Abstract

In this work, we present a new Context Aware Recommender Systems solution that looks at the whole reviews as input in addition to users and items. Along with this solution, we develop a new CAAS algorithm, based on the generic recommendation algorithm Factorization Machines, called:Contextual Aware Factorization Machines (TCAFM). TC AFM 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 TC AFM 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 (CAAS) are recommender systems that consider, in addition to users and items, other contextual information like time, weather or accompanying persons that may influence user decisions.

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

Researches in CAAS address these main issues:

- (i) obtaining contextual information
- (ii) selecting a rich contextual information to use in a particular recommendation task
- (iii) representing this information in the recommendation process.

We developed a solution to do the three task simultaneously and automatically.

In this work:

- We present a new CAAS solution that takes the whole reviews as input in addition to users and items and computes context aware rating predictions without requiring any feature engineering or ad hoc context from reviews.
- We call this solution Context Aware Recommender Systems.
- We develop a new CAAS algorithm that is tailored to textual contexts resulting from the textual vectorization stage. We build our algorithm on the generic machine learning algorithm (FM) [1] and called it Temporal 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.

"Almost a 3 weeks business trip and staying at 5 different hotels in Jeddah, Riyadh and Eastern Province. I can see that this fjord was the most suitable for me. Room service quality, quantity of food was excellent (I usually do not use the room service), service was quite efficient. Hotel 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:

- (i) capture contextual features that carry contextual information
- (ii) develop a CAAS algorithm based on the new representation and all its associated challenges
- (iii) improve recommendation results

The proposed approach

Our proposed approach consists of a two-stage process. The first stage is a review vectorization stage where reviews are represented as a new space in order to bring the contextual information. The second stage is a context-aware recommendation stage where we apply a CAAS 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 closest words into clusters. In this way, older words in terms of meaning are grouped together. This step ends up with a set of clusters.
- The next step consists of using these clusters as a space to represent documents, here reviews. The output of this step is a document vector. 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.

Context Aware Recommendation Stage

Apply a Context Aware Recommender System on resulting contextual 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 Temporal Context Aware Factorization Machines (TCAFM), and that:

- Is built on Factorization Machines (FM) [1] as this last one is tailor-made for sparse data.
- Reformulates the FM model so that, for each contextual feature, 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].

The Review Vectorization Stage

Clustering Features

Contextual Information

The Context Aware Recommendation Stage

The proposed approach

Our proposed approach consists of a two-stage process. The first stage is a review vectorization one where reviews are represented as a new space in order to bring the contextual information. The second stage is a context-aware recommendation stage where we apply a CAAS 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 closest words into clusters. In this way, older words in terms of meaning are grouped together. This step ends up with a set of clusters.
- The next step consists of using these clusters as a space to represent documents, here reviews. The output of this step is a document vector. 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.

Context Aware Recommendation Stage

Apply a Context Aware Recommender System on resulting contextual 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 Temporal Context Aware Factorization Machines (TCMF), and that:

- Is built on Factorization Machines (FM) [1] as this last one is tailor-made for sparse data.
- Reforms the FM model so that, for each contextual feature, 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].

The Review Vectorization Stage

Clustering Features

Contextual Information

The Context Aware Recommendation Stage

The proposed approach

Our proposed approach consists of a two-stage process. The first stage is a review vectorization one where reviews are represented as a new space in order to bring the contextual information. The second stage is a context-aware recommendation stage where we apply a CAAS 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 closest words into clusters. In this way, older words in terms of meaning are grouped together. This step ends up with a set of clusters.
- The next step consists of using these clusters as a space to represent documents, here reviews. The output of this step is a document vector. 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.

Context Aware Recommendation Stage

Apply a Context Aware Recommender System on resulting contextual 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 Temporal Context Aware Factorization Machines (TCFM), and that:

- Is built on Factorization Machines (FM) [1] as this last one is tailor-made for sparse data.
- Reforms the FM model so that, for each contextual feature, 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].

Conclusions

We presented Context 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 content.

Conducted experiments showed that:

- using context from reviews through proposed solution considerably improves recommendation.
- TCAFM outperforms CAFM when settings are well chosen.

These can be considered as new and important findings in Context Aware Recommendation.

References