Reducing the Cold-Start Problem in Content Recommendation Through Opinion Classification

Damien Poirier, Françoise Fessant, Isabelle Tellier

Reducing the Cold-Start Problem in Content Recommendation Through Opinion Classification

Damien Poirier, Françoise Fessant
Orange Labs
2 avenue Pierre Marzin
22300 Lannion, FRANCE
firstname.lastname@orange-ftgroup.com

Isabelle Tellier
LIFO, Université d'Orléans
Rue Léonard de Vinci
45000 Orléans, FRANCE
firstname.lastname@univ-orleans.fr

***Abstract*—Like search engines, recommender systems have become a tool that cannot be ignored by websites with a large selection of products, music, news or simply webpages links. The performance of this kind of system depends on a large amount of information. At the same time, the amount of information on the Web is continuously growing, especially due to increased User Generated Content since the apparition of Web 2.0. In this paper, we propose a method that exploits blog textual data in order to supply a recommender system. The method we propose has two steps. First, subjective texts are labelled according to their expressed opinion in order to build a user-item-rating matrix. Second, this matrix is used to establish recommendations thanks to a collaborative filtering technique.**

***Keywords*-Opinion classification; User Generated Content; Recommender systems; Collaborative filtering**

I. Introduction

The aim of recommender systems is to help users to find items that they might appreciate from large web-based catalogues. The items can be of any type, such as movies, music, books, webpages, etc... Recommender systems help users to find such items of interest based on some information about their historical preferences. There exist several approaches to build a recommender system [1]: collaborative filtering, consisting of using users' tastes to build recommendations, content-based filtering, consisting of using descriptors to categorize items and users and finally hybrid filtering. In all cases, the main challenge to building an efficient recommender to face is to collect enough data to "initialize" the recommendation process. The idea developped in this work is that it is possible to collect informative data from texts found on the Web 2.0. To fill the gap between opinion texts and recommendation, we need to refer to the growing field of opinion mining. Many works have been dedicated recently to the automatic classification of opinion texts according to the polarity (positive or negative) of the opinion they express [2]. It is now time to evaluate if the polarity inferred from texts by opinion mining technics is precise enough to "feed" a recommender system. Especially a collaborative filtering one and to answer some questions: does textual data coming from blogs/forums contains enough information to achieve recommendation? Is a good opinion classification really useful to produce good recommendations? Is this approach competitive with content-based filtering?

To evaluate our approach, we choose the well-studied domain of recommendation about movies. The complete processing chain we propose is described in section 2. Section 3 is dedicated to the description of our corpora. Then, our first experimental results for classification experiments and for text-based recommandations are given section 4. We discuss the impact of the number of distinct possible ratings chosen to classify the texts on the final recommandation results. We show that the ratings extracted from opinion texts provide very satisfying results for a collaborative filtering system.

II. Processing chain

The processing chain we propose can be separated into two main tasks. The first one concerns the analysis of textual data. This task consists in obtaining a user-item-rating matrix. The choice of texts is then important. Each text has to contain an opinion about an identified item and the author of the text has to be known too. Once the acquisition of data done, different treatments can be applied on the corpus such as removal of HTML tags or/and stop words, NLP treatments or a subjectivity classification task. The choice of the useful treatments really depends on the crawled data. Once a large number of user-item-review triplets are stocked, the opinion classification task can be applied in order to infer a rating for each review, i.e. in order to create user-item-rating triplets. The majority of studies on opinion classification have chosen a two-classe classification (like, dislike) or a three-classe classification (like, neutral, dislike). After the building of the user-item-rating matrix, collaborative filtering can be applied and recommendations can be done. This is the second task. For this step, the recommender system builds a similarity matrix of users or items thanks to the user-item-rating matrix and a recommendation can be computed for a user and an item.

III. EXPERIMENTS

We decided to test the processing chain on movie recommendation. It is a very common application in the field of recommender systems and several benchmarks are available. In this part, we start with the description of the corpora used. The description of the protocol is then given, finally, we present the experimental results.

A. Corpora

We design three corpora for our experiments. The third is a corpus of ratings while the first and the second are textual corpora.

- The corpus of ratings, called *Recommendation Corpus*, is used to evaluate the quality of recommendations, which is the last test of the experiments. It is a well-known corpus in the field of recommender systems. It is the set of logs made available by Netflix[1] for the Netflix Prize, a challenge created in order to improve the collaborative filtering algorithms for an online DVD-rental service. It contains ratings put on Netflix by 400,000 users, that is approximately 100,000,000 user-item-rating triples.
- The second corpus comes from the website Flixster[2]. We call it *Test Classification Corpus*. Flixster is a community space where many movie fans meet and share their tastes and preferences about cinema. On this website, they can create a personnal page where they can post, among other things, ratings and reviews about films. Reviews about films are connected to a user, a film and a rating given by the author. The rate summarizes the opinion expressed in the review. This corpus, which is dedicated to the test of the opinion classification task, is composed of user-item-rate-review quadruples. It contains approximately 3,300,000 reviews written by almost 100,000 users speaking about 10,500 movies. All of these 10,500 movies are also present in the *Recommendation Corpus*.
- The last corpus, called *Learning Classification Corpus* also comes from Flixster but it has no intersection with the *Test Classification Corpus*. It contains approximately 175,000 user-item-rate-review quadruples. It is used during the learning step of the opinion classification task. Thanks to this set of examples, the learning tool can build a model which will serve to classify the texts of the *Test Classification Corpus*. The texts have been selected so that the ten original classes (present in Flixster) are balanced.

The average size of texts contained in *Test* and *Learning Classification Corpora* is about 14 words which is few and unusual for a text classification task. An other unusual aspect concerns writing style, wich can be very variable depending on the author. Table I contains examples of reviews extracted from the textual corpora.

Rate	Review
4	It is a funny movie, but its not as funny as scary movie!
5	Soooo awsome i loved it..Tom Cruise did awsome in this movie..
4.5	best film ever except the orginal is better if you were to wach any trilogy i would say to you watch this set
2	Very long and just leaves you hanging.
3	Boring. Shame too because I Love Sarah Jessica Parker
3.5	Very funny. Typical and quite predictable story but pretty hilarious. I had some good laughs. :)

Table I
EXAMPLES OF REVIEWS EXTRACTED FROM *Learning* AND *Test Classification Corpora* (RATINGS ARE INCLUDED BETWEEN 0.5 AND 5, 0.5 MEANING REALLY BAD AND 5 MEANING REALLY GOOD)

The preprocessing applied to these two textual corpora are few and light. The only treatments done consists of putting every letter in lowercase and deleting uncommon words. Ponctuation, word stretching, repetitions, misspellings, etc. are kept as they are. We tried to apply more advanced treatments like removing stop words or applying NLP treatments but each of them brought a loss of information for the classification task. For the experiments, each text is represented by a vector of word frequencies. The length of the vector is about twenty thousand variables (it was longer than 150,000 before preprocessings).

B. First step: Opinion Classification (from Texts to Ratings)

The aim of this first step is to infer ratings from user reviews in order to obtain user-item-rating triplets instead of user-item-review triplets. The ratings we want to obtain measure the opinions expressed in the reviews, that's why we do opinion classification. After different tests, we choose to use machine learning tools which are more efficient than linguistic techniques for this task, and they are fully automatic. We use a Selective Naive Bayes (SNB) method. We rapidly describe the tool used before presenting the results.

1) Presentation of the classification tool: We use the **KHIOPS**[3] tool, which is a data mining tool allowing to automatically build successfull classification models, on very large volumetry [3]. In the first step consisting in preparing data, explicative variables are individually evaluated by means of an optimal discretization method in the numerical case or by means of optimal grouping values in the categorial case. In the modeling step, a classification model is built by effectively averaging a lot of selective variables based models. By only keeping most important variables during the learning construction of the model, results in test are really improved compare to classic Naive Bayes classifier.

[1] www.netflix.com
[2] www.flixster.com
[3] Downloadable shareware on http://www.khiops.com/

2) Results: Classification was done on two classes : positive reviews and negative reviews. The F_{score} calculated on the obtained confusion matrix (table II) is 0.71.

		True Classes	
		NEG	POS
Predicted Classes	NEG	**79.1**	36.8
	POS	20.9	**63.2**

Table II
RESULTS OF OPINION CLASSICATION (IN PERCENT)

This result is not really a good result compared with other works on this field. But, as it has been said in section III-A, the copora used have some particularities, like the size or the writing style, which are not helpfull for classification. Let us also specify that classes were not attributed by experts but by Web users who are not always very rigorous. The observation of the misclassified reviews shows that a lot of ratings given by the authors do not correspond to the opinion express by the text. And finally, some comments (table III) can be really difficult to classify, sometimes even for a human, because the information contained by the review is insufficient.

Rate	Review
1.5	The only good thing is the fight!
1.5	LOVE IT
3	I like the original. Have not seen the new one
0.5	very funny, without trying to be
5	it sucks how Chris Brown dies in the first scene!
5	I don't have anything to say thank you
5	Bad Stupid and Vile
4.5	Don't like the end...

Table III
EXAMPLES OF MISCLASSIFIED REVIEWS

As said earlier, KHIOPS allows an interpretation of the results. The table IV contains the 30 (out of 635) most informative variables found by KHIOPS to classify texts according to the opinon. Majority of them are opinion words but we can observe the relative importance of certain punctuations. The presence of words like "movie", "but" or "was" is more surprising. Their presence is probably because the texts expressing positive opinion are longer than the negative ones.

!	boring	awesome	movie	brilliant	waste
love	great	stupid	hilarious	awsome	but
loved	amazing	t	sucked	didn	crap
not	bad	worst	excellent	terrible	alright
ok	best	.	was	'	wasn

Table IV
THE MOST INFORMATIVE VARIABLES FOR OPINION CLASSIFICATION ACCORDING TO KHIOPS

C. Second step (Collaborative Filtering): from Ratings to Recommendations

This second step consists of doing recommendations thanks to the results obtained with the first step. As a matter of fact, these results build a user-item-rating matrix which is exploitable by a recommender system based on a collaborative filtering technique. The recommender system used for this study is based on an item-based approach. Before seeing the results of our experiments and discussing them, we present the recommender system used.

1) The recommender system used: It can work with collaborative filtering or content-based approaches. The learning step consists of building similarity tables of items. We use the Pearson correlation in the case of collaborative filtering. In the case of content-based filtering, we use the Jaccard similarity. The second step involves predicting recommendations based on the similarity table. The function used to obtain the prediction of the rating that could be given by user u on an item i is the following:

$$p_{ui} = \frac{\sum_{\{j \in S_u\}} sim(i,j) \times (r_{uj})}{\sum_{\{j \in S_u\}} sim(i,j)}$$

With the set S_u of ratings given by the user u.

In order to compare the different results, we compute the Root Mean Squared Error (RMSE), which is the most measure often used in the recommender systems field. The RMSE measures the error rate between real and predicted ratings. To give an idea of the RMSE, the Netflix challenge was to reach a RMSE of 0.85 with their data. This purpose required huge computations and efforts to be achieved. In order to evaluate the recommender system, the *Recommendation Corpus* is cut in two parts. All tests are done on the same part and the other part is used to evaluate performances of the tool in the collaborative filtering case. We consider this result as the best as possible with this tool. The learning step of the content-based case is done with data extracted from IMDB[4]. The attributes used to compute similarities are descriptors like genre, actors, director, etc. and tags which are found in aboundance on this website. RMSE obtained are presented in table V.

Collaboratif filtering (data from Netflix)	Content-based filtering (data from IMDB)
0.862	0.968

Table V
EVALUATION OF THE RECOMMENDER SYSTEM ON TWO KNOWN DATA SETS

2) Experiments: We tested the recommender system with the results obtained from the opinion classification task. In order to better understand the impact of the opinion classification task, we also performed the experiment with

[4]www.imdb.com

the real ratings, given by the authors of texts, and with random ratings. RMSE obtained are presented in table VI.

With real ratings	With predicted ratings	With random ratings
0.897	0.898	0.989

Table VI
RMSE OBTAINED WITH DATA PROVIDING FROM COMMUNITY WEBSITE

3) Discussion: Results obtained with data provided from Flixster are really encouraging, especially the result obtained with ratings predicted during the opinion classification which is as good as the one obtained with real ratings. Furthermore, the RMSE provided is significantly better than that obtained with random ratings, which means the classification step has some importance in the quality of results obtained.

IV. BROADER DISCUSSION

The first purpose of this work was to verify if textual data providing from blogs/forums contains enough information to make recommendations. The last results show that the recommendations done with external data are reasonably good but not as good as the ones obtained by learning on Netflix corpus. We think this difference is mainly due to the quantity of information present in each of the corpora. Indeed, it should be noted that, in Netflix data, each user rated an average of about 200 distinct DVD whereas, in the Flixter website, each user commented and rated an average of only 30 distinct films. Moreover, there exist other differences between the two corpora. The ratings assigned to a DVD in Netflix may rely not only on the interest of the film it contains: some properties such as its technical quality (image, sound), the interest of available bonus, etc. can also be taken into account by the users. On the contrary, only films are commented by Flixter members.

Our second goal was to verify if recommendation requires a really good opinion classification. For that, we show that a classification with an F_{score} equal to 0.71 is sufficient to obtain recommendations as good as using real ratings.

Our last observations on the results show that the RMSE obtained with external data is much better than RMSE obtained with a good content-based filtering. Indeed, IMDB can be regarded as the most complete data base from the web about cinema. The descriptors choosen are numerous. They contain actors, directors, producers, genres, country, year, writers, company, language, and many keywords. We can logically assume that even with a smaller quantity of sujective texts provided from blogs or forums, recommendations could be better than recommendations made by a recommender system based on content-based filtering. Table VII summarizes the important results obtained.

	Collaborative Filtering		Item-based Filtering
	Data from Netflix	Data from Flixster	Data from IMDB
RMSE	0.862	0.898	0.968

Table VII
IMPORTANT RESULTS OBTAINED IN RECOMMENDATION

V. CONCLUSION AND PROSPECTS

We propose a method to perform recommendations from unstructured textual data in order to overcome the cold start problem. The cold start happens when the number of users registered in the service is small. Even if the idea of combining a classification step and a recommendation one is not new, this work is the first one, to our knowledge, to achieve the whole process at a large scale. It opens new perspectives for both domains.

For the classification field, it validates the common intuition that opinion mining could serve as a pre-treatment for many other tasks. Recommendation is a very hot topic, and every possible way to fill a matrix of ratings without asking users to explicitely providing them is valuable. Our experiments also show that, at least in our case, an easy and fast to learn binary classifier is sufficient to obtain good recommendations. Nevertheless, it is still not clear when a classifier learned from any corpus can be efficiently used to associate ratings to texts from a completely different one.

For the recommendation field, our experiments first show that it is possible to feed a recommendation system with ratings coming from a completely different website. The results are not as good as with local ratings but still far better than random ones. And, if “foreign ratings” are not available, ther now still remains a possibility to infer them from texts. Ratings are difficult to collect, as they are not spontaneoulsly given by users. On the contrary, blogs, forums, social networks... are quasi-infinite sources of freely available spontaneously written texts. Most of these texts very often carry opinions on subjects easy to identify. So, a large avenue of possible work seems to open.

As prospects, we want to extract more information from texts different from polarity of opinion. This information can be semantic or statistic. With a tool like KHIOPS we can access to different statistics on variables which can be very interesting to structure texts in different ways. We then try to capture new kind of information usefull for a task of recommendation.

REFERENCES

[1] L. Candillier, K. Jack, F. Fessant, and F. Meyer, “State-of-the-art recommender systems,” *IGI Global*, pp. 1–22, 2009.

[2] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” *Found. Trends Inf. Retr.*, vol. 2, no. 1-2, pp. 1–135, 2008.

[3] M. Boullé, “Compression-based averaging of selective naive Bayes classifiers,” *Journal of Machine Learning Research*, vol. 8, pp. 1659–1685, 2007.