Towards Persuasive Social Recommendation: Knowledge Model

Javier Palanca
Departamento de Sistemas Informaticos y Computacion
Universitat Politecnica de Valencia. Camino de Vera S/N
46022 Valencia (Spain)
jpalanca@dsic.upv.es

Stella Heras
Departamento de Sistemas Informaticos y Computacion
Universitat Politecnica de Valencia. Camino de Vera S/N
46022 Valencia (Spain)
sheras@dsic.upv.es

Javier Jorge
Departamento de Sistemas Informaticos y Computacion
Universitat Politecnica de Valencia. Camino de Vera S/N
46022 Valencia (Spain)
jjorge@dsic.upv.es

Vicente Julian
Departamento de Sistemas Informaticos y Computacion
Universitat Politecnica de Valencia. Camino de Vera S/N
46022 Valencia (Spain)
vinglada@dsic.upv.es

ABSTRACT

The exponential growth of social networks makes fingerprint let by users on the Internet a great source of information, with data about their preferences, needs, goals, profile and social environment. These data are distributed across different sources of information (social networks, blogs, databases, etc.) that may contain inconsistencies and their accuracy is uncertain. Paradoxically, this unprecedented availability of heterogeneous data has meant that users have more information available than they actually are able to process and understand to extract useful knowledge from it. Therefore, new tools that help users in their decision-making processes within the network (e.g. which friends to contact with or which products to consume) are needed. In this paper, we show how we have used a graph-based model to extract and model data and transform it in valuable knowledge to develop a persuasive social recommendation system[1].

Categories and Subject Descriptors

I.2 [**ARTIFICIAL INTELLIGENCE**]: Distributed Artificial Intelligence
H.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Miscellaneous

General Terms

Algorithms

Keywords

recommender systems, data integration, social networks

1. INTRODUCTION

Over the last few years, social networks have changed the way users perform many of their daily activities on the Internet. Concretely, direct interactions between users with different final goals have won ground on mere search and navigation activities over stored information. Therefore, users have evolved from being mere consumers of information to real producers. In Spain, for instance, 93% of Internet users access social networks daily; 2 out of 3 take into account the recommendations of other users to make decisions (on products, treatments, entertainment, etc.); and of these, 69% gives a lot or some credibility to what their friends or acquaintances say in a social network.

However, the increasing number of users and information generated, the heterogeneity of the users themselves, their unpredictable behavior, and the dynamism of the network structure (members joining and leaving the network at their will, consequently evolving the network's structure), make users to deal with a high degree of uncertainty when choosing who to interact with or what information to consume [1]. In order to decrease this uncertainty, tools that help users in their decision-making processes within the network are required. A promising solution is the use of recommendation systems [2, 3], which are capable of performing effective recommendations to help users to make appropriate decisions.

Nevertheless, traditional recommender systems base their recommendations on quantitative measures of similarity between the user's preferences and the current items to recommend (i.e. content-based recommenders [9]), between the user's profile and the profile of other users with similar preferences (i.e. collaborative filtering recommenders [10]) and on combinations of both (i.e. hybrid recommenders [11]). However, [12] has stated the inability of current recommender systems to use the large amount of qualitative data available online to empower recommendations. Usually, recommender systems do not provide an explanation about the reasoning process that has been followed to come up with specific recommendations. Recommendations tend to come directly from the recommendation algorithm that runs the website and not from the acquaintances that a user has in his/her social network. However, this does not follow current trends on the Web, where discovering is becoming social and recommendations could be expected to come directly from acquaintances in a decentralised way. Moreover, people trust recommendations more when the engine can provide reasons for them [13] and when the recommendation comes from an acquaintance. Thus, what is understood as a good recommendation is changing from the one that minimises some error evaluation to the one that is really able to *persuade* people and make them happier.

Another problem with traditional recommendation approaches is that they barely take advantage of the huge amount of information underlying the network structure. Different pieces and sources of information must be treated in an uniformed way in order to improve the recommendation processes in social networks. This requires making projections between the vocabularies used as patterns in different data sources and merging data (instances) of different sources. There exists mechanisms in *RDF Schema* and *OWL* to express such relationships between terms of vocabularies and to perform transformations between schemes easily. There are also work reported in the literature about how to generate these alignments in a more or less automatic way [4, 5]. The main challenge in data fusion is the resolution of conflicts when different values for the same property of an object are obtained. There exists works in this line in the database research community [6], and the efforts on reconciliation of identities in the Web research community is also growing [7].

Following this approach, in this paper we propose the employment of graph databases as a knowledge model to integrate domain-specific and social data obtained from social networks, and use it to perform persuasive recommendations. Graph databases focus on the structure of the data, storing information as a network [8]. Data on social networks are increasingly interlinked and connected, and graph-based models allow to represent billions of nodes and relationships. Therefore, in contrast with *relational* and other *NoSQL* database models, graph-based databases have demonstrated their improved performance when dealing with connected data [8].

Our ultimate goal is to exploit the benefits of this model and use this valuable information to generate recommendations in a persuasive social recommendation system. Thus, the information stored in our graph-based models will be transformed in recommendations enhanced with explanations and arguments, able to persuade the user about the actual suitability of the recommendation that he/she has received.

The rest of the paper is structured as follow: section 2 presents a prototype of our persuasive social recommendation system; section 3 shows the proposed graph-based knowledge model; section 4 reviews related work; and, finally, conclusions are explained in section 5.

2. TOWARDS PERSUASIVE SOCIAL RECOMMENDATION

To develop and validate our knowledge model, we have focused our research on the domain of recipes recommendation. As pointed out later on section 4, there is a notable increasing demand of recommender systems to improve the health habits of their users and to help users to plan their meals when they have to observe certain dietary restrictions. More than 17 million Europeans suffer from some form of food allergy, according to the European Academy of Allergy and Clinical Immunology (EAACI), and this number is expected to increase in forthcoming years.

Currently, we are developing a prototype, based on our website `receteame.com`, which recommends recipes for its users. *receteame.com*[2] is a persuasive social recommendation system to make personalized recommendations about recipes to its users. The system retrieves recipes from the Internet and automatically calculates their nutritional information and dietary restrictions to use this information to make recommendations. We are also developing an intelligent algorithm (based on argumentation techniques and social network analysis) to learn the tastes and needs of each user and recommend fully customized recipes from two main sources of information: the votes that users give to each recipe, and, the activity of the user and friends in *Facebook*. Each time a registered user will search for a recommendation, the algorithm will receive a recipe recommendation query, including parameters describing the user profile (preferences and tastes, dietary restrictions, personal data, etc.) and the context of the query (e.g. if the user is looking for a main course of for a particular ingredient, the number of dinner guests, etc.). With this information, the algorithm will perform two main searches to select a potential set of recipes to recommend to the user. On the one hand, the algorithm will follow a content-based recommendation approach to generate a list of recipes that match the query. However, note that the accuracy of recommendations generated by this process completely relies on the amount and accuracy of previous votes that the user made to other recipes with similar characteristics. Therefore, it is highly influenced by the *cold start problem* (i.e. the performance of content-based recommenders is poor with new users that have not yet rated a sufficient amount of recipes) and the drawbacks of applying traditional recommendation approaches on large social networks (i.e. issues related with computational costs of getting an accurate recommendation, and loss of the big amount of related social information available in the network). To overcome these problems, the algorithm will perform an alternative search that will follow a social recommendation approach.

On the other hand, the algorithm will select a set of users of the system and spread the query to obtain recommendations from these users. This set will contain the set of friends of the target user, and if necessary, a randomly selected set of users to avoid the cold start problem when the target user is new on the system and still does not have an adequate number of friends. Each user that has received the query will select a set of recipes that match the original query from his own set of known recipes (those voted by this user). Then, for each user, this part of the algorithm will generate an ordered list of recipes to recommend according to two criteria: the preferences of the user that is being asked for recommendations, for instance, taking into account the votes of the user; and the preferences of the target user, for instance, taking into account the votes of the target user to a recipe (if any). Note that this process will be performed automatically, without the direct intervention of the actual users, but the algorithm will be in charge of retrieving and

[2]http://www.receteame.com;
http://buscador.receteame.com

managing the necessary information to perform these tasks and simulate the interaction between users.

With the full set of recommended recipes from other users, the algorithm will make an overall ranking of recipes employing four social criteria parameters: 1) the trust on the user who had recommended a recipe from the point of view of the target user and his friends. This parameter will be calculated by using a direct trust evaluation between these two users, and, if any, aggregating the trust evaluations of the friends of the target user that are also friends of the user that made the recommendation; 2) the reputation of the user who had recommended a recipe. This is a global parameter that will be calculated by computing the average trust regarding all recommendations made by one user to his friends; 3) the strength of the friendship between the target user and the user that had recommended the recipe. This parameter will be calculated by using several predictive friendship variables [15] and will depend on the activity of the target user on the social network where the algorithm operates (*Facebook* for now); and 4) the similarity between the target user and the recommender user in terms of their preferences. The result of this process will be a unique and ordered list of recipes to recommend to the target user. Finally, the algorithm will mix the recommendations that has obtained from both searches, assigning weights to bound content-based and social-based recommendations, and will select the best recommendation to propose. This process will also include an internal agreement procedure based on argumentation techniques, which will allows the algorithm to propose first those recommendations for which it will be able to generate better justifications [16]. This justifications will be arguments to explain the user the suitability of the recommendation provided, showing pieces of information that could persuade him/her to accept and put into practice such recommendation (e.g. to show a celiac user that enjoys chocolate cakes that the recommended recipe to cook a chocolate muffin has no gluten and is specially recommended by a trusted user in his/her social network).

As pointed out before, this system is currently under development and we focus on this paper on the knowledge model that we are using to extract and model our data to transform it in valuable knowledge for our persuasive social recommendation system.

3. GRAPH-BASED KNOWLEDGE MODEL FOR PERSUASIVE SOCIAL RECOMMENDATION

In this section we show how information is retrieved and integrated into a fast and highly available graph-based database that allows us to perform the recommendation process in an efficient and effective way. For our recommendation process we need to retrieve two types of data: social data from the users, and domain-specific data. As domain-specific knowledge, we use a big amount of recipes (over 10.000), with nutritional information about their health issues, diseases for which they are encouraged and discouraged, nutrients of each ingredient and their relationships with health and dietary labels. In what follows, we present how we have retrieved both social information and recipes information and how we represent this data in our graph-based model.

Due to the users heterogeneity and dynamism, we have to manage an enormous quantity of data that is constantly generated in an efficient way. Thus, the traditional databases main drawback (i.e. the data consistency (ACID[3] philosophy)) makes them inappropriate for our purposes. In addition, in an environment as interrelated as a social network, it becomes necessary to store the semantics that is underneath the interaction of the users (i.e. we are more interested in the relationships than in the content). The need to include new information in the database that may be related with hundreds of already stored data (e.g. users) will imply an excessive computational cost, and this is precisely one of the features for which the NOSQL databases (guided by BASE[4] principles) show their potential. Concretely, in our domain the entities can be interrelated among them, with a high growth and dynamism in such relations. Therefore, among the different types of NOSQL databases, we will work with graph databases. This type of databases are highly scalable and close to the natural structure of the data.

3.1 Social Information

In this section, we present the knowledge model that we use to store and manage the information about the user who wants to receive recipe recommendations by using the persuasive social recommender that we are developing. For this task, the site *receteame.com*, provides us an interface to interact with the user.

In this site, the user could register or log in with the user's *Facebook* account. The *Facebook* registration process is carried out by means of OAuth 2.0 protocol[5], that is widely used on social networks and applications that work with them. This method facilitates the access to HTTP services in a restricted way (setting limitations to the specific pieces of the user's *Facebook* information that the system is able to access). These limitations are imposed by the user, who provides permissions to the application in the register phase (i.e. the application requests some permissions to access to the user's mail account, birthday, or friends list, among others). This is a sensitive step of the process, due to the reluctance of users when they are requested for granting permissions related to their personal data. For this reason, we made available online a demo of our system and achieved more than 2000 registered users to test how users react to our permissions request. Our tests showed successful results, getting those permissions in over the 80% of the cases (getting access to very sensitive information such as the inbox messages, shared links and tagged photos).

Once the registration process has finished by means of the Facebook account, the application gets a token that identifies the relationship between the user and the application. With this token, the application can obtain the user's node that is stored in the *Facebook*'s databases. This node enables the application to obtain the information about the user's activities in the social network, such as "likes", comments or tags. Only the information related with the user's interactions in the social network is stored in the application's database (we anonymize other personal data to observe data protection laws, and the contents that the user has in the user's account, such as messages or posts, are not stored).

Figure 1 shows the graph-based model with the information about the user that we are able to obtained with this

[3]Atomicity, Consistency, Isolation and Durability
[4]Basically Available, Soft state, Eventual consistency
[5]http://oauth.net/2/

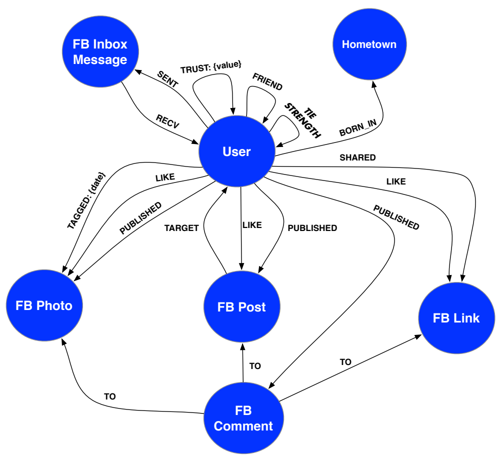

Figure 1. Graph of the user

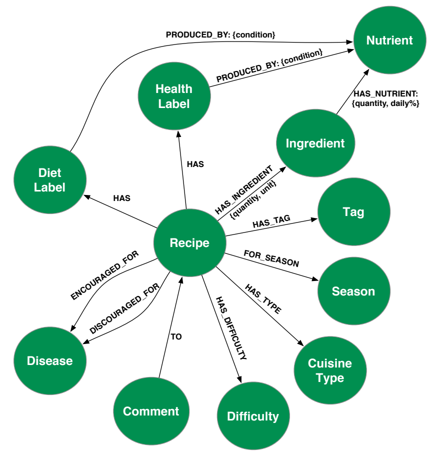

Figure 2. Recipes model graph

process. This enables the algorithm BFF[15], by means of the relations established among the users (with comments, friendship, likes, posts and tags), to provide us a *Tie Strength* value between each pair of users related in the network. This value establishes a degree of friendship between two users on the social network.

3.2 Domain Information

In this section, we present the knowledge model that we use to store and manage the information about the recipes that our system will recommend. Making recipe recommendations requires to have a big database of valid recipes that may result interesting to users. The bigger the database is, the most varied and accurate recommendations can be delivered to users. Recommending recipes also presents a special characteristic; although people may be interested in world wide recipe discovering, they usually tend to be interested in local cuisine that uses products and ingredients that they can easily obtain. This is also the rationale of *the local food movement*, which tries to reduce the distance between food producers and consumers and achieve a more sustainable food chain. Other important feature when recommending recipes are cooking techniques, since users tend to be interested in cooking what they can afford. For instance, it makes no sense to propose a recipe that requires a *tandoor oven* (a cylindrical clay or metal oven used in Southern, Central and Western Asia, as well as in the Caucasus) to users from Western Europe, since they have not easy access to this kind of cooking appliances.

These considerations have guided us to build a recipe database focused in a unique country (Spain), with *Mediterranean* eating habits and a very rich cooking tradition. Spanish gastronomy is varied and rich, making use of lots of fresh ingredients like vegetables, fruit, fresh fish and shellfish, and ingredients that are country-specific while famous, like olive oil or Iberic ham. These ingredients are accessible everywhere in Spain but are not as common outside the country. Then, there are three ways of building a good recipe database that accomplishes these characteristics: to hire nutritionists and chefs to write it, which is too expensive, to let users to upload their own recipes, which is too slow, and to crawl the web looking for the recipes. The latter was the method followed in this work.

In order to locate semantically correct recipes we made use of microformats [14]. Microformats (μF) is a semantic markup language that extends HTML with semantic tags that allow web pages to be processed automatically and extract data that contains semantic information intended for end-users. Two of the most known microformats for recipes are *h-recipe* and *Schema.org Recipe*. Both are very similar and use classes or `itemprops` to tag content. Some tageable information for recipes is *preparation time, recipe instructions, description*, or *ingredients*. Recipe microformats have also nutritional information that can be useful (when present) to classify recipes for users with special requirements. Figure 2 shows the information that we have retrieved and stored in our graph database.

The non-repeatable information is stored in the *Recipe* node (e.g. the recipe title, description, image, instructions, preparation time, servings, etc). On the other hand, all the information that may be useful to be represented as relationships has been modeled in separated nodes connected by relationships (e.g. the list of ingredients, health and diet labels, tags, difficulty, etc). With this graph representation it is easy (and quick) to find recipes that share a set of ingredients, or that are appropriate for a specific allergy (looking at the encouraged and discouraged diseases relationships or the health labels). The nutritional information is also represented in this graph, connecting all the nutrients contained in an ingredient with the health labels and diet labels which are related to. This nutrient information has been extracted from the United States Department of Agriculture, who has published a National Nutrient Database for Standard Reference[6].

Our crawling algorithm based on microformats has gen-

[6]http://ndb.nal.usda.gov/

erated a database of more than 10000 recipes that can be extended dynamically. The database growth rate is limited by the poor number of websites that use microformats (and even more limited if we focus only in recipes from a specific country or region). Microformats are very extended for calendar events (h-event) and people information (h-card), but are still not widely adopted to represent information from other prepared models, like *h-recipe*, *h-product*, *h-item*, etc, or the *schema.org* standards like *Restaurant*, *Review*, *Product* or *Health*.

3.3 Full Knowledge Model

Once explained how data have been obtained and organized following a graph-based model, the next step consists on the integration of both social and recipe information into a full graph-based knowledge model. Figure 3 shows how this integration has been defined. The use of the graph-based model allows a natural fusion of the obtained information, just adding the necessary relationships among existing nodes of both social and domain-specific data. Most relationships are between the *User* nodes and the *Recipe* nodes. These relationships between nodes mainly represent events that have occurred in the system. The relationship *VIEWED* represents that a user has viewed a recipe at a specific date. The same applies for the *RATED* relationship, which means that a recipe has been rated by a user with the rating *rate*. There is another relationship between *Recipe* and *User* but, this time, in the opposite direction. This relationship represents the event that a recipe was recommended to a user at a specific date. Finally there are relationships that do not represent events but semantic information, such as a disease that a user suffers, a diet label a user is interested in (i.e. *HIGH_PROTEIN*), or the user that has published a recipe.

One of the main advantages of using this approach is that the produced overload in the model due to the increase of the number of nodes is considered as negligible comparing with traditional approaches. Therefore, scalability is not a problem. One important feature of the proposed model is that there exists a clear imbalance between the nodes needed to represent the social data and the recipe data. After the extraction process, the number of recipes were around 10000 and the number of nodes needed to store the social information of users were around hundred thousands. Moreover, the dynamism of this kind of information can produce an exponential evolution. Nevertheless, our persuasive social recommendation prototype perfectly supports these orders of magnitude.

Another important issue is the information retrieval process for recommendation purposes. In this sense, the declarative language supported by this kind of models and the optimized storage facilitates a fast retrieval of the nodes and their relationships. To show the expressivity of the query language, empowered by this graph-based technology, let us to propose an example where we want to get a recommendation based on the relationships between two users. Let us assume that we want to make a recommendation to a user that has a disease (i.e. an allergy). Also, let us assume that people who was born in the same town share eating habits. With all this information we are going to search for two users that share a disease, who were born in the same hometown and who have a tie strength (a friendship value) calculated between them, and get a recipe that was rated by this sec-

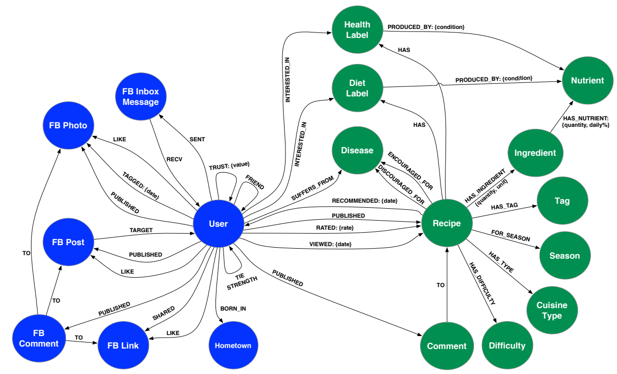

Figure 3. Integration graph model

ond user. Then, in the same query, we apply a filter to get only users with a high tie strength (more than 4.0 in this example). We also filter all the recipes that were already rated by the target user of the recommendation, in order to not recommend a recipe that the user already knows. Finally, we also filter all the recipes that are discouraged for any of the diseases that the target user has (not only the disease shared with the other user), because we do not want to recommend a recipe that may be dangerous for our user.

This query example is shown in Listing 1. Having this information represented in a unique graph empowers us with the ability to create recommendations with relatively simple queries, as shown in this example.

Note that to perform such a query, our knowledge model must accurately store links between recipes and those allergies or food intolerances that produce their ingredients. Therefore, we need to classify each ingredient of each recipe as appropriate or not for each allergy or intolerance that we want to take into account in our recommendation process. The list of allergies and intolerances that we have considered is the one proposed by the 1169/2011 EU regulation, which are:

- Cereals containing gluten
- Crustaceans and shellfish based products
- Eggs and egg-based products
- Fish and fish-based products
- Peanuts and peanut-based products
- Soybeans and soy products
- Milk and products thereof (including lactose)
- Nuts and derivatives
- Celery and celery-based products
- Mustard and mustard-based products
- Sesame seeds and sesame seeds based products
- Sulphur dioxide and sulfites at concentrations above 10 mg/l or 10 mg/Kg
- Lupin and lupin-based products

Listing 1. Example Query

```
MATCH (user1:User)-[tie:TIE_STRENGTH]-(user2:User),
      (user1)-[:HAS_DISEASE]->(diseases:Disease)
      (user1)-[:HAS_DISEASE]->(:Disease)<-[:HAS_DISEASE]-(user2),
      (user1)-[:BORN_IN]->(:Hometown)<-[:BORN_IN]-(user2),
      (user2)-[rated:RATED]->(recipe:Recipe)
WHERE tie > 4.0 AND
      NOT (user1)-[:RATED]->(recipe) AND
      NOT (recipe)-[:DISCOURAGED_FOR]->(diseases)
RETURN recipe
ORDER BY rated.rate DESC LIMIT 1
```

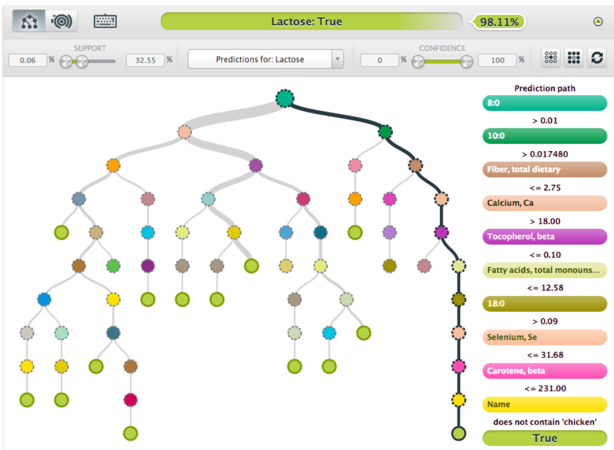

Figure 4. BigML model to detect lactose in an ingredient

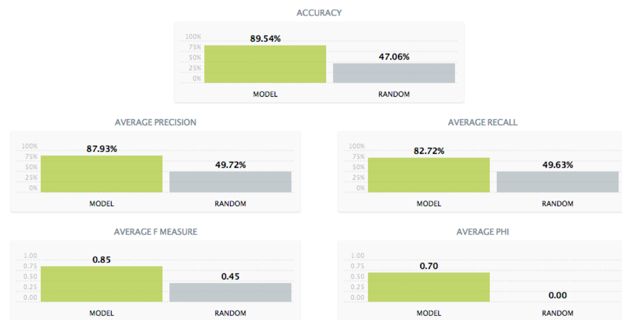

Figure 5. Classification results obtained to predict the appearance of lactose in an ingredient

- Mollusks and mollusks-based products

Furthermore, we have created a classifier to detect the presence of alcohol and another for the presence of fructose (ones of the most common food allergies). Most of our classified ingredients appear in the National Nutrient Database for Standard Reference (USDA), which provides us information about their nutrients that we can use to detect the allergies and food intolerances that they produce. However, there are Spanish specific ingredients that do not appear in this database, and among the data available for each ingredient there is not always enough information to know whether they are appropriate or not for an intolerance or allergy. For example, in the case of lactose, only 1530 out of the 8000 ingredients of the USDA database have information about this nutrient. Therefore, to classify the remaining ingredients a machine learning algorithm based on decision trees was used to determine if an ingredient contains a nutrient or not (lactose, for instance). This process was repeated for each intolerance or allergy that we wanted to take into account. As input to the decision tree we used the set of nutrients labelled in the USDA entry for the ingredient, the name of the ingredient and the Food Group. Then, we used the *BigML*[7] technology to train a model for every food intolerance and allergy, using 80% of the data for training and 20% for test. Figure 4 shows the model generated to detect lactose in the ingredients.

[7] `https://bigml.com`

Figure 5 shows the classification results obtained to predict the appearance of lactose in an ingredient. Moreover, the confusion matrices of figure 6 show false positives (in orange) and false negatives (in red) resulting in the classification.

The results obtained are promising, getting between 89% and 99% accuracy in classification (these variations depend both on the initial data set size for training and on the relationship between the different nutrients of an ingredient and the presence of the specific nutrient to be detected). However, we are continuously improving our model, since allergies detection and classification must be treated with great rigor because of its serious implications on the health of users.

4. RELATED WORK

In addition to the drawbacks of traditional recommender systems, pointed out in section 1, online recommender systems suffer from problems inherent to their use in complex social networks, where the number of users and/or items to recommend can be very high. In the case of collaborative filtering, for instance, the process for comparing two users with the aim of extracting their similarity requires that they have qualified the same objects, which can be unrealistic in large social networks. Another major weakness of online recommender systems is their trustworthiness. In an open network with a large number of users is impossible to ensure that all views expressed are true opinions of users and there

ACTUAL VS. PREDICTED	False	True	ACTUAL	RECALL	F	Phi
False	222	9	231	96.10%	0.93	0.70
True	23	52	75	69.33%	0.76	0.70
PREDICTED	245	61	306	82.72% AVG. RECALL	0.85 AVG. F	0.70 AVG. Phi
PRECISION	90.61%	85.25%	87.93% AVG. PRECISION	89.54% AVG. ACCURACY		

Figure 6. False positives and negatives obtained when predicting the appearance of lactose in an ingredient

is no tampering with the resulting recommendations. In order to overcome these problems, it is necessary to embed a social layer in current recommender approaches, taking into account aspects such as the generation of arguments that support recommendations, reputation and trust. Therefore, there are a number of open challenges for the development of a new generation of recommender systems [12], such as *exposing underlying assumptions* behind recommendations, *approaching trust and trustworthiness* from the perspective of backing recommendations and *providing rationally compelling arguments* for recommendations. Our work involves a contribution in these areas, presenting a knowledge model that a persuasive social recommendation system can use to generate recommendations and arguments to support them.

Currently, social networks are the substrate where "knowledge" is placed. The use of graph-based models to store the knowledge used in social recommender systems is a new research trend with many challenging issues, such as the study of the complex and dynamic relationships that exist among data in order to generate new knowledge. Therefore, this type of databases facilitate the understanding and analysis of the vast amount of data conveyed on social networks. Previous work reported in the literature has demonstrated that recommender systems are improved when their recommendations are enhanced with auxiliary information, such as demographic, domain-specific or contextual information [17]. Also, recent work has highlighted the benefits of using social links between users in recommender systems [19, 18] and of representing these friendship links by means of graph-based models [20].

These proposals support our hypothesis of people trusting recommendations more when the recommender system can use auxiliary information to enhance recommendations (to generate arguments to support its recommendations in our case) and when the recommendation comes from an acquaintance. However, we make a step forward in populating with and extracting social and domain-specific knowledge from the graph-model. Specifically: we create and remove links dynamically as relations among users and/or items appear/disappear; we do not only consider the existence of links between users and/or items, but also enrich these links with semantics (e.g. a computed value that represents the meaning and importance of this friendship relation [15] for friendship links or a trust value that represents the confidence as recommender that a user has in another user); and we use technological standards, as Neo4J[8], to develop our knowledge models, which will allow the widespread use of our technology.

[8]www.neo4j.org

Moreover, over the last years we can find in the literature some recipe recommendation systems [23, 25, 22, 24], which illustrate the increasing demand for this type of systems, specially when planning special meals that must observe dietary restrictions and group meals. However these proposals are not conceived as online recipe recommender systems and do not follow a social recommendation approach [21].

5. CONCLUSIONS

This work presents a graph-based knowledge model for persuasive social recommendation. The system is embedded in the web application *receteame.com* that recommends recipes that fit the preferences and dietary restrictions of its users, currently under development. For the implementation of this system, we have used a graph database which is highly scalable and close to the natural structure of the recipe data and users' social data that we work with. This approach is highly flexible and scalable, and the expected overload in the model due to the high dynamism of recipes and social information on the web (which produces a quick increase on the number of nodes) is considered as negligible comparing with more traditional approaches. In addition, the declarative language supported by our model and the optimized storage facilitate a fast storage and retrieval of the nodes and their relationships in the recommendation process. As current work, we are working on making the recommendation process faster and accurate and on designing an argumentation framework to generate arguments to justify the recommendations. We will also gradually manage more comprehensive dietary restrictions to take into account new diseases and intolerances in the recommendation process. Furthermore, *receteame.com* will be able to recommend full menus in the future, such as a weekly menu for a family that fits the preferences of all members, or a menu for a dinner with friends where the guest can cook something with the confidence that it will like to all guests and fit their dietary restrictions.

Acknowledgment

This work was partially supported by the project MINECO/FEDER TIN2012-36586-C03-01 of the Spanish government and by the Spanish Ministry of Education, Culture and Sports under the Program for R&D Valorisation and Joint Resources VLC/CAMPUS, as part of the Campus of International Excellence Program (Ref. SP20140788).

6. REFERENCES

[1] Desel, J., Pernici, B., Weske, M. Mining Social Networks: Uncovering Interaction Patterns in

Business Processes. *Business Process Management*, Berlin, vol. 3080, pp.244-260 (2004)

[2] Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. on KDE 17(6) (2005) 734–749

[3] X. Zhou, Y. Xu, Y. Li, A. Josang, and C. Cox, "The state-of-the-art in personalized recommender systems for social networking," *Artificial Intelligence Review*, vol. 37, no. 2, pp. 119–132, 2012.

[4] Ehrig M., "Ontology Alignment: Bridging the Semantic Gap," *Springer*, 2007.

[5] Euzenat, J. and Shvaiko P. , "Ontology matching," *Springer, Heidelberg (DE)*, 2007.

[6] Bleiholder, J., Naumann, F. , "Data Fusion," *ACM Computing Surveys*, 41(1):1-41, 2008.

[7] Halpin, H., Thomson, H. , "Special Issue on Identify, Reference and the Web," *Int. Journal on Semantic Web and Information Systems*, 4(2):1-72, 2008.

[8] I. Robinson, J.Webber, and E. Eifrem, *Graph Databases*. O'Reilly, 2013.

[9] M. Pazzani and D. Billsus, *Content-Based Recommendation Systems*, ser. LNCS. Springer-Verlag, 2007, vol. 4321, pp. 325–341.

[10] J. Schafer, D. Frankowski, J. Herlocker, and S. Sen, *Collaborative Filtering Recommender Systems*, ser. LNCS. Springer, 2007, v. 4321, pp. 291–324.

[11] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[12] C. Chesñevar, A. Maguitman, and M. González, *Empowering Recommendation Technologies Through Argumentation*. Springer, 2009, pp. 403–422.

[13] G. Linden, J. Hong, M. Stonebraker, and M. Guzdial:, "Recommendation Algorithms, Online Privacy and More," *Comm. of the ACM*, vol. 52, no. 5, 2009.

[14] Khare, Rohit and Çelik, Tantek, "Microformats: a pragmatic path to the semantic web" in *15th international conference on World Wide Web*, ACM, 2006, pp. 865–866.

[15] R. Fogués, J.M. Such, A. Espinosa, and A. Garcia-Fornes. BFF: A tool for eliciting tie strength and user communities in social networking services. *Information Systems Frontiers*, 16(2), 225-237, 2014.

[16] S. Heras, V. Botti, and V. Julián. Argument-based agreements in agent societies. *Neurocomputing*, doi:10.1016/j.neucom.2011.02.022, 2011.

[17] S. Berkovsky, T. Kuflik, and F. Ricci. Mediation of user models for enhanced personalization in recommender systems. In *User Modeling and User-Adapted Interaction*, 18(3), 245-286, 2008.

[18] I. Cantador, I. Konstas, and J.M. Jose. Categorising social tags to improve folksonomy-based recommendations. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(1), 1-15, 2011.

[19] I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel. Social media recommendation based on people and tags. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp. 194-201, ACM, 2010.

[20] A. Tiroshi, S. Berkovsky, M.A. Kaafar, D. Vallet, and T. Kuflik. Graph-Based Recommendations: Make the Most Out of Social Data. In *User Modeling, Adaptation, and Personalization*, pp. 447-458, Springer International Publishing, 2014.

[21] J.J. Pazos, A. Fernández, R.P. Díaz. Recommender Systems for the Social Web, Springer Berlin Heidelberg, 2012.

[22] M. Ueda, M. Takahata, and S. Nakajima. UserâĂŹs food preference extraction for personalized cooking recipe recommendation. *Semantic Personalized Information Management: Retrieval and Recommendation*, SPIM, pp. 98-105 2011.

[23] I. Mazzotta, F. De Rosis, and V. Carofiglio. Portia: A user-adapted persuasion system in the healthy-eating domain. *Intelligent Systems*, IEEE, 22(6), 42-51, 2007.

[24] A. Said, and A. Bellogín. You are what you eat! tracking health through recipe interactions. In *Proceedings of the 6th Workshop on Recommender Systems and the Social Web, RSWeb*, 2014.

[25] J. Freyne, and S. Berkovsky. Intelligent food planning: personalized recipe recommendation. In *Proceedings of the 15th international conference on Intelligent user interfaces*. pp. 321-324, ACM, 2010.