Faculty of Economics and Social Sciences

Seminar Thesis
E-business & Recommender Systems

Recommender system for the industry of tourism: which are the most interesting places to visit?

Authors: Andrea Guidicelli (09-213-929)
          Fabio Sodani (08-676-926)

Examiners: Prof. Andreas Meier
           Dr. Luis Teran

Supervisor: Aigul Kaskina

Fribourg, May 3, 2015
# Contents

1 Introduction ...................................... 1
   1.1 Overview ....................................... 1
   1.2 Structure of the paper .......................... 2

2 Why this project? .................................. 3
   2.1 Problem analyzed and goals of the project ...... 3
   2.2 The role of recommender systems ............... 3

3 The Prototype ..................................... 5
   3.1 Portal web ....................................... 6
   3.2 Samples of data and construction of the relational database (users, items) ................. 6
   3.3 Search, filtering and recommendation methods .... 7

4 The interface ..................................... 11
   4.1 Creation of the user profile and login .......... 11
   4.2 Choice of the activities .......................... 14
   4.3 Filtering and recommender systems .............. 15
   4.4 Evaluation of the activities ....................... 16
   4.5 Introduction of new activities ................. 16
   4.6 My evaluations .................................. 17

5 Conclusion ...................................... 19
   5.1 Taxonomy analysis ............................... 19
   5.2 Discussion and final observations ............... 20
List of Figures

3.1 Logo of our prototype ................................. 5
3.2 Class diagram of the database .......................... 6

4.1 User account creation. .................................. 12
4.2 User login. ............................................. 12
4.3 Welcome page. ......................................... 13
4.4 Selection of an activity. ................................. 14
4.5 Introduction of the preferences for a type of activity. . 14
4.6 Recommendations on the web platform. ................ 15
4.7 Details of an activity. ................................. 16
4.8 Grading system. ..................................... 16
4.9 New activity creation, step 1. .......................... 17
4.10 New activity creation, step 2. .......................... 17
4.11 Section My activities ................................. 18
Chapter 1

Introduction

1.1 Overview, goals, people involved

This paper is a seminar thesis on Electronic Business and Recommender systems. Here will be introduced the problem that we want to model and solve using tools like a portal web, some matching algorithms and finally recommender systems.

The content concern the decisional process of choosing a good plan for a trip. When a person want to do this, it’s necessary to take in to account more aspects like cost, dates, topics to visit, activities to do, and many other factors. In our world we are faced to a very large number of suggestions like museums, squares, monuments, churches, landscapes but also any type of restaurant or hotels, and for this reason it’s not easy to know where we really want go. Our goal is to make this process easier with using recommender systems. We will inform our customers about the exactly location, price, kind of activity, background theme and general opinion of each location that they could or want visit. A user just have to answer to one of the following questions: "Which city do you want to visit?" or "What kind of activity do you want to do?" or both. According to the opinion of other users, some locations will be advised. For example, if a person visited the theater la Scala in Milano and he liked it, probably he will like also at the Opera in Paris.

The main idea of our project is to make this process easier by recommending some places where a person with a given profile could be interested on going. Some analyses of the profiles, a scoring system, some matchings between others users and some different objects will be done to achieve this target.

As final output, we will create a portal web and a database where customers and locations will be registered and evaluated with some parameters that we need to identify (for people we have for example factors like age, hobbies, location preferences). This will be done thanks to some prepared questions to make clear all the profiles by the system.

The following questions will be answered in the thesis:

- Which factors should we consider to create a good user profile?
- Which parameters should we consider to give a useful evaluation of each item?
• How recommender systems could play a significant role in the choice of the locations to visit (from the interface point of view)?

• What are the benefits using this method?

1.2 Structure of the paper

First of all we introduce the problem statement and the role of recommender systems methods that we want to use and describe. To do this we will use formulas describing the User to User and Item to Item methods using Pearson and Cosine algorithms. In addition to that a general non-personalized score of each item and a profile filtering between users will be done.

The second part of this paper shows how concretely we will construct a prototype managing the problem described in the first part. A user interface in PHP will be developed, so that final users could interact with the system, which will analyse their needs and profiles to be able to recommend them with many different advices (here the role of recommender systems).

In the third part of this work we focused the attention on the interface point of view. We will show how users can interact with the system using the web portal. Some images will have been used to clarify this concept. The final output of this part is a little tutorial on our prototype.

We decided to keep the taxonomy analysis as the final part. Considering the fact that it’s a new and original web portal using recommender system, we think that the taxonomy analysis could be a good conclusion of the work, so that our prototype could be classified.
Chapter 2

Why this project is linked to recommender systems?

2.1 Problem analyzed and goals of the project

The overall objective is to create a prototype based on a web portal, who allow to filter informations about locations, search and analyse potential interesting geographical places to visit during a trip.

To sum up, the specific aims of this problem are the followings:

- Identify each item who belong to a city as clearly as possible, using some key parameters and questions.
- Show the quantitative and qualitative importance of algorithms and recommender systems in this type of decision process.
- Create a user friendly interface easy to understand and which could be immediately integrate as a planning tool for travelling.

2.2 The role of recommender systems

In this chapter, the important role of recommender systems in those processes will be shown and understand. The algorithms who allow to send and receive recommendations are an interesting tool decreasing the planning time and the optimization of the satisfaction of the final users.

When a Client access into the Portal, the system will show a list with some top voted activities gave by all the users involved, without any required input. It’s of course a non-personalized recommendation to give some advices as soon as possible. After that three kinds of filtering and recommendation methods will play an important role.

The first advice will be gave by a filtering method comparing the preferences of the user for a specific type of activity (like restaurants or museums for example) with all the others users’ profiles. Just giving some informations about himself, the user can be recommended so that he doesn’t need to search too much to find a good restaurant for him. It’s clear that using
these two kinds of recommendation, is possible to take quick decision. For this reason our prototype could play a role for trip planning a few times in a year, but he could also be used every day just to choose new places to eat for example. Finally, thanks to methods like User to User and Item to Item, we can recommend the users with a really good personalized system. We will consider an historical database of votes and preferences, starting from all users and ending with the most similar profiles.
Chapter 3

The Prototype

To be closed to the concept of give an overview of some locations (or even areas if we consider more buildings in a same city for example), we choose to represent our project and prototype with a giraffe (see figure 3.1). In fact, using its property of height, the giraffe can see better of anyone else the region and give a look at all small places which are hidden somewhere. Using this property we can recommend people and give satisfaction to them. We will see that this choice has had an impact in terms of ergonomics (we will keep the color yellow, which also recall the yellow pages as interesting tool of informations) and the rating system’s interface (we decided to use leaves instead of the more used five stars to give more personalization to the web portal).

Figure 3.1: Logo of our prototype
3.1 Portal web creation and users account

The users’ population in our project are all the people who wants to plan a trip. As we are constructing a prototype we would like to do analyses based on a sample composed more or less by 20-30 places and 20-30 users (we should probably choose some cities and some places like restaurants, hotels and museums for example, which are located in these cities). We decided to create fake but plausible data to be as close as possible to reality. We think that it’s more interesting to have concrete and real data once the prototype will be ready, to see if it’s working as well as we want in practice.

In the Portal web, every data is persistent thanks to a relational database (MySQL) that will save every information of every entity, like general user information, places, activities, museums, restaurants and so on (see figure 3.2). Data will be saved in a perfect structure (entity-relations) to help us to transform data in information that we will use to recommend at the user what to visit, where to eat, where to sleep etc. to have the best satisfaction for the client. As you can see in this figure, a country contain some cities,

![Figure 3.2: Class diagram of the database](image)

who are assigned to some activities. Those must have a type of activity (like restaurants, museums and so on) and are submitted to different type of evaluations voted by some users who have already did those activities. Each one of the class drew in the diagram is a table in the database of our prototype.

3.2 Samples of data and construction of the relational database (users, items)

When the connected User will finish to select his activity that will do, he has the possibility to evaluate it, using a questionnaire with different evaluations. To create this matrix, we use numbers that have been calculated as means, considering the user evaluation of each parameter of a building, according to the following formula:

\[ e_{i,j} := \frac{\sum_{k_a \in K} e_{i,j,k_a}}{\#k_a}, \quad \forall \ i, j, \]
where \( i \) are the users, \( j \) the items (or buildings), \( k_a \) the parameters or characteristics of each activity \( a \) (here we mean the type of activity like restaurants, theatres, and not the specific building like Opera of Paris).

After that, the system will save inside the database the data and it will update the Utility Matrix to use for the next recommendation.

Here we have a list of all tables (or matrix) composing our database (● are numbers between 1 and 5 or an empty value):

<table>
<thead>
<tr>
<th>restaurant</th>
<th>quality</th>
<th>quantity</th>
<th>price</th>
<th>atmosphere</th>
<th>waiting time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_1 )</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>( u_2 )</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>...</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>( u_n )</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>theater</th>
<th>sound quality</th>
<th>fancy dress</th>
<th>price</th>
<th>atmosphere</th>
<th>interpreted works</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_1 )</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>( u_2 )</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>...</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>( u_n )</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>museum</th>
<th>quality</th>
<th>quantity</th>
<th>price</th>
<th>complexity</th>
<th>waiting time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_1 )</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>( u_2 )</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>...</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>( u_n )</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cinema</th>
<th>sound quality</th>
<th>video quality</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_1 )</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>( u_2 )</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>...</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>( u_n )</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

3.3 Search, filtering and recommendation methods

In this chapter we will speak about the methods that we used to recommend our users.

Non personalized recommendations

First of all we recommend users with a simple rank of all the grades that has been given for one activity. This classification will already be visible by users. To do that we use the previous formula (see chapter 3.2) adapted for more evaluations as follows:

\[
nprec_j := \frac{1}{\#i} \sum_{i \in I} e_{i,j}, \quad \forall \ j,
\]

where \( e_{i,j} \) are the evaluations of an item \( j \) given by a user \( i \).
A filtering user method, Euclidean distances

This type of recommendation is particularly efficient for new users. In this case we don’t have any vote or just a few votes if the user still begin to use our platform. For this reason we’ve decided to calculate recommendations based on the users’ profiles and preferences. When a user decide to search on our website one kind of activity for the first time (like restaurants for example) he should give us some data concerning his preferences. An accurate management of these data allow us to predict some personalized recommendations.

First of all a user should give an evaluation of his preferences with a number between 0 and 100. After that the system will calculate the percentage of importance of each characteristic just dividing this number by the sum of all votes gave to evaluate a type of activity. We will do it for each characteristic of each type of activity. This number has been called $u_{i,k,a}$, for each user $i$, each characteristic $k$ and each type of activity $a$.

We calculated the Euclidean distances from the new user’s profile to all the others existing using the following formula:

$$
ed_i = \sqrt{\sum_{k_a,k'_a \in K} (u_{i,k_a} - u_{i,k'_a})^2}, \quad \forall \ i \in I,$$

where $u_{i,k_a}$ are the user profile characteristics $k$ of a type of activity $a$ for a user $i$.

We choose to calculate the Euclidean distance instead off a simple difference of each characteristic, so that big incongruities will be penalized.

At this point we just have to select the top $n$ ($n$ is a desired entire number) minimal numbers to have the most closer profiles for the new user. The last step of this method is to choose the best votes $e_{i,j}$ for the selected $n$ users $i$. Finally we will have some recommendations of activities $j$ to do for the new user.

It’s obvious that using this approach, we can implement the $ed$ formula without the squared root. We will choose exactly the same user in both cases (which is the closest one, with the $ed$ number nearest to 0).

User to User, Pearson method

We decided to implement the User to User method with the Pearson formulation to recommend our users on some activities. This approach is very good especially for users with some activities already voted. We’ve just find some problems for users with only one evaluation and users who gave exactly the same votes for all the activities that they did. We will see exactly were we have this problem looking at the formulas that has been implemented in the system. To explain how the User to User recommendations work, we fix a user $u_x$ who will be recommend.

First of all we consider the matrix user-item with all the votes gave by all the users on all the items. We have to be careful on the difference between the value 0 for a vote or the non-voted value (empty). The first step of
the implementation is the construction of the correlation array between two users, which are the user $u_x$ and all the others $u_i$. The following formula explain this concept:

$$\rho_{x,i} = \frac{\sum_{j \in J} (e_{x,j} - \text{mean}_{e_x}) \cdot (e_{i,j} - \text{mean}_{e_i})}{\sqrt{\sum_{j \in J} (e_{x,j} - \text{mean}_{e_x})^2} \cdot \sqrt{\sum_{j \in J} (e_{i,j} - \text{mean}_{e_i})^2}}$$

where $e_{i,j}$ is the vote of the activity $j$ gave by user $i$, and $\text{mean}_{e_i}$ is the mean of all votes gave by user $i$.

Using this formula we will obtain an array containing the correlation between the user $x$ and all others. As we anticipate, there will be problems if there exist only one vote or if there is always the same vote gave by a user $i$ or our user $x$. In this case the mean of all votes and the singles votes will be equals and the consequence is that we will obtain a number divided by 0 in our calculation, which is clearly not a good input for the system. For these exceptional cases we won’t consider the User to User method to give recommendations.

After that we will select for example the top 5 best correlated users (we could consider a number of compared users as required by a specific situation), so the values most closer to 1 (except the user $x$ of course). With a big quantity of data, we will automatically consider only the correlation near to 1.

The next step is to calculate the predictions of vote for all activities not already did by the user $x$. To achieve this target we will use the following formula, where $y$ are the users with the top selected correlations:

$$\text{uup}_{x,j} = \text{mean}_{e_x} + \frac{\sum_{y \in I} \rho_{x,y} \cdot (e_{y,j} - \text{mean}_{e_y})}{\sum_{y \in I} \rho_{x,y}}$$

In this case we should mention that if a user $y$ has not voted an activity $j$, we don’t have to keep the value $\rho_{x,y}$ for the numerator and the denominator, but the value empty (so we don’t sum this term).

The last step is clearly to recommend the top desired activities $j$ to the user $x$.

**Item to Item, Cosine method**

We start considering the matrix of votes that users gave to items. Also this time we have to fix the user connected on the web portal $u_x$, who will be recommend. Except for the activities that $u_x$ already voted, we have to create a matrix containing the similarities between all the other items.

First of all we calculated the quadratic sum of all votes that an item receive, according to the following formula:

$$L_j^2 = \sqrt{\sum_{i \in I} e_{i,j}^2}, \quad \forall j \in J,$$

where $e_{i,j}$ are the votes gave by a user $i$ for an activity $j$. The result is an array containing a number for each activity $j$.  

The second step is to calculate the similarity item matrix, for each non-voted item $j'$ by the user $u_x$. To do that we will use the following formula:

$$sim_{j',j} = \frac{\sum_{i \in I} e_{i,j'} \cdot e_{i,j}}{L_{j'}^2 \cdot L_j^2}, \quad \forall j' \in J.$$

The last step of this method is to give the recommendation to the user $u_x$ using the Item to Item approach. The following formula shows us this result (here we have to consider only the similarities if $e_{i,j}$ exist, and so only if the activity $j$ has already been voted by the user $i$):

$$ii_p_{i,j'} = \frac{\sum_{j \in J} sim_{j',j} \cdot e_{i,j}}{\sum_{j' \neq j} sim_{j',j}}.$$

To conclude this method, we will choose the top desired values referred to the most predictable interesting activities to the user $u_x$.

Even for this algorithm we remark some problems, applying it to our prototype. It’s particularly important to choose to use recommendations based on the Item to Item approach, if the connected user has already voted a large number of activities (items in this algorithm). If it’s not the case we will obtain some non-accurate recommendations.
Chapter 4

The interface: Visualization and interaction with recommended systems

This chapter is a kind of tutorial of the web portal that has been implemented using the programming language php concerning the server side and javascript for the client side (see appendix to find more informations concerning some parts of the source code). Here we will explain all the steps that a user should do to be recommended.

4.1 Creation of the user profile and login

First of all a user should be identified, just giving a nickname, an address e-mail and a password. The figure 4.1 shows how it’s possible to do this concretely.

Once that the user has been inserted in the system, he can log to the web portal just inserting nickname and password, as it has been showed in the figure 4.2.

The first page appearing after the login is given by the figure 4.3. This picture give an overview of all options gave by the platform. In the following steps of this tutorial we will analyse each of them, using some print screen to show concretely how each aspect has been presented. We have a search tool bar, the description of the user profile (indicated with his Nickname), the possibility to choose and be recommended on some activities (Find activity) and the option to see all the evaluations already given (My evaluations). To insert a new activity, the user must be in his profile description (so click on his Nickname).
Figure 4.1: User account creation.

Figure 4.2: User login.
Figure 4.3: Welcome page.
4.2 Choice of the activities

The next step concern the choice of the activities to do and the profile creation of the new user. When he will do an activity for the first time, like for example be recommended on restaurants (as in the figure 4.4), he should introduce in the system some preferences.

![Figure 4.4: Selection of an activity.](image)

These parameters are exactly the same gave in the four tables of the chapter 3.2. Our web portal allow to introduce a number between 0 and 100 for each of these parameters. After that, the importance of each aspect who describes a type of activity are calculated, just doing the calculation of the percentages. This interaction between the new user and the web portal is represented by the figure 4.5.

![Figure 4.5: Introduction of the preferences for a type of activity.](image)
4.3 Filtering and recommender systems

In this chapter we show how the user will be recommended, which is the most important aspect of this prototype. For a didactic reason, we decided to keep separated the four methods of recommendation. We could put all the results together and just advice the top activities to the users, no matter the methods used to achieve this choices. If we would like to apply it to the reality, our prototype we will probably put all together, but in this context we prefer show more details of the structure behind recommendations. The figure 4.6 show how users are recommended. To see more details about the activities recommended, the user can ”click” on the name of the activity. A window like the figure 4.7 will be open. Here he can find information like address, description, observations, comments and the mean of all votes given to this item.

![Find an activity](image)

Figure 4.6: Recommendations on the web platform.
4.4 Evaluation of the activities

Once the item has been visited, the user can vote it. According to our logo, we decided to choose a grading system like the five stars adapted to five leaves. The goal of this evaluation is to be quick, so that every user will do it. They just have to evaluate each one of the parameters describing a type of activity. We think that a focused selection of this parameters will give a good evaluation of the final activity. According to the figure 4.8, it’s clear that feedback of an activity could be gave in a few seconds, just selecting a number between 1 and 5 leaves for each of the five parameters identified.

4.5 Introduction of new activities

One of the main goal of this project is to expand the knowledge of the people about tourism. It means that every user having access to our we portal can be recommend for example on a little unknown but really nice restaurant,
4.6. MY EVALUATIONS

located in the middle of nowhere. To make this process possible, we need to allow that each user can introduce some new activities, that he decided to do, without consulting our system. The figures 4.9 and 4.10 show this process. We can also see that on the left of the page (see always the two figures) the user has an overview of all the activities that he created.

Figure 4.9: New activity creation, step 1.

Figure 4.10: New activity creation, step 2.

As you’ve seen in the previous pictures, a register of all the activities created by the user is available in the yellow box.

4.6 My evaluations

In this part of the platform, the user can see all the details that he voted, as you can see in the figure 4.11. To change a vote that he already gave, he just have to vote once again the activity. The last opinion will be saved and the old one removed.
### My evaluations

<table>
<thead>
<tr>
<th>Type</th>
<th>Activity</th>
<th>Type of Evaluation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cinema</td>
<td>CapDine</td>
<td>Price</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sound</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Video quality</td>
<td>4</td>
</tr>
<tr>
<td>Cinema Forum</td>
<td>Price</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sound</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Video quality</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Cinestar</td>
<td>Price</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sound</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Video quality</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Festival del cinema</td>
<td>Price</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sound</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Video quality</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Piatto il cinema</td>
<td>Price</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sound</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Video quality</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Restaurant</td>
<td>Lampara</td>
<td>Ambience</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Price</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quality</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quantity</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Waiting time</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.11: Section My activities
Chapter 5

Conclusion

5.1 Taxonomy analysis

To conclude this thesis we choose to do a taxonomy analysis, so that our prototype could be classified from the recommender systems point of view. The following table sum up this analysis.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Tourism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose of Recommendation</td>
<td>Make available the most possible data concerning items like museums, theaters, restaurants and so on, even if there are some small unknown activities.</td>
</tr>
<tr>
<td>Recommendation Context</td>
<td>Culture, free time, hobbies, holidays.</td>
</tr>
<tr>
<td>Whose Opinion</td>
<td>People like the user looking for the informations.</td>
</tr>
<tr>
<td>Personalization Level</td>
<td>One part is generic and non-personalized, and another part is personalized. Data and recommendations visible after the user’s search.</td>
</tr>
<tr>
<td>Privacy and Trustworthiness</td>
<td>Low risk, due to the fact that recommendations are based on public places.</td>
</tr>
<tr>
<td>Interface</td>
<td>Predictions on votes and graphical recommendations available on the web portal where the user is logged. To do that we need come explicit input like preferences, characteristics, some votes.</td>
</tr>
<tr>
<td>Recommendation Algorithms</td>
<td>We used 4 types of algorithms to filter and recommend. One of these is not personalized and we called him Non-personalized recommendations. The three others are personalized, and as been implemented using the Euclidean distance method to filter the user’s profiles, the Pearson method for the User to User approach and the Cosine method for the Item to Item approach.</td>
</tr>
</tbody>
</table>
5.2 Discussion and final observations

This chapter concludes the thesis. We would like to discuss some aspects of our prototype and propose some possible improvements to make this project usable in the reality.

First of all, we should develop a method allowing to distribute the prototype. The easiest way consists on showing it to agencies in the public domain of tourism. Associations promoting the activities located in an area could be interested in having a platform like ours, who can bring more and more users, even living really far away.

Before doing it, we should increase the performance of the web portal, especially focusing on the speed of reaction of the server. With a very large quantity of data (the growth of these could be exponential, if the prototype success) we could need too much time to give a recommendation. Two ways are possible to solve this problem. The first and easier possibility consists on choosing a NoSQL server, who will increase the speed of processing query of database. The second possibility is based on some improvement of the code that has been created to manage the 4 algorithms who allow to advice our users. This is of course the main important weakness of our prototype.

This is the main important aspect to manage. After that, we have some little and easier options to add to our prototype. Here we give just a few examples of possible improvements like send some alert messages to remember the user on grading the activities that he did (if he didn’t vote it). Another possibility is to integrate an option allowing to modify the activities created some time before. We could continue with more others possibilities, which in reality will be some updates or a 2.0 version of our prototype.

To conclude this chapter, we can consider the strengths of our prototype. As we could imagine after reading this paper, the main important strength of our prototype is the potential utilization expansion, even for small activities who can’t spend a lot of money to promote their activity. Another important aspect is the frequency of the platform utilization, which can be daily for activities like restaurants or cinemas, but also monthly or yearly for vacation planning (in this case the time spent on our platform will probably be more large, and the researches more detailed).
Bibliography


Source code with notes of the portal

User to User

```php
// USER TO USER
// matrix similairy
$arrMeanUser = $arrMean;
$i = 0;
$x = 0;
$numeratore = 0;
$denominatore = 0;
$den1 = 0;
$den2 = 0;
foreach ($arrMean as $value) {
    if (array_keys($arrMean)[x] != $SESSION['user'][id]) {
        // user connected & user are different
        if ($value != null) {
            $arrUsAct[array_keys($arrMean)[x]] = $rowAct[id];
            if ($arrUsAct[array_keys($arrMean)[x]] != null && $arrUsAct[$SESSION['user'][id]] != null) {
                $diffUserX = $arrUsAct[$SESSION['user'][id]] - $arrMean[array_keys($arrMean)[x]]
                $diffUserY = $arrUsAct[array_keys($arrMean)[x]] - $arrMean[array_keys($arrMean)[x]]
                $numeratore += ($diffUserX * $diffUserY);
                $den1 += pow($diffUserX, 2);
                $den2 += pow($diffUserY, 2);
            } else {
                $arrSim[array_keys($arrMean)[x]] = 0;
            } else {
                $arrSim[array_keys($arrMean)[x]] = 1;
            }
        } else {
            $arrSim[array_keys($arrMean)[x]] = 1;
        }
    }
    $denominatore = sqrt($den1) * sqrt($den2);
    if ($denominatore != 0) {
        $arrSim[array_keys($arrMean)[x]] = $numeratore /
    } else {
        $arrSim[array_keys($arrMean)[x]] = 0;
    }
}
```
```php
$numeratore = 0;
$den1 = 0;
$den2 = 0;
$denominatore = 0;
$x++;}
// get the best 5
$best = 0;
$x = 0;
i = 0;
$best5user = array();
$arrCorr = $arrSim;
if (sizeof($arrSim) <= 5) {
    foreach ($arrSim as $val) {
        if (array_keys($arrSim)[$x] != $SESSION['user']['id'])
        {
            $best5user[array_keys($arrSim)[$x]] = $val;
            $x++;
        }
    }
} else {
    foreach ($arrSim as $val) {
        if (array_keys($arrSim)[$x] != $SESSION['user']['id'])
        {
            if ($i <= 5) {
                $best5user[array_keys($arrSim)[$x]] = $val;
            } else {
                $index = array_search(min($best5user),
                $best5user);
                if ($val > $best5user[$index]) {
                    $best5user[$index] = $val;
                }
            }
            $i++;
        }
    }
    $x = 0;
    // final step
    mysql_data_seek($act, 0);
    $x = 0;
    $num = 0;
    $den = 0;
    $num1 = 0;
    $num2 = 0;
    while ($rowAct = mysql_fetch_assoc($act)) {
        $v = new Vote();
        if ($v->alreadyVoted($SESSION['user']['id'], $rowAct['id'])
            < 1) {
            foreach ($best5user as $v) {
                // correlation * (vote - mean)
            }
        }
    }
```
Item to Item

// ITEM TO ITEM
$num = 0;

// coeff L² part 1
foreach ($arrUsAct as $idUser => $arrUser) {
    foreach ($arrUser as $idActivity => $value) {
        if (isset($arrL2[$idActivity])) {
            $arrL2[$idActivity] += $value * $value;
        } else {
            $arrL2[$idActivity] = $value * $value;
        }
    }
}

// coeff L² part 2
foreach ($arrL2 as $val) {
    $arrL2[array_keys($arrL2)[0]] = sqrt($arrL2[array_keys($arrL2)[0]]);
    $arrKeyItem = array_keys($arrL2)[0];
}

foreach ($arrL2 as $valL2) {
    foreach ($arrL2 as $val) {
        // MATRIX I2I
// same item → 1
if (array_keys($arrL2)[i] == array_keys($arrL2)[x]) {
    $matrI2I[array_keys($arrL2)[i]][array_keys($arrL2)[x]] = 1;
} else {
    mysql_data_seek($listUser, 0);
    while ($rowListUser = mysql_fetch_assoc($listUser)) {
        if ($valL2 == 0 || $val == 0) {
            $matrI2I[array_keys($arrL2)[i]][array_keys($arrL2)[x]] = 0;
        } else {
            if (isset($matrI2I[array_keys($arrL2)[i]][array_keys($arrL2)[x]])) {
                $matrI2I[array_keys($arrL2)[i]][array_keys($arrL2)[x]] += ($arrUsAct[$rowListUser['id']] * $arrUsAct[$rowListUser['id']] / ($valL2 * $val);
            } else {
                $matrI2I[array_keys($arrL2)[i]][array_keys($arrL2)[x]] = ($arrUsAct[$rowListUser['id']] * $arrUsAct[$rowListUser['id']] / ($valL2 * $val);
            }
        }
    }
    $x++;
} $x = 0;
$i++;
}

$x = 0;
//ARRAY I2I with final results part 1
foreach ($matrI2I as $itemX => $arrItemX) {
    foreach ($arrItemX as $itemY => $value) {
        if (isset($arrUsAct[$SESSION['user']][id][itemX])) {
            if (isset($arrI2I[itemY][id][SitemX]);
                $arrI2I[itemY] += $value * $arrUsAct[$SESSION['user']][id][SitemX];
            } else {
                $arrI2I[itemY] = $value * $arrUsAct[$SESSION['user']][id][SitemX];
            }
        }
        if (isset($denomin[itemY]));
            $denomin[itemY] += $value;
        } else {
            $denomin[itemY] = $value;
        }
    }
    $x++;
}
$x = 0;$

//ARRAY IDI with final results part 1
foreach ($denomin as $val) {
    if ($val == 0 || $arrID[$arr_keys($denomin][$x]] == 0 || $SESSION[‘user’][‘id’]][$arr_keys($denomin)[$x]] == null || $arrUsAct[$SESSION[‘user’][‘id’]][$arr_keys($denomin)[$x]] != null) {
        unset ($arrID[$arr_keys($denomin)[$x]]);
    } else {
        $arrID[$arr_keys($denomin)[$x]] = $arrID[$arr_keys($denomin)[$x]] / $val;
    }
    $x++;
}

Euclidean Distance

//ALGO DISTANCE
$up = new User_preferences_type_of_eval();

//get values of other Users preferences
$lisOtherUserPref = $up->getOtherUserPref($POST[‘select_activity’]);

//get values of User preferences
$lisMyUserPref = $up->getMyUserPref($POST[‘select_activity’]);

//save into matrix weighted values (User connected)
while ($row = mysql_fetch_assoc($lisMyUserPref)) {
    $tot = $up->getSumWeightTypeActivity($SESSION[‘user’][‘id’], $POST[‘select_activity’]);
    $matrWeightMyUser[$SESSION[‘user’][‘id’]][$row[‘toe’]] = $row[‘w’] / $tot;
}

//save into matrix weighted values (Other Users)
while ($row = mysql_fetch_assoc($lisOtherUserPref)) {
    $tot = $up->getSumWeightTypeActivity($row[‘idUser’], $POST[‘select_activity’]);
    $matrWeightUser[$row[‘idUser’]][$row[‘toe’]] = $row[‘w’] / $tot;
}

$x = 0;

//insert into array the distance (valueUserConnected-valueOtherUser)^2+...
foreach ($matrWeightMyUser as $idUser => $arrWeightMyUser) {
    foreach ($arrWeightMyUser as $idTypeActivity => $weight) {
        $i = 0;
        foreach ($matrWeightUser as $weightOtherUser) {
            if (isset($arrDiffUser[$arr_keys($matrWeightUser][$i]])) {
                $arrDiffUser[$arr_keys($matrWeightUser][$i]] += pow($matrWeightMyUser[$arr_keys($matrWeightMyUser][$x]][$idTypeActivity] - $matrWeightUser[$arr_keys($matrWeightUser][$i]][$idTypeActivity]), 2);
else {
    $arrDiffUser[array_keys($matrWeightUser)[i]] =
    pow($matrWeightMyUser[array_keys($matrWeightMyUser)[x]][idTypeActivity] -
    $matrWeightUser[array_keys($matrWeightUser)[i]][idTypeActivity], 2);
} i++;
}
$x++;
}

$bestIdUser = array_search(min($arrDiffUser), $arrDiffUser);
$username = $u->getUsername($bestIdUser);
$maxValue = max($arrUsAct[$bestIdUser]);
$bestIdAct = array_search($maxValue, $arrUsAct[$bestIdUser]);