Food Recommendation System

Project Report

Ipek TATLI

1395557

June,2009

***Abstract*— This document discusses content-based recommendation systems, i.e., systems that recommend an item to a user based upon a description of the item and a profile of the user's interests. Basically it gives you a clear example of content-based recommendation systems, specifically a food recommendation system. The dataset that is used in this project is from "yemeksepeti.com". Similar foods are recommended according to previously constructed user preferences.**

***Keywords*— Information Retrieval, Recommendation Systems, Content- Based Approach, Expert Systems, Web Crawling, Feature Similarity**

1- Introduction and Problem Definition

People make decisions everyday. "Which movie should I see?", "Which city should I visit?", "What should I eat?"... There are too many choices and a little time to explore them all. Recommendation systems help people make decisions in these complex information spaces.

Recommendation systems are a type of information filtering that presents lists of items (films, songs, books, videos, images, products, web pages...) which are likely of user interest. Amazon, Last.fm, Ulike, iLike, Netflix, Pandora are the most popular recommender systems all over the world. Simply they compare user interest acquired from his/her profile with some reference characteristics and predict the rating that the user would give. Those characteristics may be from the item information (content-based approach) or the user's social profile (collaborative filtering approach).

I focus on the question "What should I eat?" in the scope of this project. My system uses content-based recommendation technique for producing food recommendations. It is based on similarity of foods. Basically, my system constructs user profiles from the previously rated features and food profiles from the ingredients of the food, then it recommends the most appropriate foods according to the preferences of the users.

This document is organized as follows. Overall system design is reviewed in the next section. System Architecture and experimental evaluation appear in Sections 3 and 4. Discussions about the problems I encountered are mentioned in Section 5. Conclusion and future work are presented in Sections 6.

2 - Overall System Design

2.1 - Recommendation Technique:

My system is a content based recommendation system. Food domain can be seen as a set of foods where each food has a set of ingredients. Content-based approaches treat the recommendation problem as a search for related items. Given a rated food, the algorithm constructs a search to find other related items with the same ingredient. If a user likes salmon, for example the system might recommend other foods having salmon like sushi. However, this is the simplest logic behind content-based recommendation systems. In my system foods are defined by their important features and represented by vectors. Thus, feature weights are crucial in these vectors.

Similarity is computed based on item attributes using appropriate distance measures. Content-based recommendation systems share in common describing the items that may be recommended, creating a profile of the user that describes the types of items the user likes, and comparing items to the user profile to determine what to recommend. My content-based recommendation system can be seen as a combination of three distinct parts; food profiling, user profiling and recommending foods according to the previous feedback of the users. Moreover, food domain can be seen as a set of features where each feature is actually an ingredient of that food. Therefore, both item and user profiles are kept as vectors of features. As my system is a content-based recommendation system, it tries to find best matches between the user profile and the food profiles.

2.2 - Item Profile Representation

In content based recommendation systems, every item is represented by a set of features or an attribute profile. A variety of distance measures between the feature vectors may be used to compute the similarity of two items. For example Euclidian or cosine similarity supposes that all the features have equal importances. However, human judgment of similarity between two items often gives different weights to different attributes. Moreover, document frequency is more commonplace to use for this purpose.

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}.$$

Here N denotes for the total number of foods in a collection, and dft is for the total number of foods that have the ingredient “t”. Thus the idf of a rare term is high, whereas the idf of a frequent term is likely to be low. This method is called “inverse document frequency” and I have assigned the calculated similarity as weights of the features.

2.3 - Knowledge Acquisition Technique & User Profile Representation

The type of the user profile derived by a content-based recommender depends on the learning method employed. Decision trees, neural nets, and vector-based representations can be all used. In this project I have used decision vector based representations constructed with the help of user ratings. At this step, my system uses explicit data collection. Specifically, after each recommendation, user can explicitly state whether the recommendation is satisfying or not. The next recommendations will be mostly based on this user feedback.

3 – Architecture

The implementation environment

I implemented my project with Php. The reasons that I choose this language is that it is simple, intuitive, easy to use, easy to control multiple users at the same time; has a perfect web integration platform, support for web services, XML & legacy systems etc. I developed it on Linux and used Mysql. As you know that Linux, Apache, Php and Mysql is a good quadruple called LAMP. Moreover, http is chosen as application layer protocol because it suits my needs and works well with today's Internet infrastructure.

3.1 – Information Retrieval

Information retrieval (IR) is a data search technology which includes crawling, processing and indexing of content, and querying for content. Web crawling is the process by which we gather pages from the Web, in order to index them and support a search engine. The objective of crawling is to quickly and efficiently gather as many useful web pages as possible, together with the link structure that interconnects them.

I have done web crawling to create my dataset. I have written some code to extract the food information from the web. My code connects to "yemeksepeti.com", it firstly extracts the main food types. Keeping them in a text file, then I have extracted the restaurants related to each main food type. I store those restaurants in a database table called "restaurants". Lastly, I extract the meals with their ingredients. I also store them in another database table named "meals".

The meals are also parsed into their ingredients. Another database table is created for the ingredients and their calculated weights.

3.2 – Item Profiling

My system first constructs item profiles from the information it collects from "yemeksepeti" database. It uses this information to build food profiles through assigning feature weights to each feature in the food domain. Ingredient weights are calculated with the method I previously explained.

3.3 – User Profiling

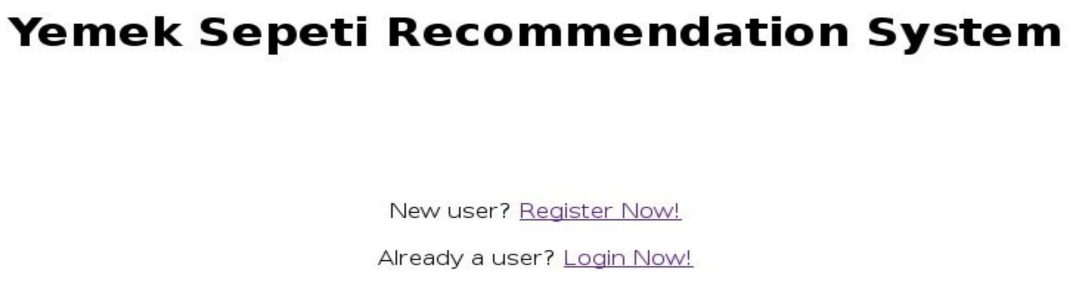
After the registration to the system, a set of random ingredients are asked to the user in order to be rated from 0 to 5. If an ingredient has already been rated before, the new rating will be considered in the next logins after registration. I made such an estimation that if the rating is smaller than 3, the user does not like eating that ingredient. Therefore, the ratings smaller than 3 are considered as negative ratings, the ratings equal to 3 are considered as neutral ratings, and the remainings (4, 5) are considered as positive ratings. Briefly, user profile vectors are constructed with the positively rated ingredients.

3.4 – Recommendation and User Feedback

The recommendations are done by calculating the foods which are most similar to the profile of the user. However, only the foods which have positively rated ingredients are considered in this similarity process. Moreover, users can evaluate the recommendations by stating whether they liked the recommendations or not. I made another estimation here. I assume that the user likes the recommendation according to the restaurant in which that recommended food can be eaten. This part is the user feedback step of my recommendation system. After these evaluations, the next recommendations are built on both personally liked (positively rated) ingredients and the previously liked recommendations, which are affects the similarity score positively.

4 – Evaluation of The System

I firstly implemented my system considering that all the features have the same weight. However I was not satisfied with the recommendations. I have read something abut Inverse Document Frequency and used it; then the results changed dramatically. The second version of my project was without the evaluation of recommendations (without user feedback part). I have used the system with a few people. Because of the limited time, it was not clear to see its development in making good recommendations. I have added the recommendation evaluation module later. I have used it with a few people, too; but it is making better recommendations day by day.

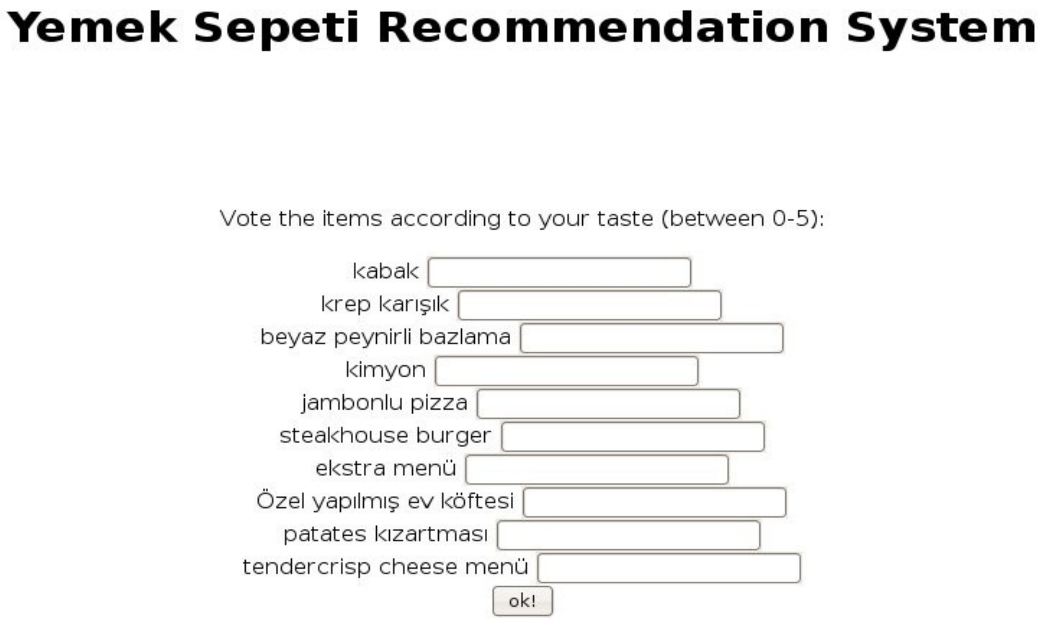


Yemek Sepeti Recommendation System

New user? Register Now!

Already a user? Login Now!

Figure 1 – Login/Registration The System

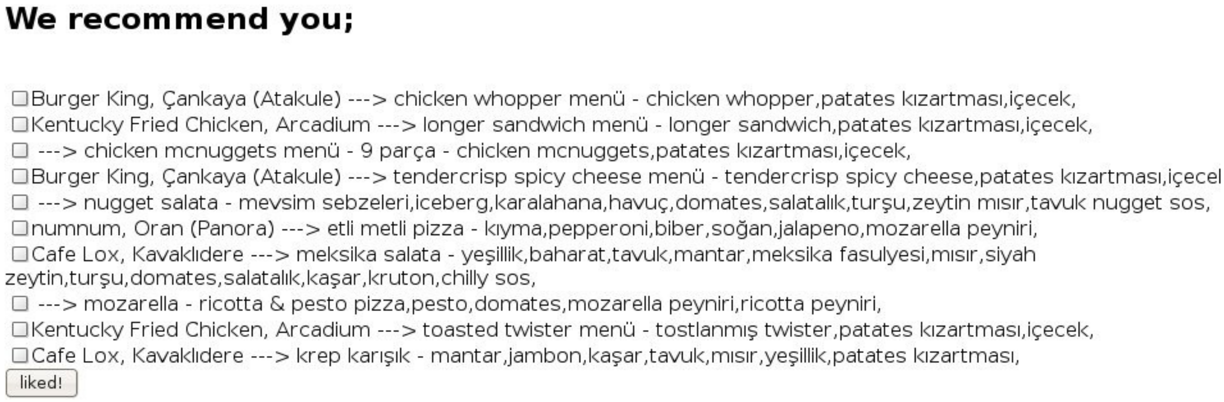


Yemek Sepeti Recommendation System

Vote the items according to your taste (between 0-5):

kabak
krep karışık
beyaz peynirli bazlama
kimyon
jambonlu pizza
steakhouse burger
ekstra menü
Özel yapılmış ev köftesi
patates kızartması
tendercrisp cheese menü

Figure 2 – User Profiling



We recommend you;

- □Burger King, Çankaya (Atakule) ---> chicken whopper menü - chicken whopper,patates kızartması,içecek,
- □Kentucky Fried Chicken, Arcadium ---> longer sandwich menü - longer sandwich,patates kızartması,içecek,
- □ ---> chicken mcnuggets menü - 9 parça - chicken mcnuggets,patates kızartması,içecek,
- □Burger King, Çankaya (Atakule) ---> tendercrisp spicy cheese menü - tendercrisp spicy cheese,patates kızartması,içecel
- □ ---> nugget salata - mevsim sebzeleri,iceberg,karalahana,havuç,domates,salatalık,turşu,zeytin mısır,tavuk nugget sos,
- □numnum, Oran (Panora) ---> etli metli pizza - kıyma,pepperoni,biber,soğan,jalapeno,mozarella peyniri,
- □Cafe Lox, Kavaklıdere ---> meksika salata - yeşillik,baharat,tavuk,mantar,meksika fasulyesi,mısır,siyah zeytin,turşu,domates,salatalık,kaşar,kruton,chilly sos,
- □ ---> mozarella - ricotta & pesto pizza,pesto,domates,mozarella peyniri,ricotta peyniri,
- □Kentucky Fried Chicken, Arcadium ---> toasted twister menü - tostlanmış twister,patates kızartması,içecek,
- □Cafe Lox, Kavaklıdere ---> krep karışık - mantar,jambon,kaşar,tavuk,mısır,yeşillik,patates kızartması,

Figure 3 – Recommendation and User Feedback

5- Discussions About The Problems I Encountered

The first problem I encountered was that “yemeksepeti.com” was not a good site to crawl because it keeps a cookie for each client and shows all the data in the same link. Although I navigate to other pages through buttons, the links of the pages do not change. Knowledge acquisition with web crawling was too difficult for me.

Another problem I want to mention is that “yemeksepeti.com” is in Turkish. There were so many spelling errors and sometimes some irrelevant data. Cleaning the data and fixing them manually was also a boring and tiring job.

The other problem is that the time was very restricted for me because I have spent most of the time with creating my dataset as I mentioned in the previous paragraphs. Therefore, I could not try my system with many people. Moreover, I have made the dataset smaller to make the query completion time shorter and to test easier.

I also wished to use Jess as a rule engine in my project but it can be used with Java. Jess is Java version of Clips. Then I found Phlips, the Php version of Clips. Its documentation was really bad and I have discovered many things about PhLips on my own. Nevertheless, I could not use Phlips for this project because of the difficulty of producing rules. Rule Production is a time consuming job and finding appropriate rules for this project was not a good idea for these versions of my project.

6 – Conclusion and Future Work

To sum up, designing this system was a very useful exercise and I believe that the literature survey I have done was vey useful for me in order to understand the logic behind the recommendation systems. I have also searched about some rule engines that I wanted to use in this project. I have implemented some tiny programs using Jess-rule engine; but I realized that constructing rules for such a food recommendation may be a difficult work. Then I gave up the idea of using Jess. Moreover, implementing such a project demonstrated how a knowledge based system can actually encapsulate knowledge. It also showed the limitations of knowledge based systems and the assumptions that must be made when designing them.As an instance for these assumptions, I have assumed that the user likes the recommendation according to its restaurant. Moreover, while finding the similarity between user profile and candidate item profiles, I was calculating a score with the weights and ratings. This calculation logic may also be leading to some failures. I should implement a more improved version of the algorithm for finding the similarity. Moreover, my system is static that if any other restaurant of meal is added, this does not affect my system. Dynamic knowledge usage can also be a god version in the future.

7 – References

1- Recommender System, http://en.wikipedia.org/wiki/Recommender_system

2- Content-based, Collaborative Recommendation,

www.ischool.utexas.edu/~i385d/readings/Balabanovic_Fab_97.pdf

3- Learning New User Preferences in Recommender Systems,

http://www.grouplens.org/papers/pdf/voi-final.pdf

4- Item-Based Collaborative Filtering Recommendation Algorithms,

http://www.grouplens.org/papers/pdf/www10_sarwar.pdf

5- OpenMore: A Content-Based Movie Recommendation System, METU Electronic Library

6- CLIPS: A Tool for Building Expert Systems, http://clipsrules.sourceforge.net/

7- JESS, http://en.wikipedia.org/wiki/Jess_programming_language

8- Inverse Document Frequency, *http://nlp.stanford.edu/IR-book/html/htmledition/inverse-document-frequency-1.html*

9- Feature Weighting in Content Based Recommendation

System Using Social Network Analysis

, http://www2008.org/papers/pdf/p1041-debnath.pdf