

Review Aware Recommender Systems

Fatima Zahra Lahlou, Houda Benbrahim and Ismail KASSOU

ALBIRONI Research Team, ENSIAS, Mohamed V University, Rabat, Morocco

fzl.lahlou@gmail.com, benbrahim@ensias.ma, kassou@ensias.ma



Abstract

In this work, we present a new Context Aware Recommender Systems solution that takes the whole reviews as input in addition to users and items. Along with this solution, we develop a new CARS algorithm, based on the generic recommendation algorithm Factorization Machines (TCAFM). TCAFM takes as input contextual data where the textual reviews are considered as context and compute context aware rating predictions. Experiments show that using the proposed solution significantly improves recommendation quality. Furthermore, using TCAFM leads to additional improvements.

Keywords: Context Aware Recommender Systems, Text Classification, Machine Learning, Natural Language Processing.

Introduction

Recommender Systems are systems that filter information depending on users' profiles and suggest items that might match their preferences. Most RS focus on users and items in computing predictions. However, there is other contextual information (such as time, weather or accompanying persons) that may influence user decisions.

Context Aware Recommender Systems (CARS) are Recommender Systems that consider, in addition to users and items, other contextual information for computing predictions.

Context: "any information that can be used to characterize the situation of an entity"

Researches in CARS address three main issues:

- obtaining contextual information
- selecting which contextual information to use in a particular recommendation task
- incorporating these information in the recommendation process

⇒ We developed a solution to do the three tasks simultaneously and automatically

In this work:

- we present a new CARS solution that takes the whole reviews as input in addition to users and items and compute context aware rating prediction without requiring any feature engineering to extract context from reviews. We call this solution: Review Aware Recommender Systems.
- We develop a new CARS algorithm that is tailored to textual contexts resulting from the Review Vectorization stage. We build our algorithm on the generic algorithm Factorization Machines [1] and we call it Textual Context Aware Factorization Machines (TCAFM).
- TCAFM takes as input contextual data where the textual reviews are considered as context and compute context aware rating predictions.
- Experiments show that using the proposed solution significantly improves recommendation quality. Furthermore, using TCAFM leads to additional improvements.

Contextual Information In Reviews

Even if reviews are primarily written to express sentiments and preferences about items, they can carry some contextual information.

"After 3 weeks business trip and staying at 5 different hotels in Jeddah, Riyadh and Eastern Province I can say that this Hotel was the most suitable for me. Room service quality, quantity of food was excellent (I usually do not use the word excellent), service was quite efficient. Health club is a story by itself energetic and still you have a beautiful scenery of the sea and gardens quite breath taking."

Getting Context From reviews:

What if we Consider the whole Review as Context ?

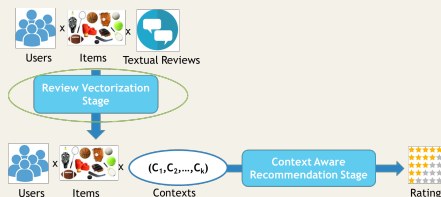


We are seeking to:

- Represent reviews in such a way that contextual information is brought out.
- Develop a CARS algorithm tailored to this new representation and all its associated challenges.
- Improve recommendation results !

The proposed approach

Our proposed approach consists of a two-stage process. The first stage is a review vectorization one where reviews are represented into a new space in order to brought out contextual information. The second stage is a context aware recommendation stage where we apply a CARS algorithm on resulting contextual data.

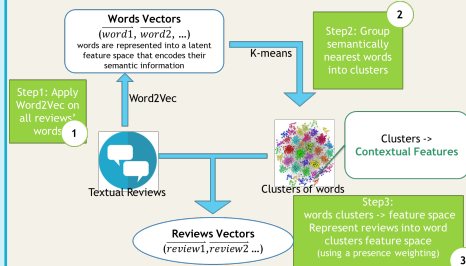


The Review Vectorization Stage

How to represent reviews into a vector space in such a way that contextual information are brought out?

Within the review vectorization stage:

- we use Word2Vec technique to produce vectors for reviews words.
- Then, using these resulting vectors, we apply the K-means clustering algorithm to group semantically nearest words into clusters. In this way, closer words in terms of meaning are grouped together. This step ends up with a set of words clusters.
- The next step consists of using these clusters as a space to represent documents, here reviews. The output of this step is document vectors. Because we are interested in contextual information, and words carrying such information may appear only once in the review, unlike noisy words (as 'table', 'food' ...) that may appear frequently, we are interested in presence weighting instead of frequency. This step finally ends up with reviews represented in the new word clusters feature space.



The Context Aware Recommendation Stage

Apply a Context Aware Recommender System on resulting contextualized data.

But,

- Resulting data are even sparser!
- There are clusters (contextual features) that not reflect contextual information!

We develop a new algorithm, that we call Textual Context Aware Factorization Machines Algorithm (TCAFM), and that

- is built on Factorization Machines (FM) [1], as this last one is tailored for sparse data.
- Reformulates the FM model so that, for each contextual factor, a corresponding weight is added in order to capture its importance. These are learned automatically along with other model parameters in the optimization procedure.

More details about TCAFM and the whole solution can be found here [2].

$$\text{Factorization Machines (FM)} \quad y(x) = w_0 + \sum_{i=1}^p w_i x_i + \sum_{i=1}^p \sum_{j=1}^p \langle v_i, v_j \rangle < v_i, v_j > x_i x_j$$
$$\text{TCAFM} \quad y(x) = w_0 + \sum_{i=1}^p w_i \beta_i x_i + \sum_{i=1}^p \sum_{j=1}^p \langle v_i, v_j \rangle < v_i, v_j > \beta_i \beta_j x_i x_j$$

Add a weight β_i for each feature
If x_i represent a user or an item, then $\beta_i = 1$

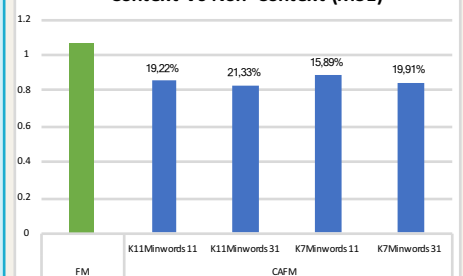
$$x_i \rightarrow \beta_i x_i$$

Experiments

We conduct experiments on a dataset taken from the Yelp Dataset Challenge and consisting of 23794 reviews for 1414 restaurants and 6267 users. Using this dataset, we compare different setting for the review vectorization stage, and TCAFM with the CARS state-of-the-art algorithm, CAFM [3] in the recommendation stage. We also compare using context with not using context using FM.

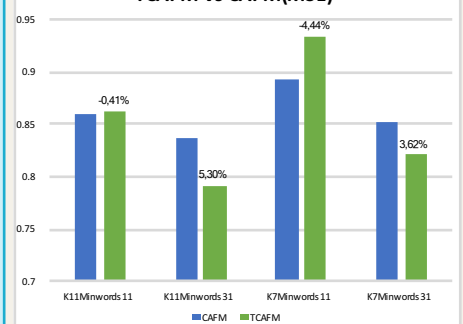
Results show that using context from reviews throughout our proposed solution considerably improve recommendation. Moreover, results demonstrate that TCAFM outperforms CAFM when settings are well chosen.

Context Vs Non-Context (MSE)



Comparing Non-contextualized and Context Aware Recommendation in terms of MSE (Less is better)

TCAFM vs CAFM (MSE)



Comparing TCAFM and CAFM in terms of MSE (Less is better)

Conclusion

- We presented Review Aware Recommender Systems, a solution that allows to:
 - automatically use relevant contextual information from reviews and
 - incorporate it into recommendation automatically, simultaneously, effectively and also without requiring any feature engineering.
 - We also presented TCAFM, a new context aware recommender algorithm that is adapted to textual context.
 - Conducted experiments showed that:
 - using context from reviews throughout our proposed solution considerably improve recommendation.
 - TCAFM outperforms CAFM when settings are well chosen.
- These can be considered as new and important findings in CARS research.

References

- [1] S. Rendle, "Factorization Machines," in 2010 IEEE International Conference on Data Mining, 2010, pp. 995–1000.
- [2] F. Z. Lahlou, H. Benbrahim, and I. Kassou, "Textual Context Aware Factorization Machines: Improving Recommendation by Leveraging Users' Reviews," in Proceedings of the 2nd International Conference on Smart Digital Environment, Rabat, Morocco 2018, pp. 64–69.
- [3] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast Context-aware Recommendations with Factorization Machines," in Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 2011, pp. 635–644.