Hybrid model rating prediction with Linked Open Data for Recommender Systems

Andrés Moreno[1 2] Christian Ariza-Porras[1], Paula Lago[1], Claudia Jiménez-Guarín[1], Harold Castro[1], and Michel Riveill[2]

[1] School of Engineering, Universidad de los Andes, Bogotá, Colombia
[2] Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, 06900 Sophia Antipolis, France
{dar-more, cf.ariza975, pa.lago52,cjimenez, hcastro}@uniandes.edu.co, riveill@unice.fr

Abstract. We detail the solution of team uniandes1 to the ESWC 2014 Linked Open Data-enabled Recommender Systems Challenge Task 1 (rating prediction on a cold start situation). In these situations, there are few ratings per item and user and thus collaborative filtering techniques may not be suitable. In order to be able to use a content-based solution, linked-open data from DBPedia was used to obtain a set of descriptive features for each item. We compare the performance (measured as RMSE) of three models on this cold-start situation: content-based (using min-count sketches), collaborative filtering (SVD++) and rule-based switched hybrid models. Experimental results show that the hybrid system outperforms each of the models that compose it. Since features taken from DBPedia were sparse, we clustered items in order to reduce the dimensionality of the item and user profiles.

Keywords: semantic web, recommender systems

1 Introduction

Recommender systems (RS) are automatic agents that attempt to suggest new or interesting items to users. A number of different algorithms have been proposed to improve the performance of recommender systems, which can be classified in two groups: collaborative-filtering techniques and content-based filtering techniques. Collaborative-filtering techniques (CF) are based on the fact that similar users like similar items and thus base their predictions in the ratings provided by similar users. Content-based techniques (CB) build a user profile of interests based on the features of the items the user has rated. On cold-start situations, when items have few ratings, neither system can perform well. This is because they don't have the amount of data needed to find either true similarities among users (CF) or to construct the user profiles (CB).

In these circumstances, more data is needed, either to describe the items or the users. Thanks to linked open data initiatives, information about items can be found on the web. Task 1 of the linked open data enabled recommender systems challenge purpose was to predict the rating a user would give to an item in a

cold-start situation. In order to be able to use a CB solution, linked-open data is used to obtain features that describe items in machine-readable format.

The paper is organized as follows: we describe the provided dataset, the performance metric used to evaluate the predictions, give an overview of the proposed solution and discuss the obtained results.

Dataset description The DBbook dataset contains 75559 ratings of 6166 books by 6181 users. The possible ratings that a user can assign to an item are $\mathcal{O} = \{0, 1, 2, 3, 4, 5\}$. The `ratings` file has 3 fields: a user id, an item id, and the rating. Each item has been rated by at least one user, but the evaluation set includes some books not rated in the training set, representing a cold-start situation. The dataset also provides a mapping of each item id to a DBPedia URI which gives access to a semantic description of items. Given this description, we can define each book with a set of concepts C_i taken from DBPedia. We use the following concepts to describe a book: author, categories, literary genres, and subject. Figure 1 depicts the feature extraction process. The feature space size is 14001 concepts. Each book has an average of 16.49 features with standard deviation (std) of 6.18. Each feature appears in an average of 9.62 books with std of 118, and a max of 4030.

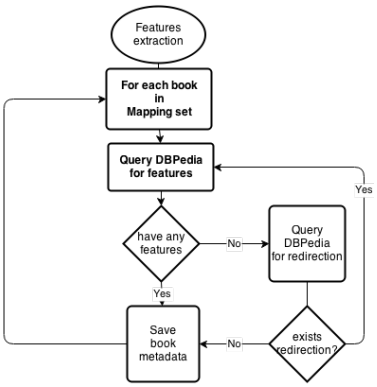


Fig. 1: Semantic Features Extraction

2 Prediction Model

Burke [1] describes different ways in which recommender system models can be combined. The *switched* strategy maintains different models in parallel and reports to the user the prediction of the model with higher confidence. We use as base model a widely known CF algorithm (SVD++) (Section 2.2). However, since traditional CF systems usually make incorrect predictions when no previous

ratings about the item are known, the prediction of the switched hybrid model on cold-start situations is delegated to the CB model explained in Section 2.1. The measure used to evaluate the predictive performance of the system is the Root Mean Square Error (RMSE). Let T be the rating set of a hold-out set (test set), T_{ui}, the rating that the user u gave to item i and $\hat{r_u i}$ the model prediction, the RMSE is defined as $RMSE := \sqrt{\frac{1}{|T|}\sum_{T_{ui}\in T}(\hat{r}_{ui} - T_{ui})^2}$. In the remainder of this section, we will describe the models that take part in our system.

2.1 Content Based model

On a CB model, a user u has a profile with a list of non duplicate concepts C_u and a set of $|\mathcal{O}|$ vectors $w^o \in \mathbb{R}^{|C_u|}, o \in \mathcal{O}$. For each example of user-item interaction, each of the concepts that are related to the item (C_i) are considered for addition into the user's list C_u. We use an inclusion policy using a sliding window min-count sketch structure [2] based on the work developed in [4]: All concepts seen by the user at least N times during the window duration of the sketch are present in the user's list, and the size of the vectors w^o is updated. After modifying the list and the w^o vectors' length, the weights of the vector are adjusted using a stochastic logistic regression strategy. Let $r_{ui} \in \mathcal{O}$ the rating user u gives to item i and $m_{ui} = meta(C_i \times C_u) \rightarrow \mathbb{R}^{|C_u|}$ a function that takes the concept set of an item and converts it into a binary vector where each coordinate is 1 if the user's concept belong to the items list ($m_{ui}[f] = \mathbb{1}_{C_u[f]\in C_i}$). For each vector w^o $\mathcal{O}$, we predict $\sigma(\langle w^o, m_{ui}\rangle)$ and update each of the vectors as in $w_u^o \leftarrow w_u^o - \gamma(\sigma(\langle w^o, m_{ui}\rangle) - \mathbb{1}_{r_{ui}=o})m_{ui}$, where $\sigma(c)$ is the sigmoid function. The rating prediction under this model is calculated as in $\hat{r}_{ui} = \frac{\sum_{o\in\mathcal{O}}\sigma(\langle w^o, m_{ui}\rangle)\times o}{\sum_{o\in\mathcal{O}}\sigma(\langle w^o, m_{ui}\rangle)}$

Feature Generation and evaluation We use DBPedia to retrieve book features as described in section 1. Using all the retrieved features the predictor performance was lower than expected and, as shown in Figure 2, if we increase the minimum inclusion rate, the performance declines. A quick evaluation of the features shows that some of them are highly correlated, which led us to consider that clusters of features may provide more information to the predictor. We created clusters of features by co-occurrence, using k-means with cosine distance, convergence delta of 0.01 and 200 iterations. Figure 3 depicts the dataset generation process.

We vary the number of clusters (k) and measure the performance against the test set. In Figure 4a, we can see that the predictor performance using clusters is better than using all the extracted features. Although with 23 clusters we have a slightly better result against the test set, we use the 50 clusters because this had better performance using the evaluation tool.

With these 50 clusters as features, each book has an average of 2.1 features with a std of 0.57. Each feature appears on average in 344 books, with a std of 1332. When trained with these new features, the predictor improves its performance notably with a min inclusion rate of 2, as shown in Figure 4b. The best RMSE with the content-based predictor on the evaluation tool was 0.969.

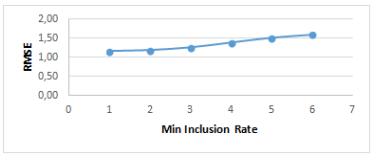


Fig. 2: RMSE vs inclusion rate for book features

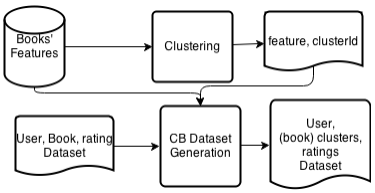
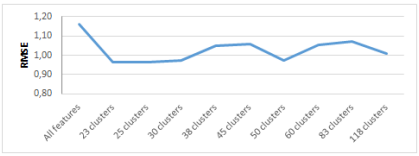
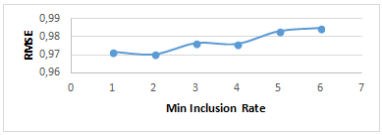


Fig. 3: Content Based Dataset Generation



(a) All features vs different cluster's size



(b) With 50 clusters changing the minimum inclusion rate

Fig. 4: Content Based predictor performance using clusters

2.2 Collaborative Filtering model

The SVD++ algorithm [3] prediction rule uses the global average of ratings (μ) and the *bias* or deviation from the mean for each user (b_u) and each item (b_i) as model parameters. In order to account for the user-item interaction the SVD++ model represents each user as a vector x_u and each item as an vector $y_i \in \mathbb{R}^k$. Each item is represented by an extra vector $z_i \in \mathbb{R}^k$ that is used by the prediction rule to represent the items the user has rated into her profile. Let $R(u)$ the set of items the user u has rated, the prediction under the SVD++ model is given by $\hat{r}_{ui} = \mu + b_i + b_u + y_i^T \left(x_u + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} z_j \right)$. When an item has not been seen by the system, the prediction rule only uses the sum of the global mean and the user bias. Parameters of the model are learned using a regularized stochastic gradient descent strategy.

3 Model validation

To test the performance of the hybrid model, we generated 5 datasets with approximately 80% for training and the 20% for testing, each of these datasets had a different percentage of cold-start ratings varying from 5% to 25%. The model delegates the prediction to the CB model only when it has not seen the item before. Fig. 5 shows the RMSE of the hybrid model as the number of new items in the test set increases. The results show that the hybrid model outperforms CF for a low number of cold-start items.

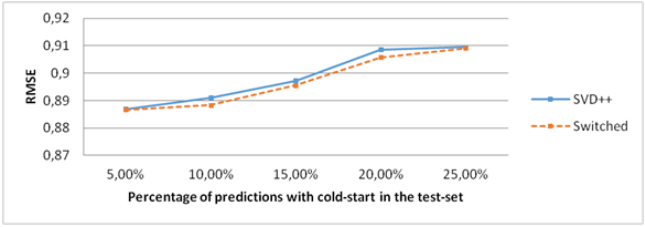


Fig. 5: RMSE of the hybrid model vs SVD++ on cold-start

4 Conclusions

We have described our approach to improve the performance of recommender systems using linked open-data that is freely available on the web. Open data provides descriptions of items that help the recommender system understand better why a user likes an item (a user may like a book because of its author, its literary genre, its main subject, etc). This approach can help alleviate the new item cold-start problem. However, users may like items based on subjective features such as tone which are not provided in the open-data repositories used. For this reason, we proposed an hybrid model based on rules that uses a pure collaborative approach when enough ratings are present, and uses a content-based approach in the other cases. Our model had a RMSE of 0.8787 against the quiz set provided by the challenge. Open-data such as data from social-networks can also be used to describe users and calculate similarities of new users based on this data. This could further improve the performance of recommender systems under a new-user cold start problem.

References

1. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (Nov 2002), http://dx.doi.org/10.1023/A:1021240730564
2. Dimitropoulos, X., Stoecklin, M., Hurley, P., Kind, A.: The eternal sunshine of the sketch data structure. Comput. Netw. 52(17), 3248–3257 (Dec 2008), http://dx.doi.org/10.1016/j.comnet.2008.08.014
3. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 426–434. KDD '08, ACM, New York, NY, USA (2008), http://dx.doi.org/10.1145/1401890.1401944
4. McMahan, H.B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Golovin, D., Chikkerur, S., Liu, D., Wattenberg, M., Hrafnkelsson, A.M., Boulos, T., Kubica, J.: Ad click prediction: A view from the trenches. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1222–1230. ACM, New York, NY, USA (2013)