Exploring Deep Space: Learning Personalized Ranking in a Semantic Space

By: Jeroen B.P. Vurrens, Martha Larson & Arjen P. de Vries (Netherland, 2016)

Present By : Khun Chansopheak (2018124649)

Content

1. Introduction

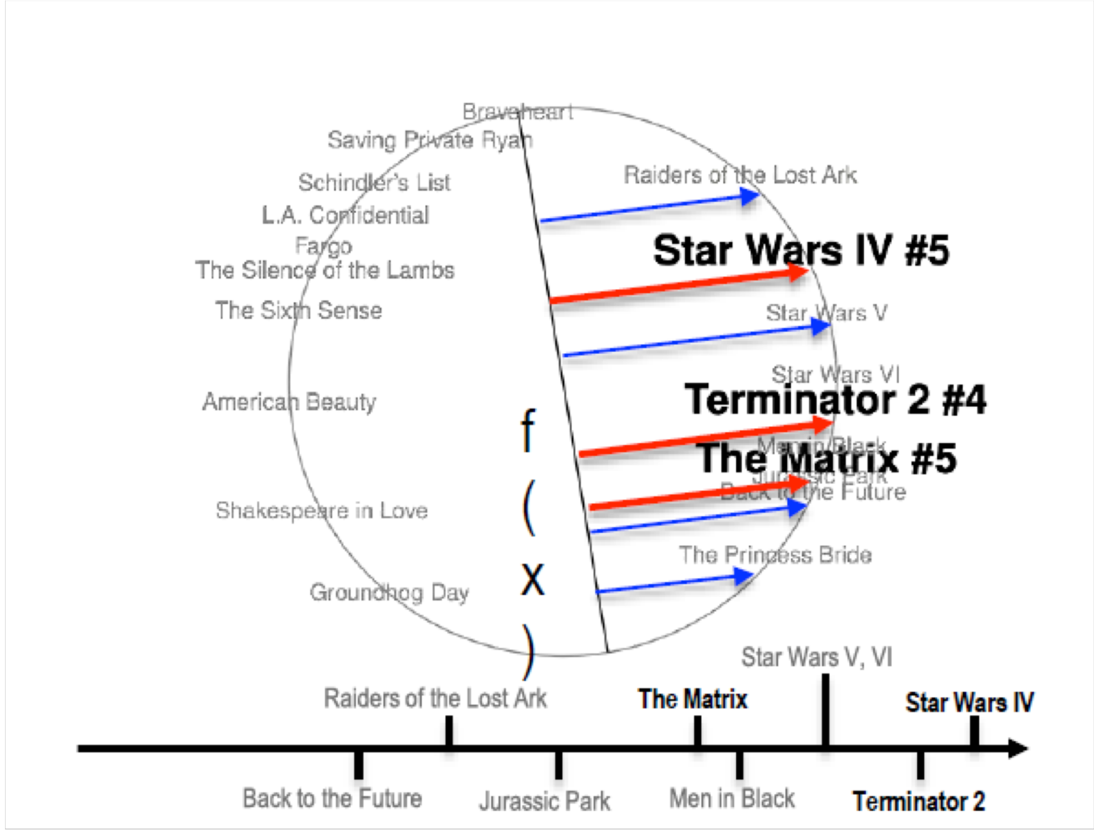
- State-of-art collaborative-filtering systems recommend items by analyzing the history of user-item preference.

- Content-based system analyze data about the items and suggest item to a user that are most similar to the items she like in the past.

- In recent years, semantic space models for various task such as translation and analogical reasoning has been used.

1. Introduction

- In this work, a novel approach for the recommendation of items:
  1. Structure items in a semantic space
  2. Transform this space into a ranked list of recommendations that matches the user's preferences.



1. Introduction

- We show that the same architecture can be used to effectively recommend items using either the text of user reviews or user-item ratings.

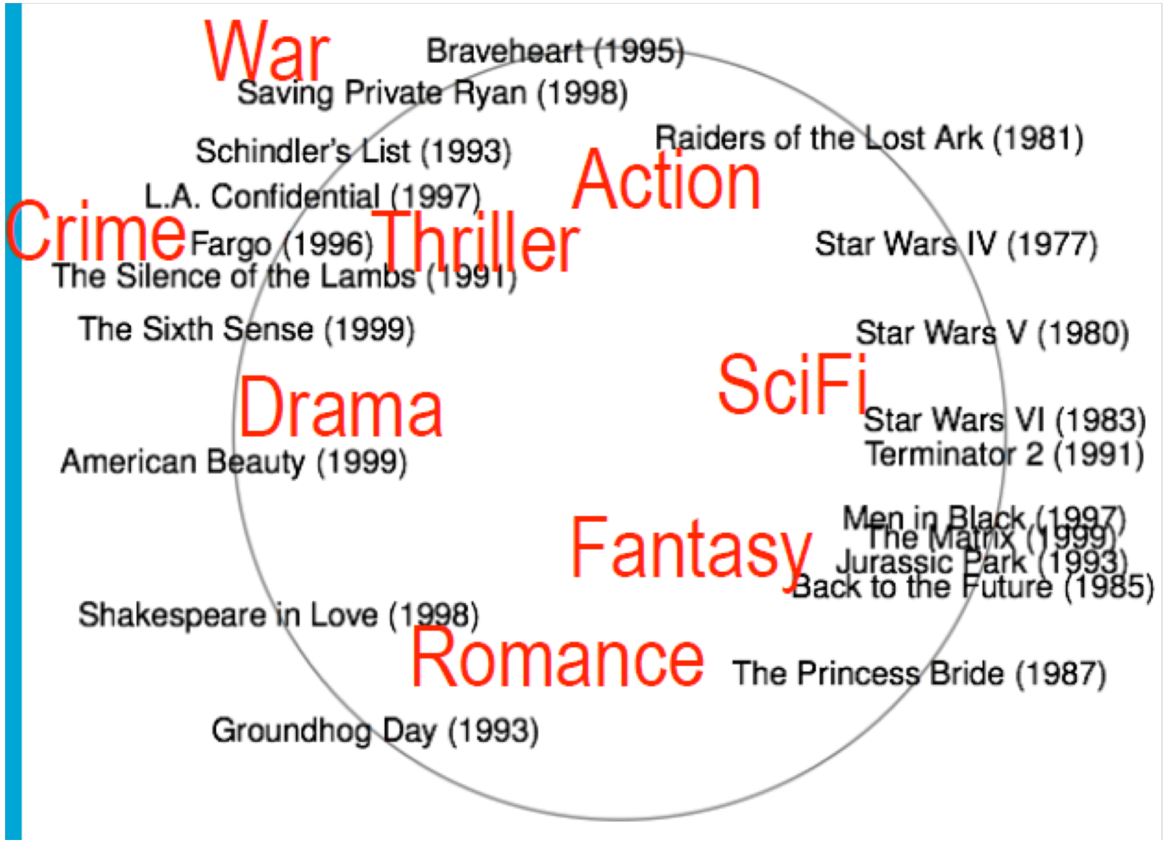
- We evaluate this approach using the MovieLens 1M dataset

- Result show that the proposed approach using user-item ratings significantly outperforms state-of-the-art recommender system (That time, Year 2016)

2. Semantic Spaces for Recommendation System

Semantic spaces

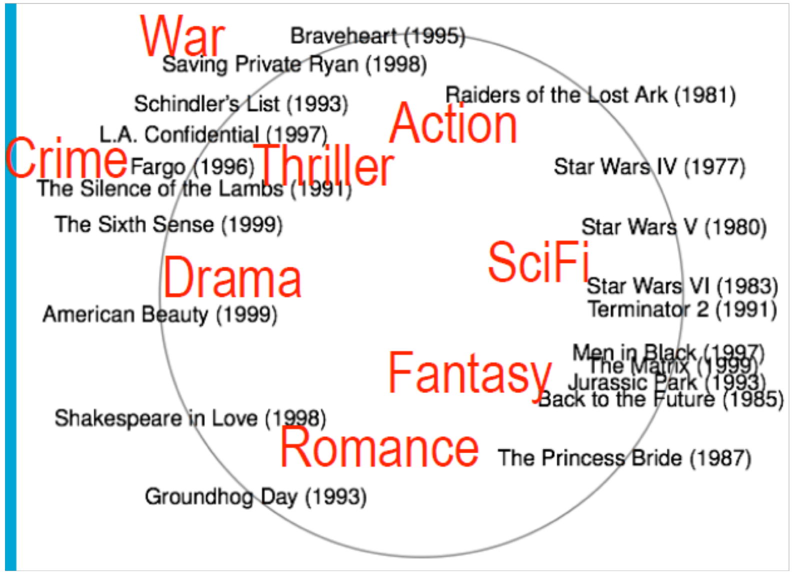
A semantic space model is a way of representation the similarities between contexts in a Euclidean space.



2. Semantic Spaces for Recommendation System

Semantic spaces

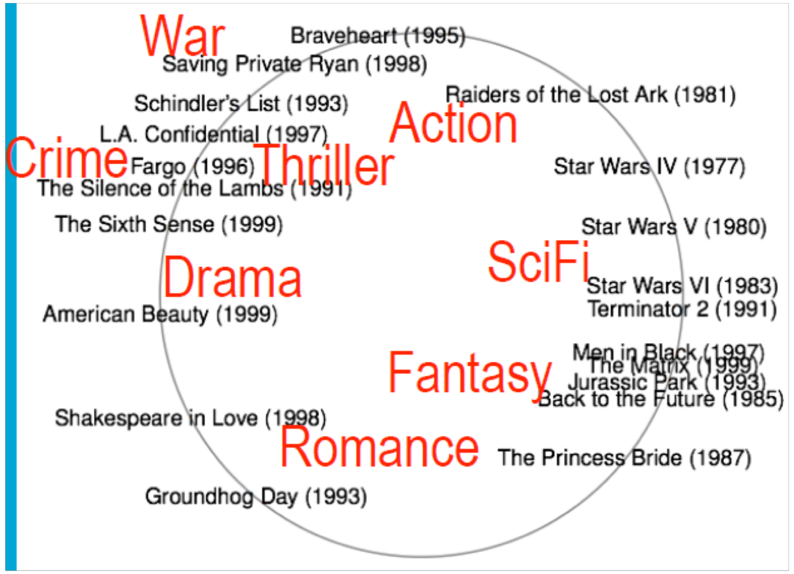
The substitutability between items can be inferred from the observation of being jointly liked by a subset of user, or in a content-based setting by having similar descriptions



2. Semantic Spaces for Recommendation System

Semantic spaces

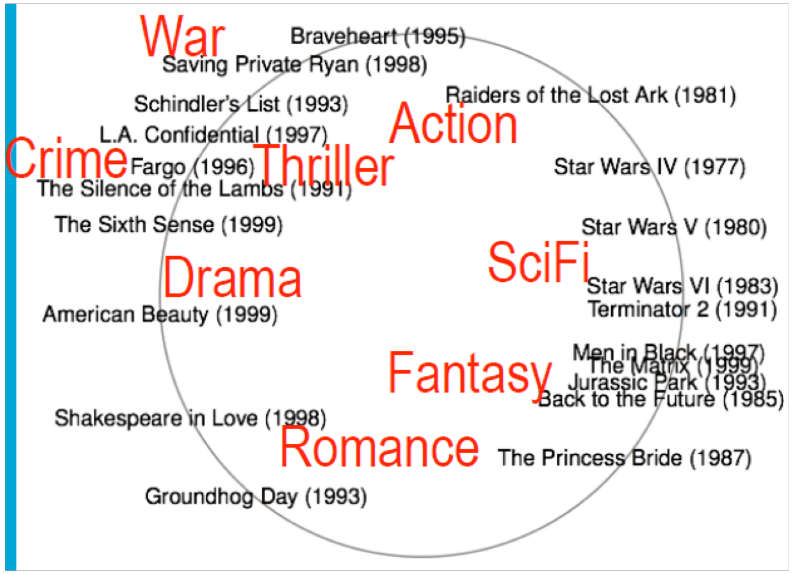
In a near-optimal high-dimensional semantic space, the 'best' recommendation candidates are likely to be positioned in close proximity to the items the user rated highly.

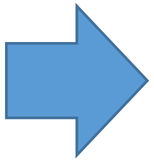


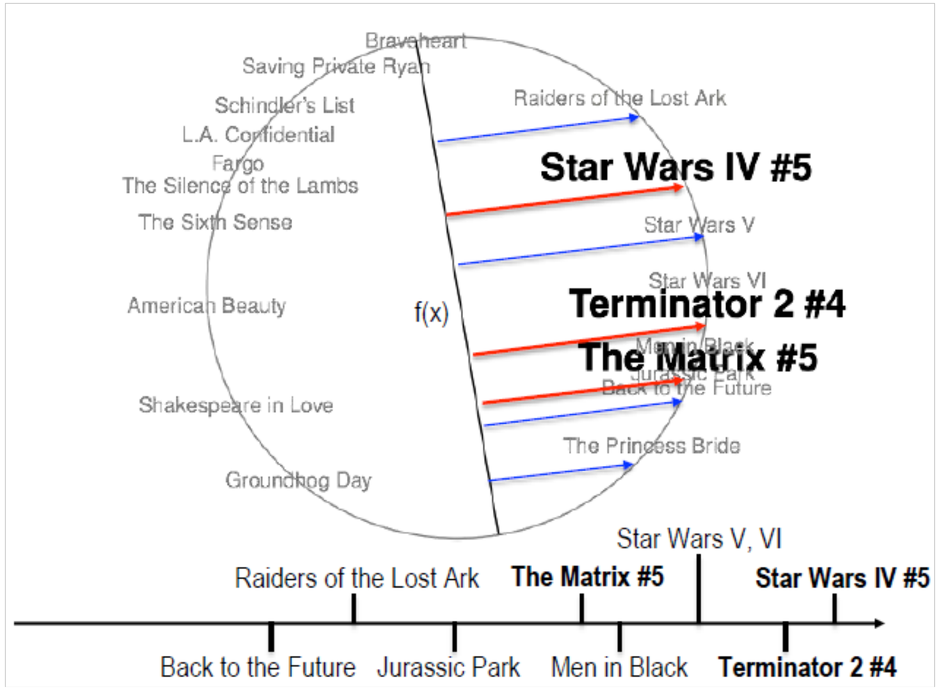
2. Semantic Spaces for Recommendation System

Ranking items

To recommend items to a specific use:



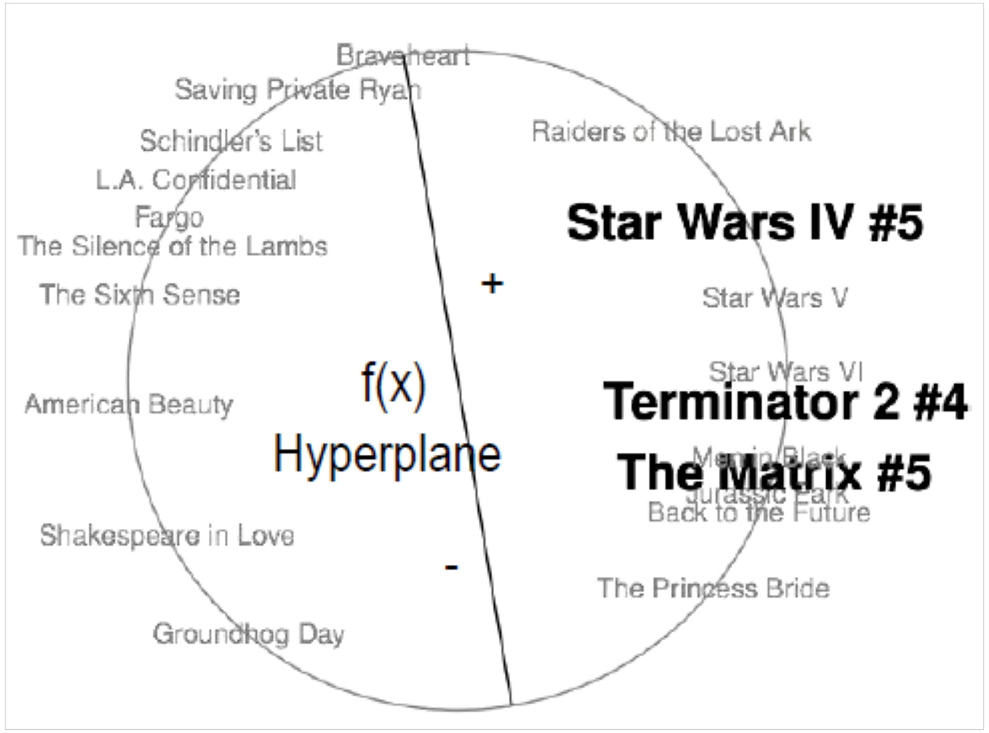




2. Semantic Spaces for Recommendation System

Ranking items

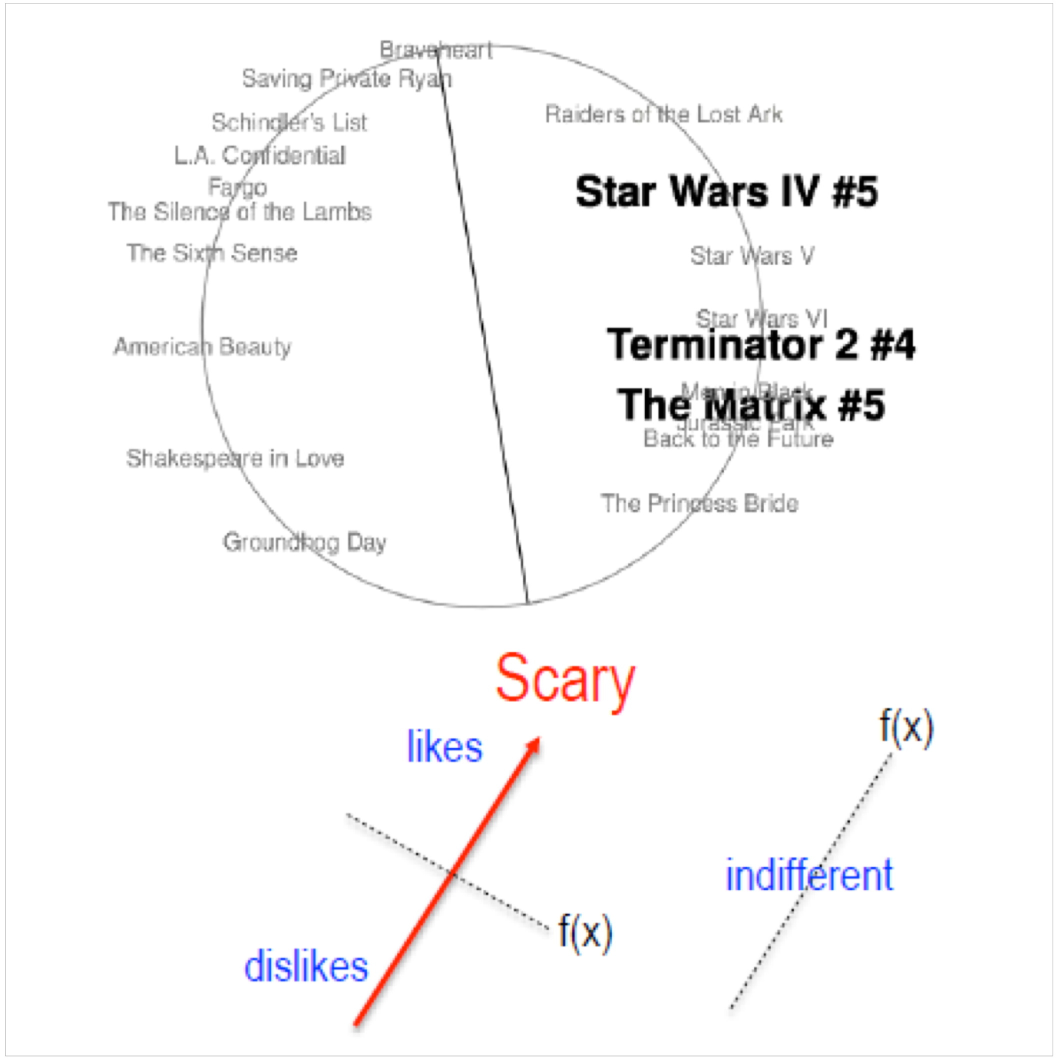
Hyperplane



2. Semantic Spaces for Recommendation System

Ranking items

Hyperplane



3. Implementation: Learning Vector

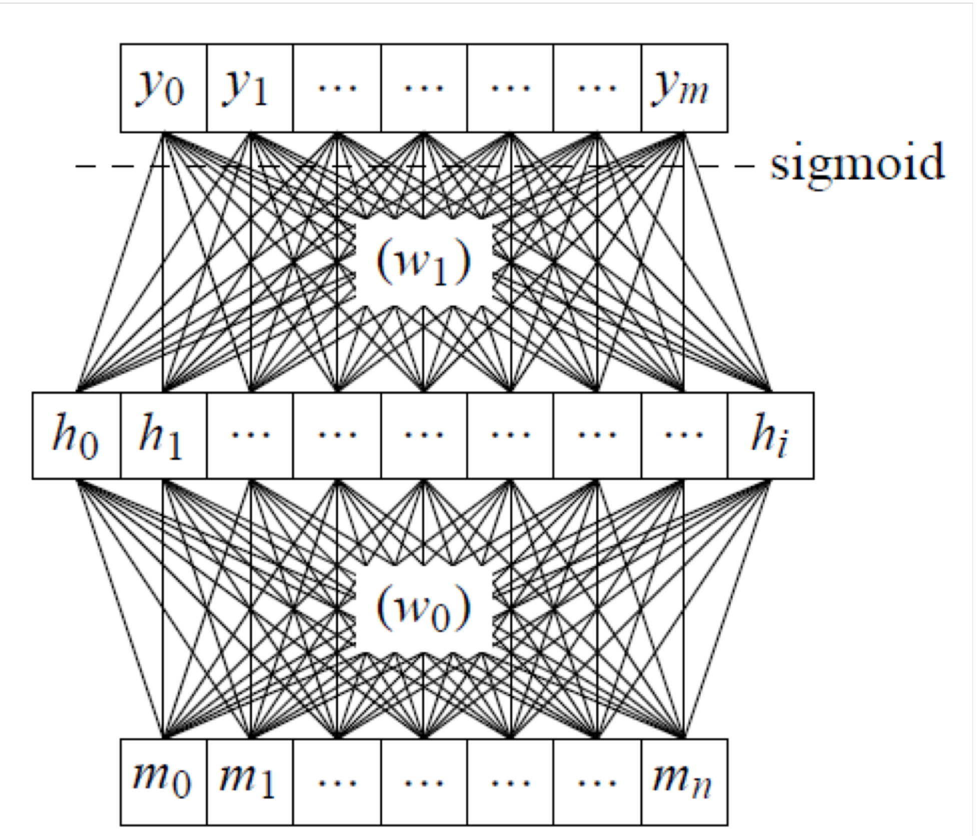
Paragraph Vector

1. The input (bottom) is a '1-hot lookup' vector
    - contains as many nodes as there are items
    - for every training sample only has the node that corresponds to the movie ID set to 1
    - While the other nodes are set to zero
    - which effectively looks up an embedding

    for a given movie *m* in weight matrix w_0

2. Places it in the hidden layer (middle).

3. The output layer
    - contains a node *y* for every possible observation in the training samples.
    - The weight matrices *w*0 and *w*1 respectively connect all possible input nodes, hidden nodes and output nodes.



Paragraph Vector

3. Implementation: Learning Vector

- We learn the embedding by predicting the outputs in a hierarchical softmax

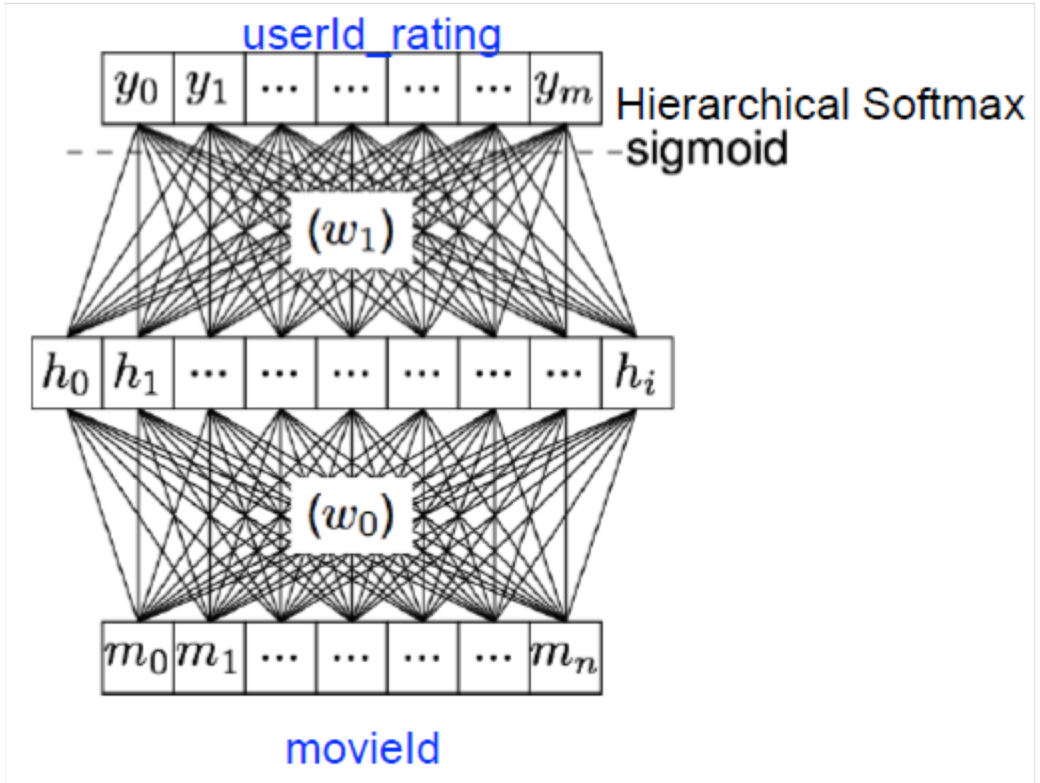
- The input is transformed so that every observation becomes a single word

- Rating less than average rating1; equal or above is rating2

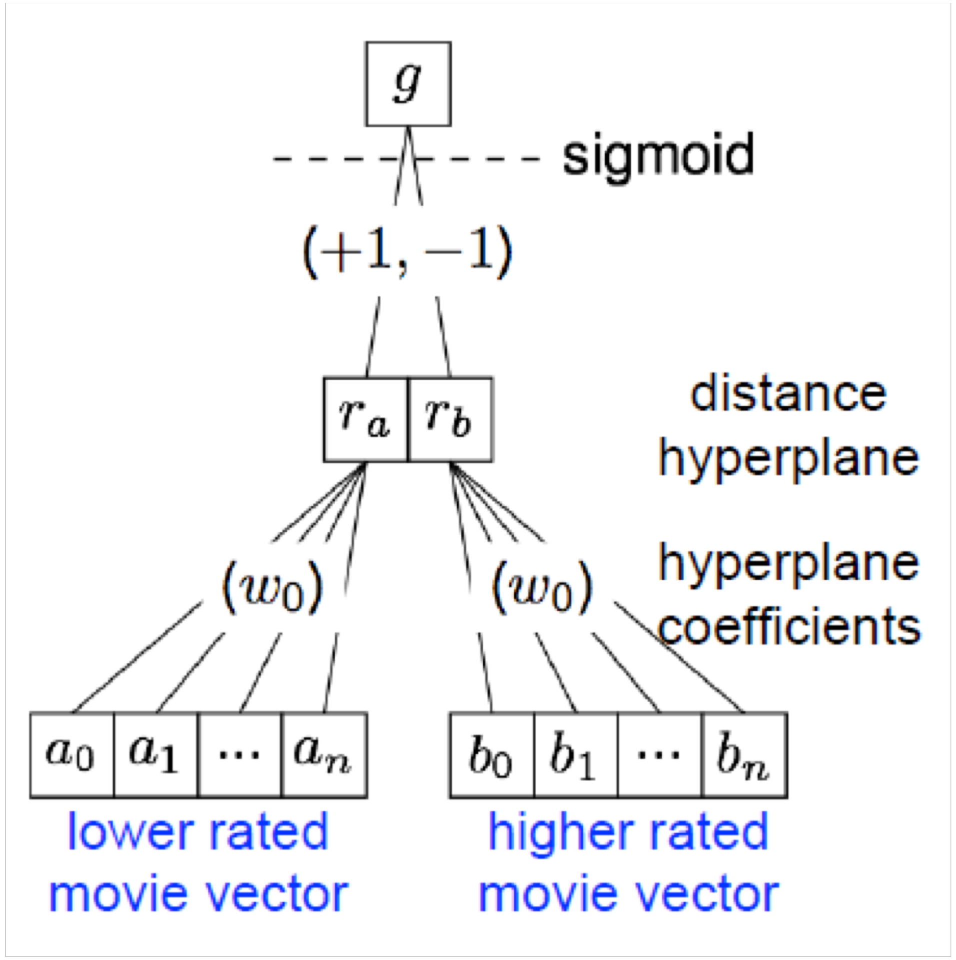
- For example: Star Wars IV, which has id 240 and average rate 3.4; User with id 73 give rate 3

  => (240, ‘user73_rating1’)

- In a content-based setting a review fragment that contains “The masterpiece, the legend that made people...” is transformed into
(240, ‘the’), (240, ‘masterpiece’), (240, ‘the’), etc..



3. Implementation: Ranking items



Update W_0

$$W_0 = W_0 + \propto . (g . b - g . a)$$

Where $\propto = 0.025 \; to \; 0 \; is \; learning \; rate$

4. Experiment

- MovieLens 1M (3952 Users, 6040 Movies)
- Rating with 5-point scale

- For content-based experiments:
- IMDB’s user review (without username or their rating)

- Order user by their number of rating and time they submitted

- The effective of their recommendation system:
  using Recall@10 metric for ratings >=4 (~10k ratings)

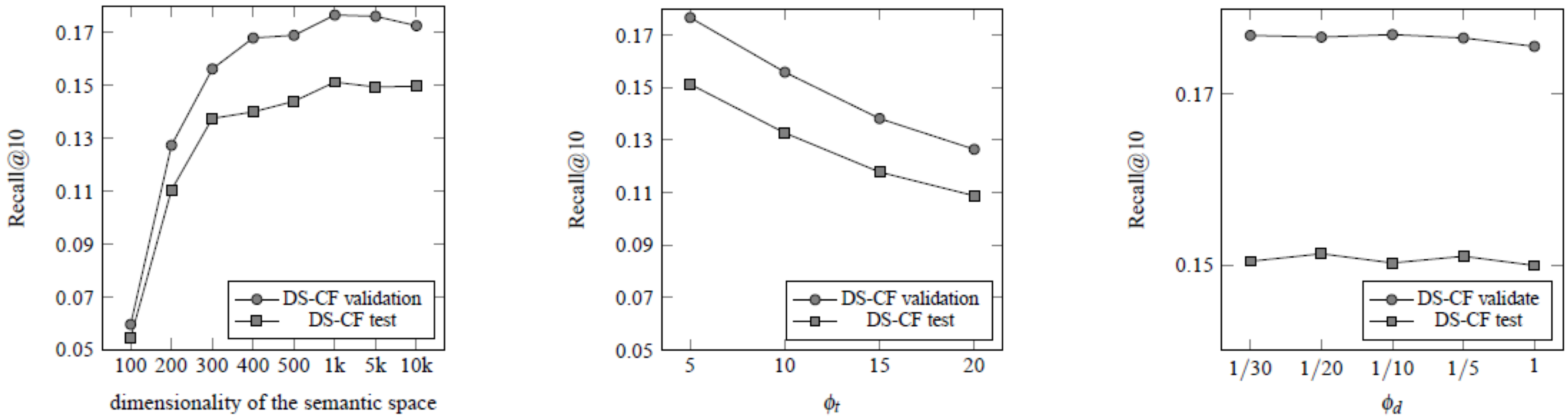
- Compare against BPRMF, WRMF and UserKNN

5. Result

Table 2: Comparison of the effectiveness on MovieLens 1M. The subscripts in the column "sig. over" correspond to a significant improvement over the corresponding system, tested using McNemar test, 1-tailed, *p-value* < 0.001.

System	Recall@10	sig. over
Pop	0.053	
BPRMF [1]	0.079	4
UserKNN [2]	0.087	4
WMRF [3]	0.089	4
DS-CB-10k [4]	0.075	
DS-VSM [5]	0.119	1,2,3,4
DS-CF-500	0.144	1,2,3,4,5
DS-CF-1k	0.151	1,2,3,4,5

6. CONCLUSION



(a) The effect that the dimensionality has on effectiveness.

(b) The effect that using only the number most recently rated movies has on effectiveness.

(c) The effect that downsampling the use of unrated items has on effectiveness.

Figure 4: Sensitivity of hyperparameters

THANK YOU FOR LISTENING