Meta Level Hybrid Recommender System Algorithms

How Graphs can be used to improve performance of recommender systems

Mauriana Pesaresi PhD Seminars
May 18th, 2020

Asma Sattar
PhD Student in Computer Science
asma.sattar@phd.unipi.it

Department of Computer Science
University Of Pisa

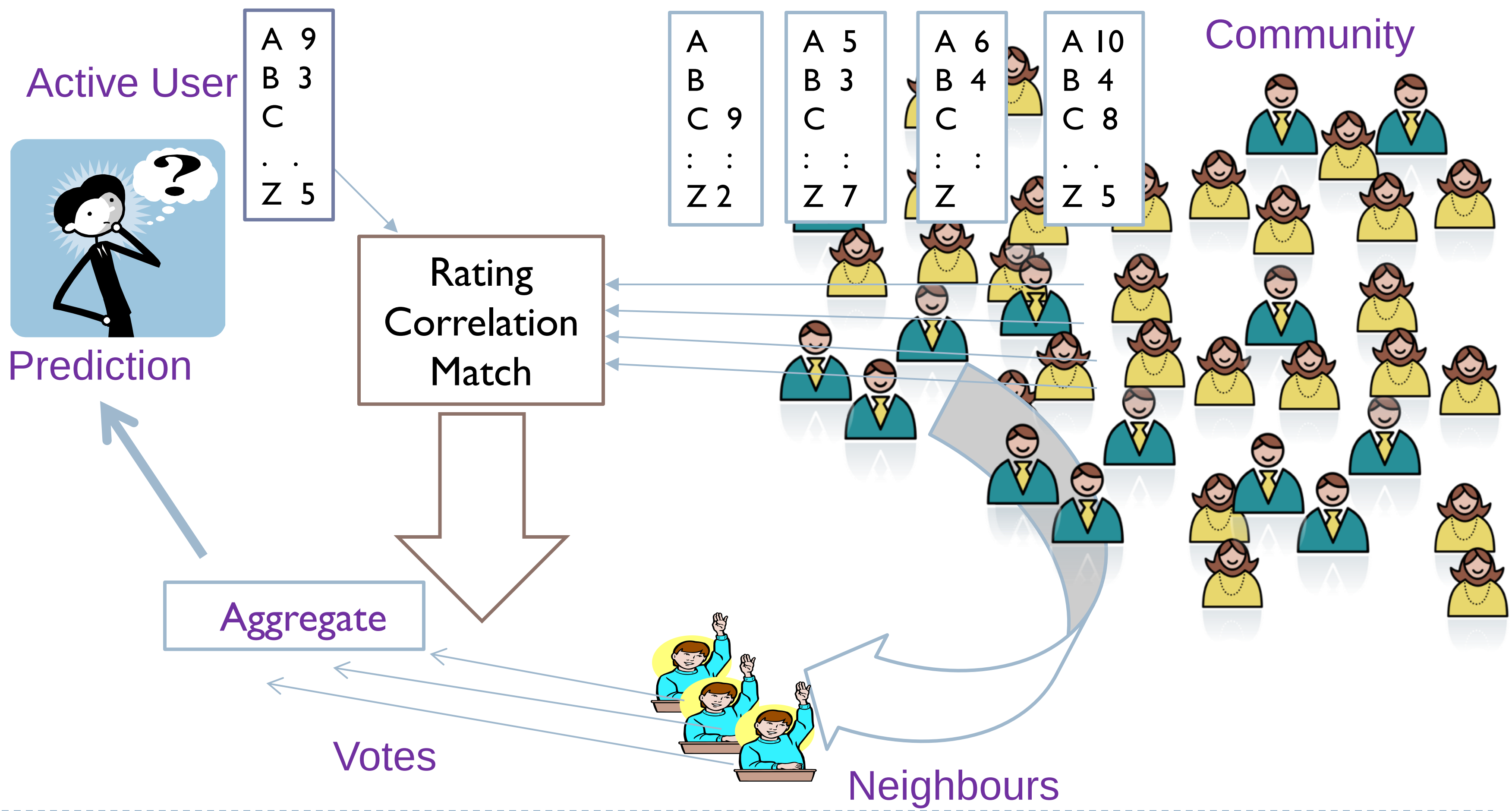
Outline

- Recommender System
- Types of Recommender Systems
- Meta level hybrid Recommender system
- Recommender Systems and Graphs
- Future of Graph based Recommender systems
- References

Recommender Systems

- Information filtering systems that make recommendations on items based on a model of user preferences.
- Key elements are users, items, and rating matrix
- Examples

Collaborative Filtering



Collaborative Filtering

- Identifies the taste of users and suggests the items based on preferences of users with similar taste in those resources.

- Memory based CF
  - Item based CF
  - User based CF

Content Based Filtering (Machine Learning)

Recommendation are generated by **matching the features stored in the user** profile **with those describing the items to be** recommended.

User Profile

Item Profile

Content Based Filtering (Machine Learning)

- Recommends items based on a correlation between the content of the items and a user profile.

- Examples
  - Naïve Bayes Classifier
  - Support Vector Machines Classifier

Motivation

- Availability of vast amount of choices for consumers
- Recommender systems hold the key access to big data.
- To provide intelligent recommendations to consumers.
- Businesses stand to profit if useful recommendations are provided
- Retailers need to retain customer interest
- Netflix reports that at least 75% of their downloads come from their RS, thus making it of strategic importance to the company

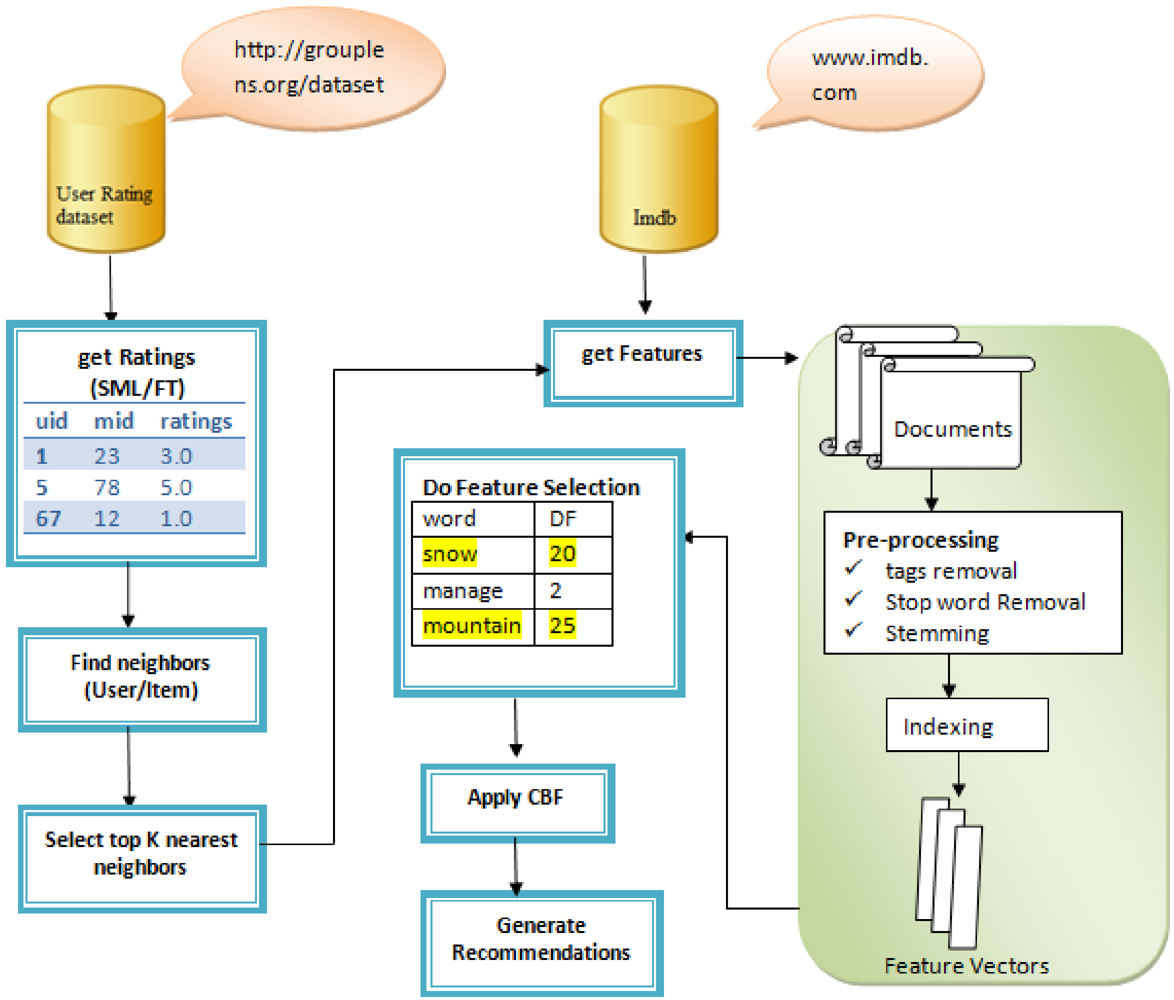
Meta level hybrid Recommender system

Need of hybrid algorithm for accurate recommendation.

- Cold start and sparsity problems in CF
- CF-based algorithms ignoring the feature about items.
- Feature Selection

Framework of Proposed meta level Hybrid Algorithm

$$sim(u,v) = \frac{\sum_{c \in I_{uv}} (R_{u,c} - \overline{R_u})(R_{v,c} - \overline{R_v})}{\sqrt{\sum_{c \in I_u} (R_{u,c} - \overline{R_u})^2}\sqrt{\sum_{c \in I_v} (R_{v,c} - \overline{R_v})^2}}$$



Implementation

- Environment
  - Eclipse(Java)
  - SQL server
- Datasets
  - **For Ratings :**
    - MoviesLens [1] & FilmTrust [2]
  - **For Features :**
    - Internet movie database (imdb) [3]
  - Divided in five folds (four training, one testing)
- Metric
  - MAE

Implementation

- Find optimal number of neighbors of a target item using adjusted cosine similarity
- Crawl features of these neighbor items from imdb
- Preprocessing of features
  - Tag Removal
  - Stop word Removal[4]
  - Stemming using Porter stemmer algorithm [5]
- Use TF-IDF approach to represent Features
- Apply feature selection technique (TF and DF Thresholding)
- Build CBF model over selected features of items
- Use trained model to predict rating of target item

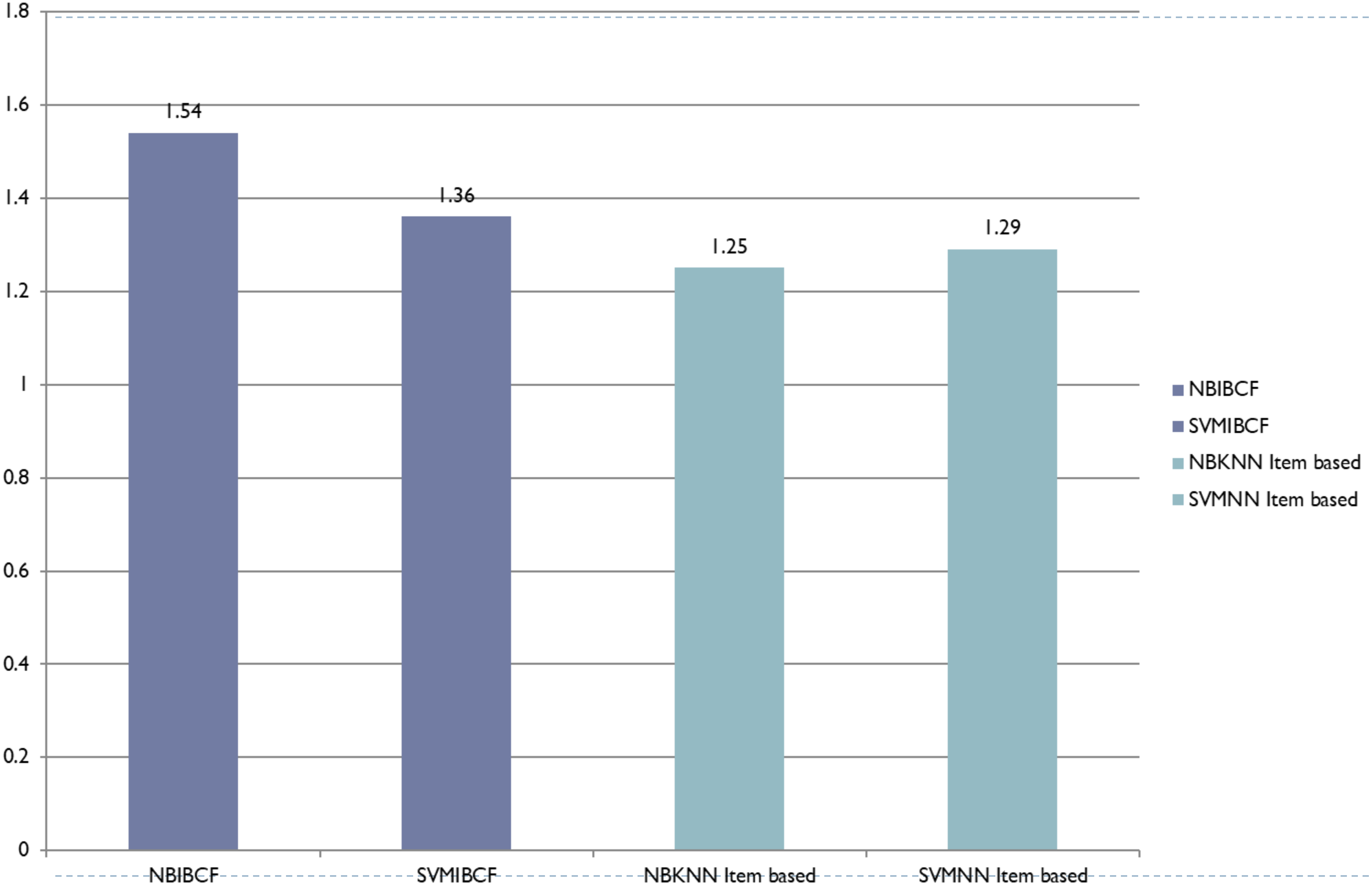
Implementation

- Use MAE to evaluate difference in predicted and actual target item's rating.
- Create scenarios like cold start user, cold start item, skewed/sparse dataset and evaluate performance of proposed algorithm

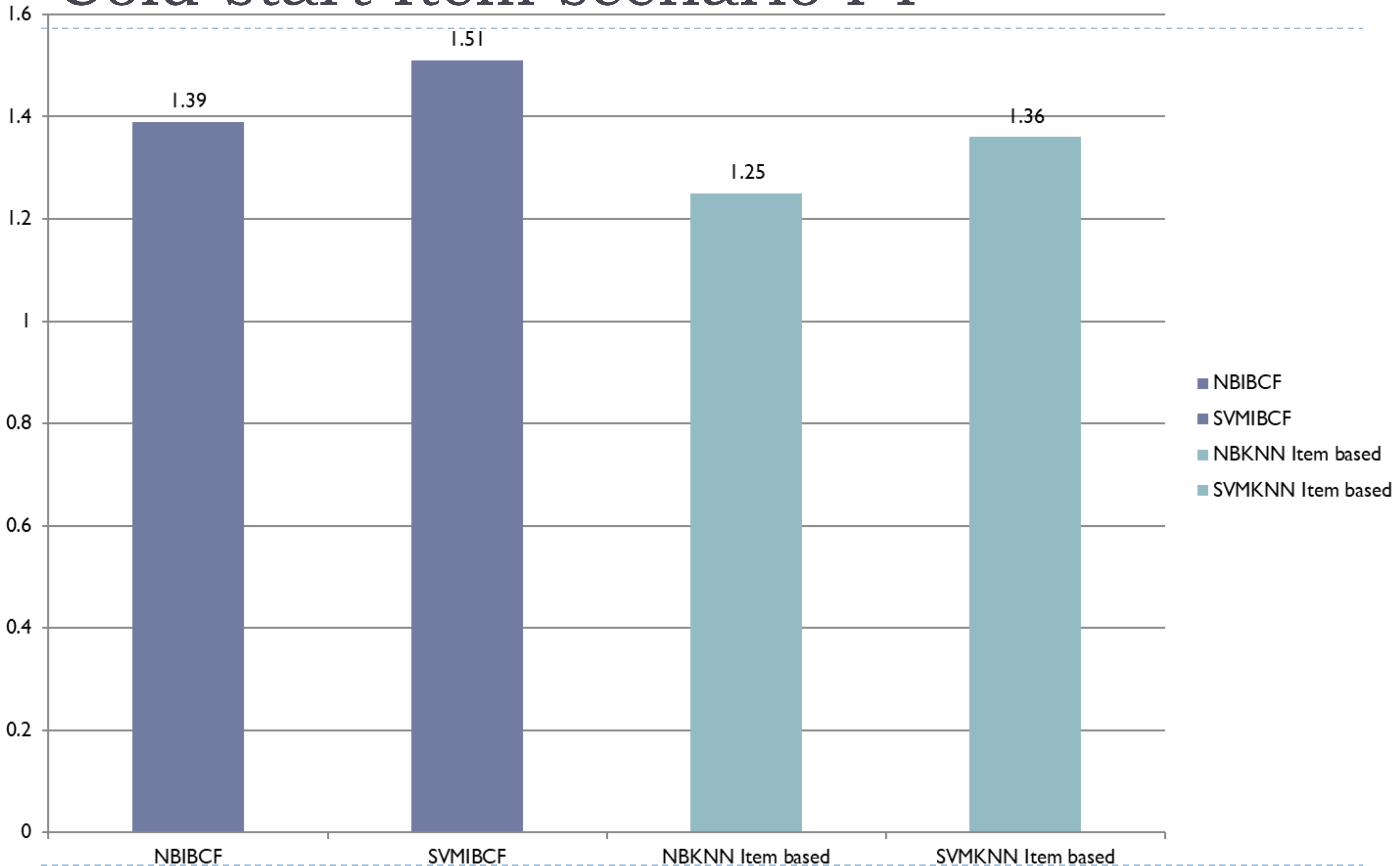
Results

- Comparison with Naïve Hybrid approaches
  - Results under cold start user scenario
  - Results under cold start item scenario
  - Results for sparse dataset
- Benchmark Results

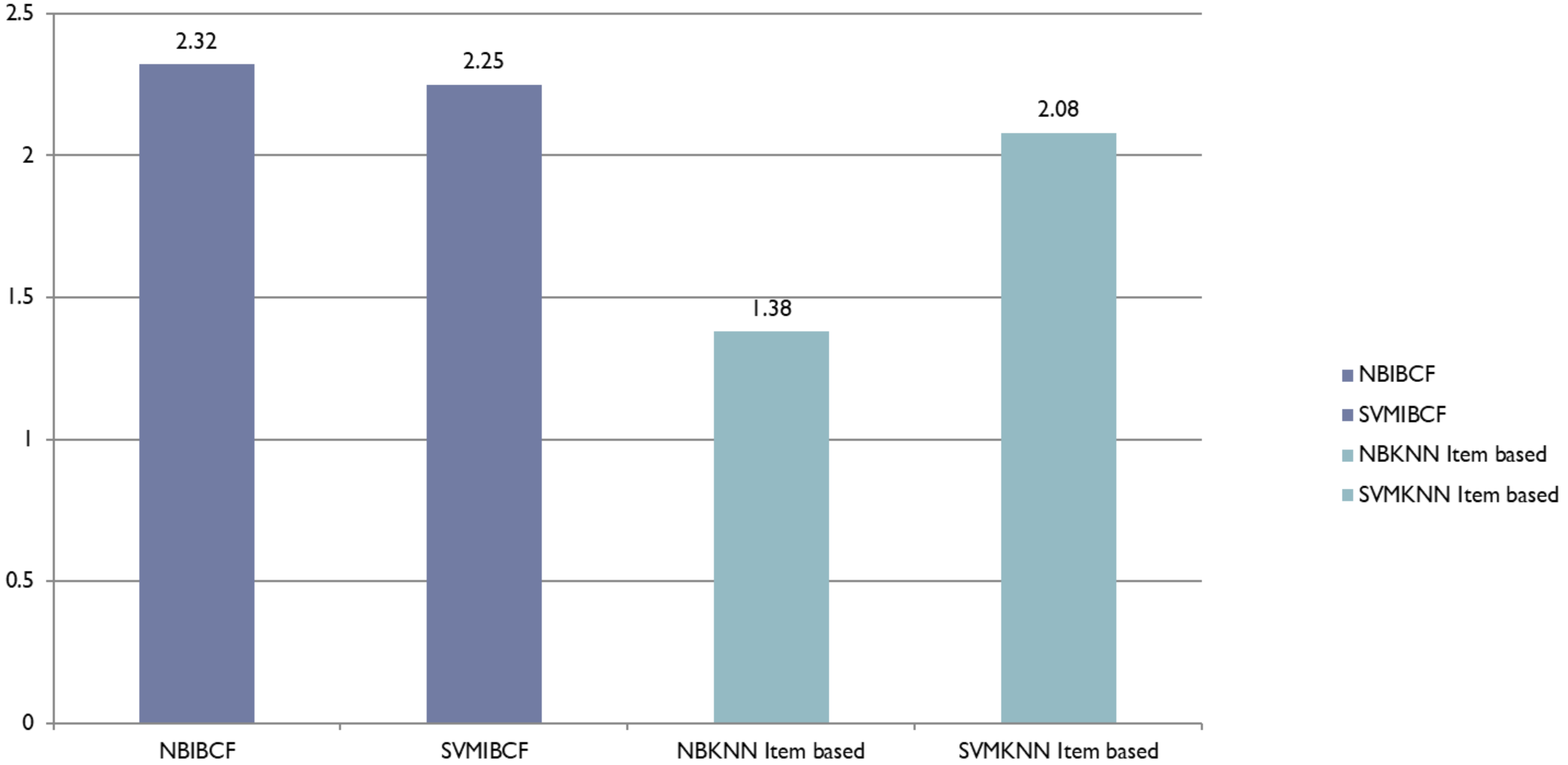
Cold start User scenario FT



Cold start Item scenario FT



Sparsity FT



2.5
2
1.5
1
0.5
0
2.32
2.25
1.38
2.08
NBIBCF
SVMIBCF
NBKNN Item based
SVMKNN Item based
NBIBCF
SVMIBCF
NBKNN Item based
SVMKNN Item based

Benchmark Results (FT)

Cold Start user Scenario		MAE
Our Best Approach	**NBKNN Item based**	**1.25**
Literature Approaches	NBIBCF	1.54
	Switching NBCF [6]	1.53
Cold Start Item Scenario		
Our Best Approach	**NBKNN Item based**	**1.26**
Literature Approaches	NBIBCF	1.39
	Switching NBCF [6]	1.30
Sparsity Scenario		
Our Best Approach	**NBKNN Item based**	**1.38**
Literature Approaches	NBIBCF	2.32
	Switching NBCF [6]	2.01

What we concluded after this research work?

- Hybrid approaches perform better than individual techniques used for recommendation
- Producing good results for imbalanced datasets and under cold start scenarios
- Careful selection of appropriate approaches can produce accurate recommendation under different scenarios.

Recommender Systems and Graph

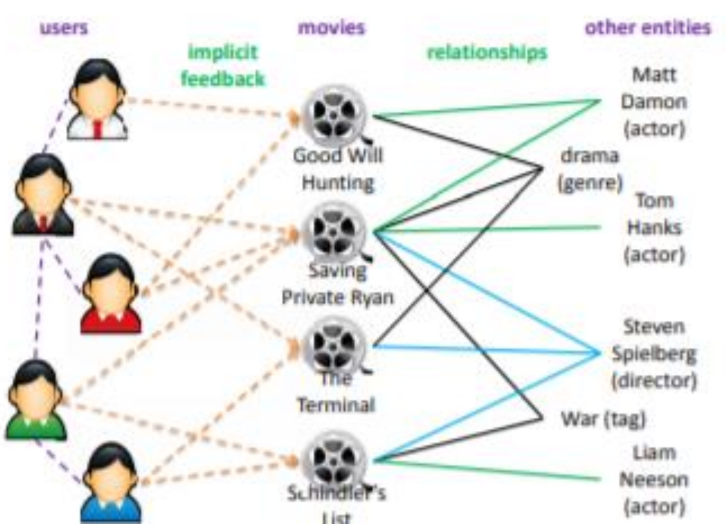
- Recommendation systems task can be reduced to a matrix completion task
- Traditionally, recommender systems are built on a CF or CBF to a matrix completion task
- Undirected bipartite user-item graph can be used to represent recommender system
- Representation of user and item data in separate user and item graphs.
- Clearly, graph-structured data arises naturally in the recommendation task

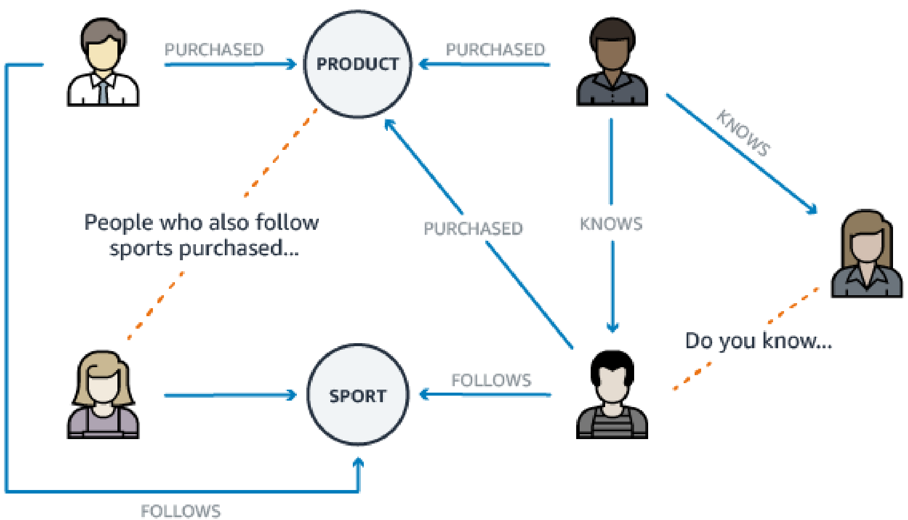
Future Research Direction

- Handling Heterogeneous Graph
- Handling multiplex networks
- Node Classification and Link Prediction in Heterogeneous Graph
- Dealing with Dynamic Graph
- Learning from Contextual information

Handling Heterogeneous Graph

- Graphs that contain different types of nodes and edges
- Different types of nodes and edges tend to have different types of attributes that are designed to capture the characteristics of each node and edge type





Handling multiplex networks

- Two or more separate graphs contain information for the same nodes, and for which we want to do some multiplex network analysis.
- To exploit transferring knowledge from different graphs can improve recommendation accuracy
- An interesting research direction would be to analyze problem settings with more than one graph.

Node Classification and Link Prediction in Heterogeneous Graph

- Node Classification: capturing aspects of an individual's preferences or behavior
  - demographic labels : , such as age, gender and location
  - Encode Interests : hobbies, and affiliations
  - Can be contextual information in our case (Weather, mood, Day of the week etc)

- Suggesting new connections or contacts to individuals, based on finding others with similar interests, demographics, or experiences.
- Work over generalized graph structures, such as hypergraphs, graphs with weighted, labeled, or timestamped edges, multiple edges between nodes, and so on.

Node Classification and Link Prediction in Heterogeneous Graph

- In a link prediction problem, all nodes are observed, but random entries of adjacency matrix/list A are missing.
- The problem objective is to predict the missing edges to complete the adjacency matrix A, based on the feature vectors and the known graph structure of all the nodes.

Dealing with Dynamic Graph with respect to context

- Recommendation in online communities is a challenging problem
  - Majority of research work done in field of recommender system involves static preferences of user
  - Users' interests are dynamic
  - User interest are influenced by social events
  - capture the user's rapidly-changing interests
  - To produce meaningful recommendations by using contextual user-item rating information.

Learning from Contextual information

- A context is a vast term that may consider various aspects.
- Current algorithms is good at matching the users' preferences and the recommendations, gives a good mix of familiar and new options but the recommendations can however still be perceived as poor. Example: Restaurant recommendation
- Example: time, mood, location, weather, company, day type, an item's genre, location, and language.
- Typically, the rating behavior of users varies under different contexts.

Learning from Contextual information

Contextual variable	Description
time	morning, afternoon, evening, night
daytype	working day, weekend, holiday
season	spring, summer, autumn, winter
location	home, public place, friend's house
weather	sunny/clear, rainy, stormy, snowy, cloudy
social	alone, partner, friends, colleagues, parents, public, family
mood	positive, neutral, negative
physical	healthy, ill
decision	user picked the item, item suggested by other
interaction	first, n-th

Datasets (tentative)

- Datsets that are mostly used in recommender systems:
  - Movielens
  - Filmtrust
  - Amazon
- Dataset with Contextual Information:
  - LDOS Comoda
  - DePaulMovi
- Dataset for Heterogeneous/hypergraphs
  - IMDB
  - Yelp
  - ….

Tools/Libraries (Tentative)

- Language:
  - Python
- Tool:
  - PyCharm
  - Jupiter Notebook
  - VScode
- Libraries:
  - PyG
  - DGL

Conclusion

- What we want to achieve:
  - Represent heterogenous information for recommender systems using graph structure
  - Exploit information from hypergraphs in best way
  - Learning which contextual information is improving accuracy of recommender systems
  - Learn from the dynamic Graphs thus learning dynamic interest of users
  - Apply deep learning graph algorithms on hypergraphs to find useful recommendation with good accuracy

References

1. MovieLens 100k, available on-line at http://grouplens.org/datasets/movielens.
2. FILM TRUST, available on-line at http://www.filmtrust.net
3. Internet Movie Database online at www.imdb.com
4. Google Stop words list available at ranks.nl/resources/stopwords.html
5. Porter stemming algorithm http://tartarus.org/martin/PorterStemmer/def.txt
6. Mustansar Ali Ghazanfar, " Robust, Scalable, and Practical Algorithms for Recommender Systems", 2012
7. Sattar, Asma, Mustansar Ali Ghazanfar, and Misbah Iqbal. "Building accurate and practical recommender system algorithms using machine learning classifier and collaborative filtering." Arabian Journal for Science and Engineering 42.8 (2017): 3229-3247
8. Cummings, David, and Ningxuan Jason Wang. "Network-based recommendation: Using graph structure in user-product rating networks to generate product recommendations."
9. Derrick Mwiti, How to build a Simple Recommender System in Python (https://towardsdatascience.com/how-to-build-a-simple-recommender-system-in-python-375093c3fb7d)
10. Berg, Rianne van den, Thomas N. Kipf, and Max Welling. "Graph convolutional matrix completion." arXiv preprint arXiv:1706.02263 (2017).
11. Zhang, Muhan, and Yixin Chen. "Inductive matrix completion based on graph neural networks." arXiv preprint arXiv:1904.12058 (2019).


QUESTIONS?