Temporally Driven, Context-Aware Recommender System

Luca Leeser

09/30/17

Abstract:

My research proposal is a novel recommender system that utilizes context-aware topic modelling neural networks within a temporal context. Unlike products, restaurants change over time, whether it is due to quality change, service change, menu change, or some other latent factor. I am interested in potential project because of my interest in recommender systems as a whole. I want to explore the use of neural networks for developing contextually aware recommender systems, I also believe that adding a temporal component to recommender systems is a novel approach since most rec systems are biased towards static products.

Research Context:

Most recommender systems are based on the concept of matrix factorization. Given a Rating vector R $(R \in R^{N\,x\,M})$, where element r_{ij} is a rating given by user i for product j. This matrix can be factorized using singular-value decomposition to form matrices U, Σ and V. Where U forms the basis of the column space of R, Σ being a diagonal matrix of eigenvalues, and V being the basis of the row space of R. Each row of U represents a user, u_i, while each row of V represents a product v_j. The dimensions of these matrices form the latent factors that are being considered in a rating. Such latent factors in the context of Yelp could be seen as type of food (Mexican, American, Italian, etc) and other latent factors like food quality, service, cost, etc. In a real world context, this matrix R is sparse, so it is impossible to perform SVD on it. This is solved by performing gradient descent on the matrix R using a loss function like the following:

$$L(R) = \sum^{N}\sum^{M} I_{ij}(r_{ij} - u_i^T v_j)^2 + \lambda_u \sum^{N}(u_i^T u_i) + \lambda_v \sum^{M}(v_j^T v_j)$$

This method is clearly possible with the Yelp dataset since we have access to user ratings and restaurant ratings. However, this method is lacking since it does not consider reviews, which capture latent factors that matrix factorization would likely miss. This is why LDA can be useful in this context, it is able to capture the latent factors seen in the review text in an unsupervised manner, which is also utilized to improve recommender systems. However, this is also lacking, since LDA fails to capture *contextual* semantics in text, thus giving us an incomplete picture of the latent factors that exist. An example of this can be seen as follows:

Consider these two sentences:

1. People would not trust the man, because he would often betray them.
2. People would often trust the man, because he would not betray them.

To us it is obvious that they mean very different things. However, under a bag-of-words model like LDA, they are practically identical (they both have the exact same word distribution). This is why the relations between the words in a sentence should be considered for a more complete picture. This problem has already been solved by one of my reference papers, titled "Convolutional Matrix Factorization for Document Context-Aware Recommendation". The authors of this paper trained a convolutional neural network on vectorized product descriptions to generate a product vector v , which could be used to find the rating by finding the inner product between v and a user vector. This architecture resulted in state-of-the-art performance when compared to MF, PMF, and even other deep learning based recommender systems on multiple datasets. However, this method is also limited in some ways. For example, it estimates the entire user matrix U using coordinate descent, failing to capture contextual information found in the user reviews. I also want to explore including a temporal component to this contextually aware method of matrix factorization, since businesses are not static products.

Data / Design

I would use restaurant reviews to generate the latent factor vectors for each restaurant. I would also use user reviews to generate a restaurant/user rating matrix, which I could use to estimate matrix U through coordinate descent, where you optimize a variable while treating other variables as constant. I could train on other datasets containing user reviews and product reviews. Other datasets I found being used in the papers include Amazon instant video reviews, and movie review datasets.

Methods

As described in the paper, “Convolutional Matrix Factorization for Document Context-Aware Recommendation”, I would optimize matrix R by finding U and V. This can be done in close form as follows:

$$u_i \leftarrow (V I_i V^T + \lambda_u I_K)^{-1} V R_i$$

This is derived by taking the MAP estimation of matrices U, V, W and performing coordinate descent by treating V as a constant. This is explained in further detail in the paper.

I would then construct a neural network that is trained on each document in the form of an embedding generated by a pretrained model, such as Glove. This network would be trained to generate topic models such that the inner product of the user vector and the generated topic model represents the predicted rating for restaurant r for user u. I want to explore different kinds of deep net architectures for generating convolutional network including LSTMs and CNNs. The network would also ideally be pretrained in an unsupervised setting utilizing an autoencoder model, since we are ultimately creating a feature picking model, which would benefit greatly from such a pre-training regime.

As for implementing a temporal component to reviews, I have several ideas that I want to pursue. The most simple option being creating a weighted average over restaurant reviews to model the expected topic model for the restaurant drawn from a distribution skewed temporally. Another option is to train a RNN to predict this expected restaurant topic vector, where the RNN potentially has an auxiliary input being a weight representing the distance that review is from being the most recent one. Thus, putting more weight on more recent reviews. The RNN would output the predicted next topic model given the prior distribution (trend prediction). There are a few issues that need to be addressed, firstly preventing the RNN from becoming an identity function to the prior trained LSTM, another being dealing with variance in topics over time, since it wouldn't be unusual to see a sequence of 1 and 5 stars in a time sequence of reviews. Some form of data preprocess on the latent topic vectors would probably be required, potentially through PCA.

Significance

This model would be helpful to Yelp because it would provide them with a context-aware recommender system that incorporates reviews over time, since a restaurant from a few years ago may no longer be the same as it is now. This application domain is not specific to Yelp and could prove useful to other services such as Google Reviews, GrubHub, Seamless, etc.

References

https://arxiv.org/pdf/1705.08200.pdf

https://arxiv.org/pdf/1604.02038.pdf

https://dl.acm.org/citation.cfm?id=2959165

https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2015-56.pdf