Testing Collaborative filltering techniques on Banking Information

Santander Reccomendation Challenge

Pedro Bahamondes
Pontificia Universidad Católica de Chile
pibahamondes@uc.cl

Henry Blair González
Pontificia Universidad Católica de Chile
hblair1@uc.cl

ABSTRACT

On 2016, Santander launched a Kaggle competition with the goal to get better recommendations for their clients. On this document, we compare different methods based on collaborative filtering with respect to the error obtained with each algorithm. We also show how the error changes when the dataset is filtered according to specific categorical features on a smart way.[1]

CCS CONCEPTS

• **Theory of computation → Machine learning theory**; *Database theory*; • **General and reference**;

KEYWORDS

Recommendation Systems, Collaboartive Filtering, Parallel Programming, SVD, SVD++, Slope One, Co Clustering, Non Negative Matrix Factorization, DataSet Filtering

ACM Reference Format:
Pedro Bahamondes and Henry Blair González. 2018. Testing Collaborative filltering techniques on Banking Information: Santander Reccomendation Challenge. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

During year 2016, the Santander Group launched a Kaggle competition in order to provide better product recommendations for their customers [1]. As stated in the contest page [1]:

> "Under their current system, a small number of Santander's customers receive many recommendations while many others rarely see any resulting in an uneven customer experience. In their second competition, Santander is challenging Kagglers to predict which products their existing customers will use in the next month based on their past behavior and that of similar customers."

The contest rewarded a total of USD$60,000 among the three top teams, and was a total success with a total amount of 1,787 teams that submitted their answers, and a total amount of 28,772 submissions. [1]

In this context, the Santander Group provided a huge database with historical data of almost one million (fictional) users, including their personal data, banking data and product consumption at different times. The aim of the competition was to predict future consumption of the their users in order to better anticipate the products that will be bought. [1] Hence, we have a recommendation system problem.

Although the top contestants used a method called XGBoost [2], the aim of this paper is to explore the performance of different classical collaborative filtering methods in two different scenarios. In a first scenario, we divide the database into several subdatasets according to the personal data of the users and then we build one prediction model for each of these subdatasets. In the second scenario, we build a prediction model for the entire dataset. We finally compare both approaches using typical error metrics, that although may be not ideal for the dataset, do reveal an improvement in prediction performance in the first approach with respect to the second one.

This paper is divided as follows. Firstly, we state the objectives of our work. Secondly, we explore the banking dataset that is given for the contest to reveal useful patterns. We then propose the hypothesis, extracted from the dataset exploration, that justifies our experiments. Next, we describe our experiments' methodology and the error metrics used to compare the different collaborative filtering methods. After that, we summarize the conclusions of our experiments. Finally, we explore the limitations of our work and we propose corrections and extensions for future work.

2 RESEARCH OBJECTIVES

Through this work, we aim to:

- Apply a number of collaborative filtering techniques in a database with bank information and compare their performance.
- Use a programming tool called `surprise`, which works as an alternative to the known `pyreclab` tool and describe it qualitatively in comparison to `pyreclab`.
- Compare the performance of the collaborative filtering approaches applied in two different scenarios: after clustering the dataset using the user's personal and banking data and then building one prediction model for each cluster using the consumption information, versus building a single model

[1]The algorithms were implemented in the python 3 library `surprise`

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

on the entire dataset only looking at the consumption information.

This last objective is the main objective of our paper and it also derives in our main contribution: Integrating additional information about the users significantly improves the quality of the recommendations while reducing the computation time and memory resources needed to build the recommendations model.

3 DATASET

The contest dataset has historical personal, banking and consumption data for 956, 645 users, for a total of 13, 647, 309 48-dimensional entries. Each entry has in its first dimension a date, in its second entry an user ID, followed by 46 other dimensions describing the status of the associated user at the given date.

The status of a user includes dimensions associated to their personal information (sex, age, annual income, marital status, origin country, residence country, deceased index, ...), their banking information (customer seniority, novelty index, client type, ...) and product consumption (current account, private account, receipt, ...).

The product consumption is described by a one-hot codification, stating, for each of the 24 dimensions associated to a product, whether the associated product has been bought by the user associated to the entry's ID up to the entry's date. It should be noticed that the dataset was arranged in order to be compatible with the collaborative filtering approaches, since the ones we explored only work with explicit feedback.

In the following, figures 1 and 2, we show the distribution of the database according to their age and their income levels, which are the variables with respect to which we did the clustering in our approach.

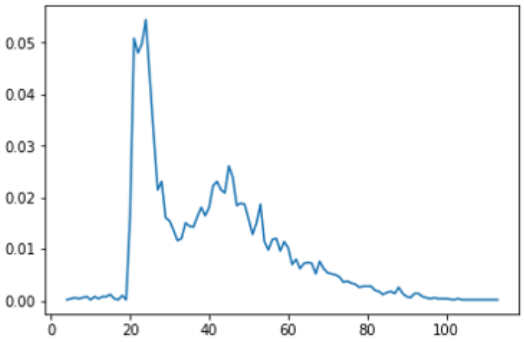

Figure 1: Age distribution of the users in the database.

With respect to age, it is possible to see three main age groups. The first group is a small group composed of people under 18, with very low product consumption. Then there are two main groups in the dataset with different consumption patters, a first group between 18 and 30 years old, and a second group with more than 30 years. It can also be noticed that there is a considerable amount of people over 100 years. In fact, this is due to the presence of deceased people in the database, which are obviously expected not to consume any more products in the future.

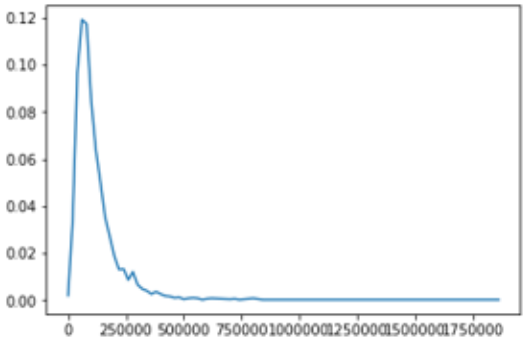

Figure 2: Yearly income distribution of the users in the database.

When analyzing customer behavior with respect to age, people with income greater than 250000ff are the ones that are more prompt to consume products, howeven, as shown by figure 2, most of the database lies in the $0 - 250,000$ff income range.

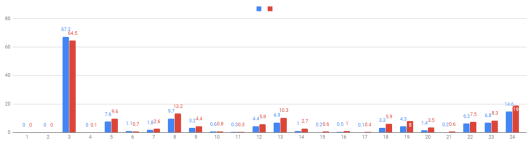

Figure 3: Purchase behavior for each gender (V for male, H for female). Bars show the percent of each gender that has bought each of the 24 available products.

Let us have a look at figure 3. As shown in the figure, males and females consumption behavior differs. For instance, product 3 (current account) is bought by a greater proportion of men, whereas product 8 (particular account) is bought more frequently by a greater proportion of women.

This observations lead to our main hypothesis: Consumption behavior depends on personal and banking information, and not only in previous consumption behavior. Thus, it should be possible to improve the prediction by building models around clusters of users that were themselves created using this pertinent information.

4 EXPERIMENTS

For our experiments, we compared a number of collaborative filtering approaches under two different error metrics. We compared their performance with respect to each other and with themselves in two different scenarios, depending on how the data was partitioned.

4.1 Collaborative Filtering Methods Employed

We employed a total of 5 collaborative filtering techniques: Slope One [6], SVD [**?**], SVD++ [5], CoClustering [**?**] and Non-Negative Matrix Factorization [4] for our experiments. The implementations of these algorithms used the library `surprise` [**?**].

5 ERROR METRICS EMPLOYED

We used two different error metrics, mean root square error (RMSE) and mean average error (MAE), on the vectors of predicted consumption. They are respectively defined from the observed data (without hat) and the predicted data (with hat) by

$$RMSE = \sqrt{\frac{1}{|\hat{R}|} \sum_{\hat{r}_{ui} \in \hat{R}} (r_{ui} - \hat{r}_{ui})^2}$$

$$MAE = \frac{1}{|\hat{R}|} \sum_{\hat{r}_{ui} \in \hat{R}} |r_{ui} - \hat{r}_{ui}|$$

5.1 Experiments Description

As previously mentioned, we ran our experiments with each of the models after converting the data in the datasets in two different scenarios, using only two of the 24 available parameters in the personal/banking information of the users.

In the first scenario, we split our dataset in four according to two possible ranges of income level and two possible ranges of age. For each of these groups, we built a prediction model. The testing phase then would first see in which of these clusters the user was in, to then use the corresponding model in the prediction. We denote these clusters s00, s01, s10 and s11.

In the second scenario, we completely ignored the personal and banking information and built a single model for the entire dataset using only consumption information. In the testing phase, every input would use this model the prediction.

We measured errors according to our metrics for both the post-clustering case (first scenario) and the all-in-one case (second scenario).

5.2 Results

Table 1: Errors in the subdatasets

		SlopeOne	SVD	SVD++	CoClustering	NMF
s00	RMSE	0.1321	0.1321	0.1323	0.2004	0.1341
	MAE	0.042	0.0402	0.0413	0.0524	0.0315
s01	RMSE	0.1551	0.1551	0.1541	0.1736	0.1492
	MAE	0.0580	0.0580	0.0576	0.0457	0.0400
s10	RMSE	0.2374	0.2374	0.2373	0.2376	0.2207
	MAE	0.1252	0.1252	0.1243	0.1027	0.0881
s11	RMSE	0.2452	0.2452	0.2451	0.2593	0.2278
	MAE	0.1343	0.1343	0.1343	0.1259	0.0973

It can be noticed that in general NMF reached the best overall performance. Also, and maybe most importantly, it is visible that the first scenario gave considerably lower error in almost all algorithms when compared to the second scenario.

6 CONCLUSION

As a conclusion, the implementation of the Surprise library was successfully achieved and important improvements were observed when using the clustering and filtering of the dataset before using the recommendation algorithms.

Table 2: Errors with and without clustering overall

Method	Post Clustering RMSE	Post Clustering MAE	All in One RMSE	All in One MAE
SlopeOne	0.1457	0.0957	0.2103	0.0991
SVD	0.1454	0.0948	0.2102	0.0975
SVD++	0.1457	0.0956	0.2104	0.0988
CoClustering	0.1544	0.0860	0.1917	0.0803
NMF	0.1366	0.0686	0.2004	0.0723

As future work, we propose to implement ALS method [3], which is more natural for this type of data, to improve the clustering mechanism in order to maximize the quality of the recommendations, and to use a recommendation list-oriented metric, such as MAPk.

ACKNOWLEDGMENTS

The authors would like to thank PhD. Denis Parra for providing the background needed to develop the methodology used in this document, along with his teachings assitants, without whom this work would have never been possible.

The authors would also like to thank data exploration provided by the users in the contest forum that allowed a better comprehension of the dataset.

REFERENCES

[1] The Santander Group. 2016. Kaggle: Santander Product Recommendation. Retrieved December 05, 2018 from https://www.kaggle.com/c/santander-product-recommendation

[2] Tianqi Chen & Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), 785–794. https://doi.org/10.1145/2939672.2939785

[3] Koren Y. & Volinsky C. Hu, Y. 2008. Collaborative filtering for implicit feedback datasets. *Eighth IEEE International Conference on Data Mining* (2008), 263–272.

[4] Zhou M. Xia Y. & Zhu Q. Luo, X. 2014. An efficient non-negative matrix factorization-based approach to collaborative filtering for recommender systems. , 1273-1284 pages.

[5] Karypis G. Konstan J. & Riedl J. Sarwar, B. 2002. Incremental singular value decomposition algorithms for highly scalable recommender systems. *Fifth International Conference on Computer and Information Science* (2002), 27–28.

[6] SlopeOne 2018. Retrieved May 27, 2017 from https://surprise.readthedocs.io/en/stable/slope_one.html#surprise.prediction_algorithms.slope_one.SlopeOne