and will discuss two approaches of explaining recommendations with those two being **Explanations in case-based recommendation** and **Explanations in collaborative filtering**.

4.1 Aims of explanations

The aims of explanations in recommender systems can be summed up as follows according to Janach et al. (2010):

4.1.1 ***Transparency.*** Transparency is given when the user gets information about **why** he or she receives a certain recommendation. The information can also present why the recommended item was chosen over other items, and for example a transparency explanation could state that it used the fact that a user bought some kind of music and therefore he or she is getting recommended movies of the same genre, according to Janach et al. (2010).

4.1.2 ***Validity.*** To create validity of a certain recommendation, the recommendation system can compare the needed and recommended product features in order for the customer to assess whether the recommendation is rather valid or not. The focus can be switched to the constraints the user put on and the explanation can feature some words about how the recommendation system took the user's preferences and no-go's into consideration.

4.1.3 ***Trustworthiness.*** Trustworthiness can be created by delivering a true information about what the recommended product has achieved in the past. For example, an explanation like "Most of the tennis tournaments won, were played with this racket." Trustworthiness covers some aspects of the phenomenon "social proof".

4.1.4 ***Persuasiveness.*** In order to persuade the user to follow the recommendation and buy the recommended product, an rather unethical approach is known composed of just listing the product's positive features and not having a single say about the product's disadvantages or negative aspects.

4.1.5 ***Effectiveness.*** How effective a recommender system is, will eventually be determined by the fact of how much the user can rely on support to make the best possible decision. Explanations boosting the effectiveness will help the user locate his or her preferences in order to make that mentioned best possible decision.

4.1.6 ***Efficiency.*** How efficient a recommender system is, will be determined by how well the user effort is getting minimized, so explanations in this context usually are used to decrease the amount of time the user has to spend while making the decision.

4.1.7 ***Satisfaction.*** Satisfaction is given when a user enjoys getting a helping hand from the recommendation system and increase the chance of the user returning to buy again one day.

4.1.8 ***Relevance.*** When a conversational recommendation system is used, which we by the way are not discussing in this work, some extra information coming from the user may be needed to make a more precise recommendation, so an explanation containing a simple "In order for us to make the best possible recommendation we need Information XYZ." could work well.

4.1.9 ***Comprehensibility.*** If we talk about comprehensibility we want the recommendation system to relate the user's known concepts to the concepts to the employed by the recommending system (Janach et al. (2010)).

4.1.10 ***Education.*** Explanations can be made to deliver information to the user which help him or her understand the product domain better. This way it helps the user gaining an understanding and what to look for when buying products of the respective domain.

4.2 Explanations in case-based recommendation

We will discuss the explanations in case-based recommendation based on the work of Janach et al.(2010). **Short recap:** Case-based recommendation is based on identifying the products which fit best to the query made by the user where a case represents a product. When a user puts on so called constraints on attributes of products, the amount of products which still qualify for a recommendation usually decreases as there are more features to look at.

Assume some user wants to buy a car but not just any car, the car needs to be of blue color, must have more than 300HP and is requested to be a 4-seater. Thus, cars with a color other than blue, regardless of them being a 4-seater and having more than 300HP will automatically disqualify to match every single one of the user's requests. A blue car with four seats and 280HP slightly fails to satisfy all of the user's preferences but comes very close, so if assumed that's the best case the user can get offered from a certain company, he or she can make up his or her mind in order to decide to either buy the car or not. Let's call this offered car the **case with the biggest similarity to the users query**.

So called "weights" play a big role as well, meaning the user should be clear about to what extent all the attributes really matter and above all, how much they really matter for him or her to make a choice.

Let Q be the user's query and A the attributes of a case C. Then A_Q represents a subset of the attributes A. The similarity of a particular case C to the user's query Q is defined as follows with w_a being the mentioned "weight" of the respective attribute of cases. "Weight" stands for the importance the user awards the attribute. To come back to our example: If the amount of HP is weighted lower by the user than the color of the car, the color of the car is more of importance than the HP for the user.

$$sim(C,Q) = \sum_{a \in A_Q} w_a sim_a(C,Q)$$

With the formula above, the user will get recommended a set of cases, so products, which are most similar to the query the user just expressed. When the user gets a set of recommendation put in front of him or her, he or she may wonder why exactly those items are getting suggested. This is a so called **"why-question"** of recommendation, meaning the user asks **why** he or she is getting that specific recommendation or set of recommendations.

In order to deliver an answer to a **"why-question"** in case-based recommendations, it is a common approach to just compare the recommended case to the preferences named by the user. This way it can be shown which constraints of the user are met and which are not (Janach et al. (2010)). The user can then have some thoughts again on how important he or she feels the respective attributes are.

id	price	mpix	opt-zoom	LCD-size	movies	sound	waterproof
p1	148	8.0	4x	2.5	no	no	yes
p2	182	8.0	5x	2.7	yes	yes	no
p3	189	8.0	10x	2.5	yes	yes	no
p4	196	10.0	12x	2.7	yes	no	yes
p5	151	7.1	3x	3.0	yes	yes	no
p6	199	9.0	3x	3.0	yes	yes	no
p7	259	10.0	3x	3.0	yes	yes	no
p8	278	9.1	10x	3.0	yes	yes	yes

Table 1: Example product assortment: digital cameras. Janach et al.(2010)

Consider **Table 1** by **Janach et al.(2010)**. The enclosed upper row is A, the set of attributes. $A = (price, mpix, opt - zoom, LCD - size, movies, sound, waterproof)$. With the user then setting constraints, the included attributes form a subset A_Q. Imagine the user wants a camera with at least 10 mpix which is waterproof, then the subset would look like this: $A_Q = (mpix, waterproof)$, representing the attributes of the user's query. As we have a closer look at the table, we recognize that the only camera which is waterproof and at the same time has 10 mpix is case p4, so the logical recommendation for the user would be the camera with id p4.

If the user now wonders why he gets recommended camera p4 and demands an explanation, the explanation can be as simple as **"As you were looking for a waterproof camera with at least 10 mpix, out of the overall eight cameras, the camera p4 is the only option which fulfills your requests."**. This is a short and simple method of telling the user that his constraints are satisfied and the way to answer the **"why-question"** of the user.

The user's weights can also play a big role in determination of the best suited case for the user. Let's say the user is not willing to pay more than 200€ but wants a camera to have at least 10x opt-zoom. The user's preferences say he or she weighs opt-zoom much more than price. So the reasonable recommendation would be camera p4, even though it costs 7€ more than camera p3 which also has got 10x opt-zoom. Reason behind this choice of recommendation is the user's weighing. Note that the user weighed opt-zoom more than price, so it's fairly appropriate to spend 7€ more and receive a camera with 12x opt-zoom instead of one with 10x opt-zoom.

The explanation made could look like this: **"As you were looking for a camera with a price of less than 200€ and with at least 10x opt-zoom, consider camera p3 (price = 189€, opt-zoom = 10x) and camera p4 (price = 196€, opt-zoom = 12x)."**.

We should also consider the case that all cameras may fail to satisfy the user's needs. Assume the user wants to buy a camera with 10 mpix but is not willing to pay more than 190€. If we now take a look again on Table1, there's not a camera which fulfills these preferences. So the constraints the user just put in leave us with no solution and therefore we are forced to find the most similar solution to the user's constraints. In this case, camera p4 is the most similar solution to the user's constraints because the camera has got 10 mpix and costs 196€, so 6€ more than what the user was ready to pay. An explanation for the recommendation of camera p4 may look like this: **"There's no camera offering 10 mpix with a price less than 190€, however the camera p4 costs 196€ but has got your desired 10 mpix."**

According to a statistical evaluation, arguments which are personalized in terms of the user and his or her preferences, constraints and needs are more effective than general arguments that have no reference to the respective user or no arguments at all. Delivering personalized arguments leads to a higher probability of the user accepting the recommendation as a good choice (Janach et al. (2010)).

4.3 Explanations in collaborative filtering

We will discuss the approach of explanations in collaborative filtering based on the work of Jannach et al. (2010).

Let's bring the idea of collaborative filtering back to our mind. Jannach et al.(2010) indicate that there are three common steps of collaborative filtering with those three being

- **Customers rate products**
- **Finding neighbors (customers with similar ratings)**
- **Non-rated items receive a rating composed of the ratings of the user's neighbors**

The idea behind collaborative filtering is to manage an imitation of the "word-of-mouth" propaganda operated by human beings. The "word-of-mouth" term means talking to another person about a certain item in order to convince the other person to try or buy that item. To deliver explanations in collaborative filtering the approach composes of creating transparency by setting out how the "word-of-mouth" marketing actually works.

4.3.1 ***Customers rate products.*** The importance of explanations in collaborative filtering recommender systems regarding customers rating items is immeasurable, underlined by following scenario: Assume a user once buys a book about true-crime. There are some recommender systems which just observe this and draw a conclusion to it according to Jannach et al. (2010). Consequences would be recommendations of several movies about true-crime. The problem is though, that the purchase of the book may just be a present for one of the user's friends. But how should the collaborative filtering recommender system figure out that it was a purchase as a present? It may get confused and start recommending movies about true-crime which the user has absolutely no personal interest in and it gets even more complicated when the user not only orders one but several true-crime books for his or her other friends. Collaborative filtering could get tricked this way as the user's score for true-crime products is fairly high and he or she will end up getting paired up with absolute true-crime-fanatics as neighbors in the collaborative filtering system. Therefore it is important to **deliver explanations and make the user understand where recommendations come from.**

Showing the user where recommendations come from boosts the transparency.

4.3.2 ***Finding neighbors (customers with similar ratings).*** To predict ratings of a user, the collaborative filtering system now looks out for customers with similar ratings to those of the considered user.

Imagine there is a customer who, this time around, likes true-crime movies and there is a group of other customers that likes romantic movies. The considered customer now gets grouped up with those other customers as his or her neighbors only due to the fact that the ratings of the customer and the other customers about comedy movies and historic movies are nearly identical. Is that helping the user to find more of his or her beloved true-crime movies? Not really. This is why it's of relevance to **create transparency by keeping the user updated what kind of neighbors he or she is grouped up with as this is a solution where the user can have a say when it comes to evaluating the collaborative filtering recommendation system him- or herself.**

4.3.3 ***Calculate ratings by combining ratings of neighbors.*** The third and final step of collaborative filtering consists of combining the ratings of the neighbors of the considered user. Jannach et al. (2010) points out that this step is normally implemented by using the weighted average and that the number and variance of the individual ratings play a role when it comes to checking the reliability of such a calculation. Furthermore in the named source it is said that a large number of ratings of neighbors with a low variance is generally more reliable than a few ratings with a high variance. Therefore the authors of the named source suggested to **provide the user with this information in order for him or her to have chance to help and give feedback regarding the quality of the recommendation and to boost his or her confidence in the collaborative filtering recommending system.** The aim is to create a win-win situation for both selling agent and buying agent.

Herlocker et al. (2000) have lead a study about different type of explanations in the domain of the system "MovieLens". Basically, there were 21 different methods of explaining a recommendation of a movie and the study's participants were to admit how likely (on a scale of 1-7) they will watch the recommended movie after each of the overall 21 types of explanation.

Jannach et al. (2010) also referred to the study of Herlocker et al. (2000).

The results showed that explanations including the neighbors' ratings were very successful as the argument of past experience, meaning an exemplary explanation like "MovieLens has provided accurate predictions for you 80 percent of the time in the past" was very successful as well. The explanation of similarity to other movies rated and the argument of the favorite actor or actress performed very well, too.

Just leaving the user with no extra explanation didn't perform well or bad, it was about average.

Poorly designed explanation interfaces made the users not want to follow the recommendation, so in order to build trust it should be spent some time and effort into portraying or displaying the explanation to the user in an appropriate way. Furthermore, too much information had negative effects as well, so keep it short and simple. Contrary to the aspect of "social proof", surprisingly explanations which included the opinion of websites which are known for rating movies, were not accepted more than explanations without the opinion of those websites.

Those were the results of the study of Herlocker et al. (2000), pointed out by Jannach et al. (2010) and summed up by us.

5 CONCLUSION

Until recently, explanation in recommender systems appears to have been a relatively neglected issue (Mcsherry (2005)) However, recent research has highlighted the importance of making the recommendation process more transparent to users... (Mcsherry (2005)).

At the beginning of this work we stated that we think the user has the right to get to know why he or she receives a recommendation and we stand by it as the discussed aims and overall benefits speak volumes. The trust in the recommendation system the user gains and the transparency the user enjoys is something we don't want to even imagine to miss because it just makes the joint understanding of selling agent and buying agent way easier than without having explanations implemented in recommender systems.

Considering case-based recommendation, we presented an approach and stated that to a statistical evaluation, arguments which are personalized in terms of the user and his or her preferences, constraints and needs are more effective than general arguments that have no reference to the respective user or no arguments at all. So personalized arguments are key to backing the recommendation up with validity. The user needs to notice that the recommendation wasn't created by rolling the dice but by listening to the user's needs and preferences.

In terms of explanations regarding collaborative filtering, the study of Herlocker et al. (2000) showed results which admitted that explanations need to be made with designing effort and that the ratings of neighbors are important to the user. Furthermore, it got clear that the fact of past experience was of importance for the user as well, meaning it's important to let the user to know that the recommending system was successful in recommending items to him or her in past occasions.

All in all, explanations should be created with their aims in the back of the design-oriented developer's mind and with a view on maximizing the benefits of both selling agent and buying agent.

6 REFERENCES

[1] Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (2022). Recommender Systems Handbook (3rd ed.). Springer Verlag, New York, NY, USA.

[2] Jannach, D., Zanker, M., Felfernig, A., Friedrich, G. (2010). Recommender Systems: An Introduction (1st ed.). Cambridge University Press, New York, NY, USA.

[3] Herlocker, J.L., Konstan, J.A., Riedl, J. (2000). Explaining Collaborative Filtering Recommendations https://doi.org/10.1/358916.358995

[4] Mcscherry, D. (2005). Artificial Intelligence Review (Explanation in Recommender Systems) 24: 179-197 DOI 10.1007/s10462-005-4612-x

```python
import pandas as pd

# name: stu222703 (Michael Eiramia)
# number:

# Datei einlesen und printen
data = pd.read_csv(r'C:\Users\Michi\Downloads\Cars.csv')

# create a list which will be later filled with the matching cars
qualified = []
# a variable counting the matching cars
hits = 0

# displaying the data
for index, row in data.iterrows():
    print(row["ID"], row["PRICE"], row["COLOR"], row["HP"], row["SEATS"])

# conversational recommending system, let's play around with asking the user out about his or her preferences regarding
# price and horsepower
print("Hello dear customer!")
price = int(input("How much are you willing to pay for your new car?"))
horsepower = int(input("How much horsepower should your new car have?"))
# iterate through the dataframe
for index, row in data.iterrows():
    if (price >= row["PRICE"]) and (horsepower <= row["HP"]):
        # counting +1 for each matching car
        hits += 1
        # fill the list with matching cars
        qualified.append(row["ID"])

```

```python

# delivering recommendation + an explanation after considering the user's constraints about price and horsepower
if hits == 0:
    print("Unfortunately there are 0 cars matching your constraints.")
elif hits == 1:
    print("There is" + " " + str(hits) + " " + "recommendation about cars for you as we considered your constraints"
                                                " regarding price and horsepower.")
    print("Recommended car: " + " " + str(qualified))
else:
    print("There are" + " " + str(hits) + " " + "recommendations about cars for you as we considered your constraints"
                                                 " regarding price and horsepower.")
    print("Recommended cars: " + " " + str(qualified))

```