Restaurant Recommendation Algorithm

collaborative filtering - by calculating the similarity between one user’s preferences and the preferences of other individuals.

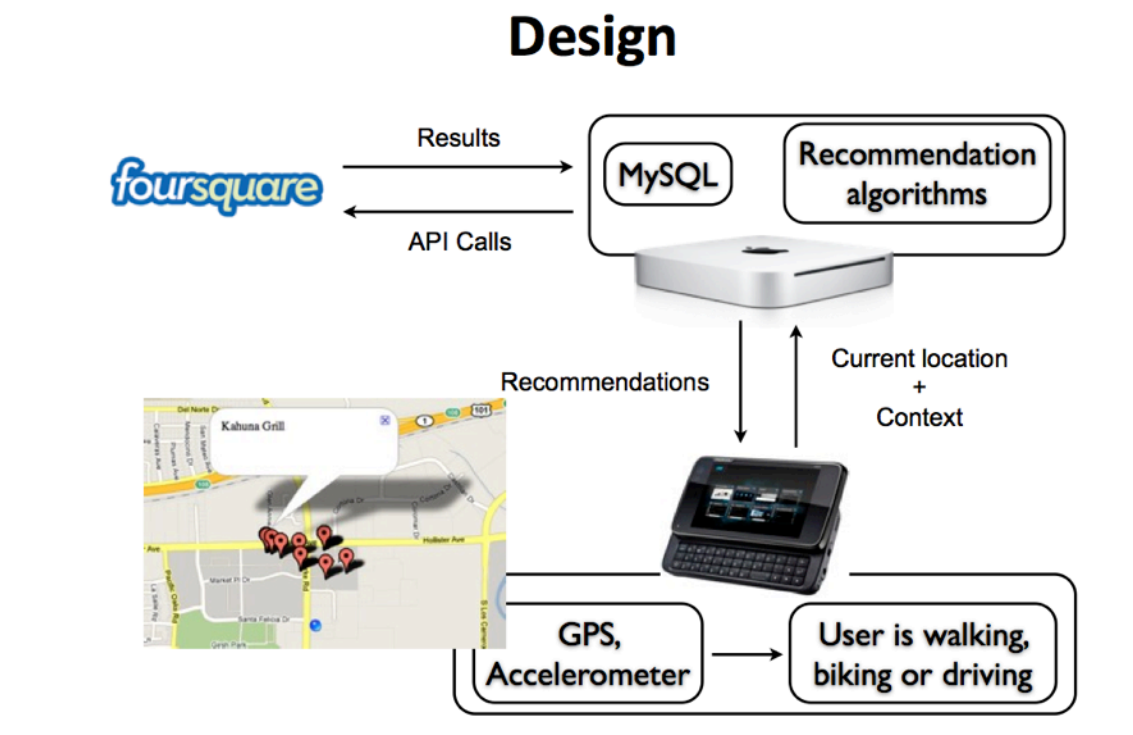
In this study, the items denoting the user preferences are the restaurants visited by the user along with their associated information: tags and assigned category.

content based filtering - hold a set of items denoting the user’s preferences.

Cold Start Problem: The task of the algorithm is to classify an unseen item (an option the user has not expressed any opinion about), as something relevant or irrelevant for the user. The unseen items to be classified are places “near” the user that they have never before visited. Because the space of places to analyze is immensely large, our system utilizes the user’s capability constraints, preferences and mood, to delimit the search.

given the current location of the user, the foursquare API is utilized to return all of the places within a certain radius to where the user is.

Foursquare returns only places within 400 meters radius from the user's current location. But our place retrieval algorithm requires a radius, which varies according to person's mode of transportation. An algorithm for automatically increasing the considered radius had therefore to be implemented. The functionality of this

Figure 6. Overview of I’m feeling LoCo system operation. Through foursquare API calls, a server is constantly updating an individual’s user model, and storing in a database this information. On the other hand, a mobile application is continuously detecting the user’s transportation mode as well as offering an interface through which the user can request place recommendations. When the user queries the system for a recommendation, the mobile phone submits to the server, the user’s location and contextual information: their transportation mode. The server inputs this data, along with the generated user model to the recommendation algorithm, that utilizes these features to decide what places are the most suitable to be recommended. The server returns to the mobile phone a list of the best K places for the user to visit. The mobile phone then displays these places on a virtual map.

Clustering based on Context-aware data – Location, Time of Data, Categories Tags,

<http://recommender.no/info/hybrid-recommender-systems/>

<http://recommender.no/info/content-based-filtering-recommender-systems/>

Recommender systems attempt to profile user preferences over items and model relation between users and items.

Yep, I want that information. I and do you see how you can click on her reviews and it'll show you every single review she's ever made? I want to take her friends list, strip every review ever written by everyone of her friends, and let her search through that list for anything she wants and display the top results. So let's say 40 of her friends have eaten at Sushi places in Seattle, WA. 20 out of those 40 friends ate at the "Good Sushi" restaurant. 20 of her friends ate at "Bad Sushi Restaurant". When she searches for "Sushi" in "Seattle, WA" she'll only get Good sushi and Bad Sushi.

Just found my own answer through a rather round about process.

The Yelp API doesn't allow for any access to your friends because said data is attached through Facebook. Using the Facebook Graph API and Login API to allow for Yelp users to Login using their Facebook profile and then use the Graph API to pull all friends associated with that user account. Once that information is gathered you can use the Yelp API with reviews.user and match to the friends data gathered from the Graph API to pull the appropriate business and reviews.

Hope this helps someone else.

<http://www.yelp.com/user_details_friends?userid=inz8izVu0oSGeJJdY32zYA>

**In the proposed approach, firstly users are classified by applying clustering technique on ratings data. Subsequently, rule-based reasoning (RBR) and case-based reasoning (CBR) are employed separately to choose classes (neighborhoods) of an active user and then collaborative filtering (CF) is applied on these neighborhoods to produce recommendation lists. These two techniques are respectively called RCF (combination of RBR and CF) and CCF (combination of CBR and CF). The proposed weighted hybrid recommender system (WRCCF) combines RCF and CCF schemes. Experimental results reveal that the proposed WRCCF consistently outperforms Pearson CF (PCF), RCF, and CCF in terms of prediction and classification accuracy.**

Moreover, several data mining or machine learning algorithms including Bayesian belief nets (*BNs*) [6], clustering techniques [7], and latent semantic analysis [8] have been employed for building a model in CF techniques.

The system takes the benefit of k-means clustering algorithm to classify users into different clusters.

These clusters are used as a basis to select a neighborhood (nbd) of an active user.

similarity between two users demographic information such as age, gender, occupation and location respectively Typically, CF explores similar users by recognizing commonalities between the user and his neighbors on the basis of their previously expressed preferences and then accordingly suggests new items or products based on inter-users comparison

In this paper, we propose a method which combines two model-based CF algorithms by a weighted scheme to improve the quality of recommendation as well as coverage of CF.

Steps:

1. **classified by applying clustering**
2. **rule-based reasoning (RBR) and case-based reasoning (CBR) are employed**

Sparsity, scalability, neighbor transitivity, and accuracy are the main problems of CF.

To handle the problems of CF, other recommendation techniques such as *content-based filtering* [1], [5] and *knowledge-based filtering* [1], [4] have been combined with CF by using hybrid algorithms.

Does not depend on any feedback system.

First, a lexicon that is constructed to facilitate the analysis of a specific domain such as a newspaper, product review or movie review, and the lyrics of songs or a general-purpose lexicon are not sufficient for the sentiment analysis of a restaurant review. Therefore, in this thesis a lexicon that includes the sentiment words associated with a restaurant review is constructed by analyzing the restaurant reviews.

Secondly, the original lexicon was built with a unigrams pattern, but in such a case it is difficult to find a polarity information of the sentiment word with a bigram pattern that has a negative word or an adverb that intensifies the meaning of the sentiment. Thus the unigrams pattern as well as the bigrams pattern, which includes negative words and intensive adverbs, were included in the lexicon in this thesis. Additionally, the experiment was conducted by considering the unigrams + bigrams pattern that combines the unigrams and the bigrams for the purpose of performance comparison.

**SentiWordNet**.

a named entity tagger and found a sentiment word by using WordNet. the General Inquirer Lexicon and the WordNet database

a lexicon is constructed by expanding the synonyms and antonyms of each seed word.

a supervised learning algorithm such as Naïve Bayes or SVM was used for classification.

Among the sentiment word candidates, a method that selects a word with an occurrence frequency in the overall document higher than the threshold as the final sentiment word was proposed.

Thirdly, a sentiment word that is made up of two words such as ‘‘not delicious’’ cannot be found in SentiWordNet. This is because SentiWordNet only expresses the sentiment score for one word.

not delicious’

The process of its construction is as follows:

(1) Reviews associated with a restaurant were collected by developing a wrapper-based Web crawler.

(2) A part of speech was tagged by eliminating the unnecessary characters from the collected reviews, such as a number, symbol and HTML tag, and by using a POS tagger.

(3) The unigrams, bigrams and sentiment patterns were found by a manual process and were added to the lexicon by determining the positivity and negativity of the applicable pattern.

The senti-lexicon L is made up of a set as shown in (1).

L ={TYPE; TARGET; PATTERN; POLARITY}

The values that correspond to TYPE are unigrams or bigrams. The unigrams in TYPE refer to a pattern made up of one word. The examples of such pattern are ‘‘cheap’’, ‘‘delicious’’ and ‘‘good’’.

The bigrams is a pattern type made up of two words. The examples of such pattern include ‘‘very delicious’’, where the first word is an accentuating adverb (Z) that stresses the meaning of ‘‘delicious’’; and ‘‘not delicious’’, where a negative word is attached and gives it the opposite meaning. A TARGET represents a subject that expresses an emotion. This represents an evaluative attribute such as the food’s taste, facility, mood and price that can be felt when visiting a restaurant. When TARGET is not shown explicitly, it is expressed as common. A PATTERN expresses an emotive word pattern. A part of speech is also expressed at this time. It is made up of one word (e.g. Nos. 1 and 2) or two words (e.g. Nos. 3 and 4) according to a TYPE. The parts of speech of words that are considered in this thesis are adjectives, verbs and nouns. Additionally, even with a same part of speech, a parameter whereby the extraction should be made by each part of speech was selected. Because most of the words paired with an adjective as the part of speech express emotions, the decision was made to extract all adjective words. It was decided that words in verbs and nouns that express emotions would be extracted. For example, ‘‘recommend’’ is a verb, but a word such as ‘‘recommendation’’ or ‘‘kindness’’ is expressed as a verb or an adjective in the form of a noun in a noun. Additionally, a word such as ‘‘eat’’, which carries little emotive meaning, is eliminated and a pattern that expresses an emotion through an adverbial phrase such as ‘‘eat deliciously’’ is built into a lexicon. A POLARITY is expressed as positive or negative. Table 2 shows an example of a lexicon that is constructed in this thesis.