Multi-Criteria Model for Recommendation Systems

Presented By:
Abhishek Yadav (2017csb1063)
Nitin K (2017csb1093)

Introduction (1)

- Single rating Recommender Systems have been in existence everywhere. Multi-criteria predictions have been proved to be more accurate
- Deep learning for recommender systems began to gain massive interest, and many recommendation models based on deep learning have been proposed (capable of capturing complex relations)
- Authors proposed a novel multi-criteria **collaborative filtering** model based on deep learning (not yet any developed)

Reference to the paper: http://www.sciencedirect.com/science/article/pii/S0950705119302862

Introduction (2)

- Most Recommendation Systems (RSs) use single ratings in predictions and this is considered as a limitation because when a user might take into consideration more than one aspect, so additional aspects may increase the accuracy of the prediction
- Therefore, Multi-Criteria ratings can lead to recommendations which may be more accurate.
- **For example**, in a music recommender system, some users may like music based on its rhythm, beat, or timbre, while others may like the same music but for its melody, tempo, texture, or any other combinations of the distinct attributes of that music.

Introduction (3)

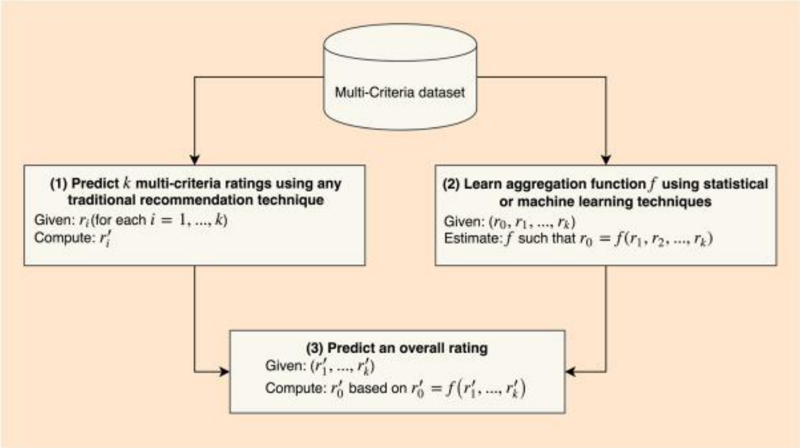
This model comprises of two stages:

- In the **first stage**, we obtain the users and items features and use them as an input to a deep neural network that predicts the criteria ratings, and
- In the **second stage**, we use a Deep Neural Network to learn the relationship between the overall ratings and the criteria ratings.
- The authors claim is that on a real-world dataset, model achieves better results that other state-of-the-art methods.

Related Works

- **Collaborative filtering** is divided into two general classes: (1) neighborhood method (user-based or item-based), and (2) model-based methods (learn predictive model and make recommendations)
- **Multi-criteria recommendations** techniques can also be divided into two classes: (1) memory-based, and (2) model-based techniques (build a predictive model)
- Two ways to calculate memory-based techniques, (a) Use aggregation methods (average, weighted sum) to calculate aggregate value. (b) Calculate the distance between multi-criteria ratings directly using multi-dimensional distance metrics (Euclidean, Manhattan).

System Overview



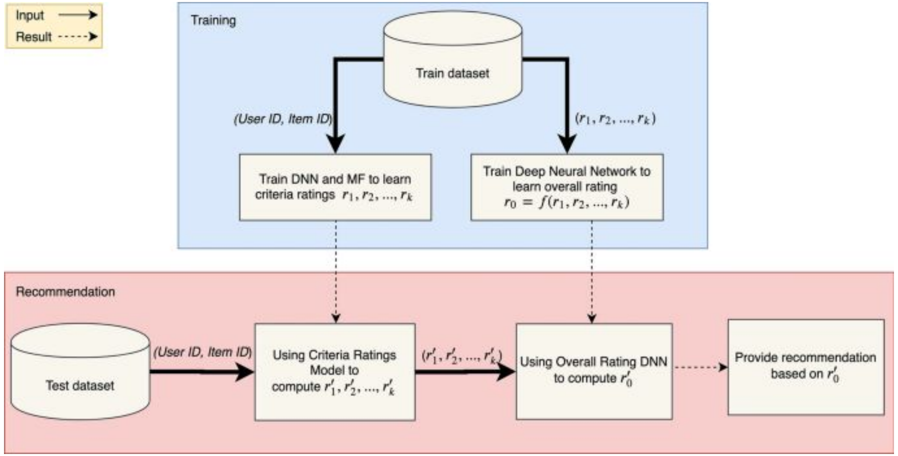
The proposed model follows the aggregation function-based approach:

1. Predict unknown k multi-criteria ratings using any recommendation technique
2. Learn aggregation function **f** using statistical or machine learning techniques
3. Predict unknown overall ratings using **f** and predicted multi-criteria ratings

Implementation (1)

Predict criteria ratings:

- Used a deep neural network to predict the criteria ratings for a user on an item.
- Concentrates only on pure collaborative filtering, so used only itemID or userID as a feature.



Implementation (2)

Learn aggregation function:

- In the aggregation-function-based approach, the general and intuitive assumption is that the overall rating can be estimated by an aggregation function of multi-criteria ratings

$$\mathbf{r_0 = f(r_1,\ r_2,\ \ldots,r_k)}$$

- Normalization $\mathbf{(z_i = (r_i\text{-}m)/s)}$ and then dense ReLU followed

Implementation (3)

Recommendation:

- Model parts were separately trained without knowing each other
    a. **Compute Criteria Ratings:** Get the userID and itemID pairs and feed them as inputs to the Criteria Ratings DNN, and then we predict the criteria ratings r'_1,r'_2, ..., r'_k.
    b. **Compute Overall Ratings:** Normalize the criteria ratings r'_1, r'_2, ..., r'_k computed in step (a), feed them as inputs to the Overall Rating DNN, and then predict the overall ratings r'_0.
    c. **Provide Recommendation:** Finally, recommend the optimal items to the user using the overall rating r'_0 as in traditional single rating recommender systems.

Experiments and Results (1)

Dataset

- Multi criteria ratings TripAdvisor Dataset for hotels.
- The dataset contains userId and hotelId.
- It has 7 criteria ratings and 1 overall rating (Range 1 to 5).
- Total hotels : 1850
- Total User : 6136,
- Total Ratings: 55,534

Experiments and Results (2)

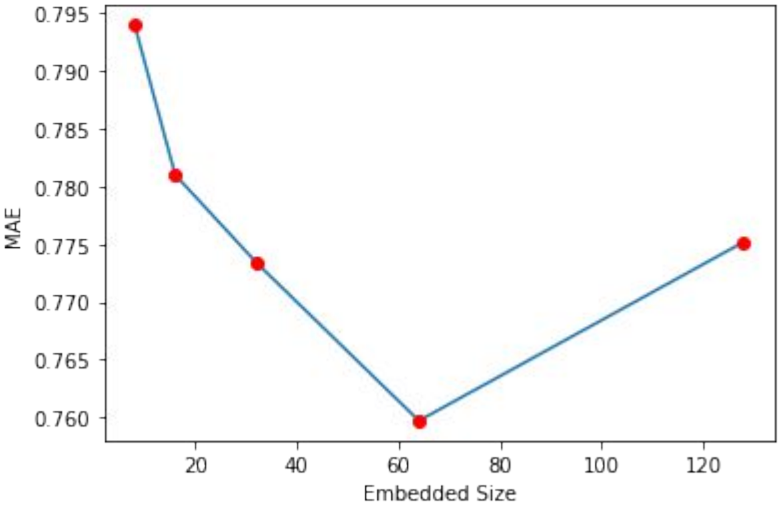
Settings (1)

Criteria Ratings DNN Settings (1)

- Random initialization of DNN parameter with (Mean = 0 and SD = 0.05)
- Used Adam Optimizer with learning rate = 0.001
- For model training we used batch size = 512, epoch = 20, validation = 0.2

Experiments and Results (3)

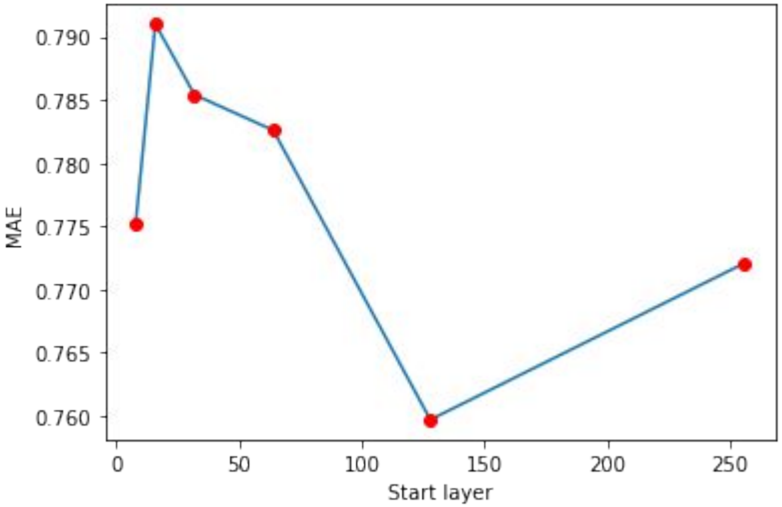
Criteria Ratings DNN Settings (2)



Tested on multiple embedded vector size.
We found that **size = 64** perform best.
Final Embedded vector size = 64

Experiments and Results (4)

Criteria Ratings DNN Settings (3)



Tried different hidden layers.

Hidden layers => [128, 64, 32, 16, 8]

perform best as shown in the Figure.

Final Hidden layers = [128, 64, 32, 16, 8]

Experiments and Results (5)

Settings (2)

Overall Ratings DNN Settings

- Random initialization of DNN parameter with (Mean = 0 and Std = 0.05)
- Used Adam Optimizer with learning rate = 0.001
- For model training we used batch size = 512 , epoch = 20 , validation = 0.2
- **Hidden layers : [64, 32, 16, 8]**

Experiments and Results (6)

Results

- We use 80% training set and 20% validation set and calculated MAE for both criteria ratings model and overall rating model
- **MAE** for Criteria ratings model **= 0.7597**
- **MAE** Overall rating model **= 0.1593**

This is the best result run for different embedded vector size and different hidden layers.

Thankyou

Any Questions/Suggestions?