HAROKOPIO UNIVERSITY of ATHENS
Department of Informatics & Telematics

Scaling Collaborative Filtering to large–scale Bipartite Rating Graphs using Lenskit and Spark

Sardianos Christos, Iraklis Varlamis, Magdalini Eirinaki

sardianos@hua.gr

IEEE BigDataService 2017

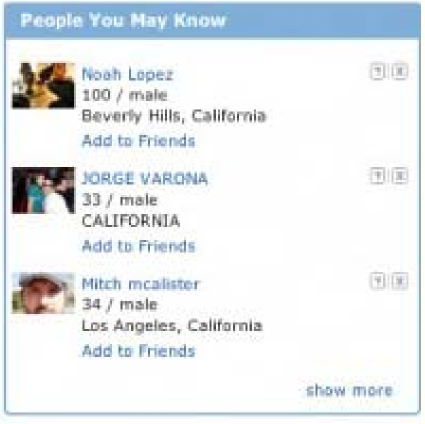
Role of Recommender Systems

- ❑ In many Web 2.0 applications users can interact with the applications in terms of social activity.
  - ❑ They can express their trust for another user or another user’s review.

Social Recommender Systems (RS)

- In the Social Web

MySpace



People You May Know

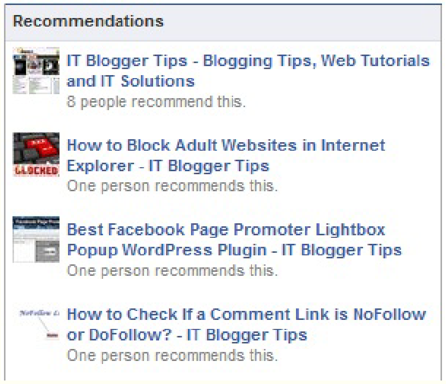
Noah Lopez
100 / male
Beverly Hills, California
Add to Friends

JORGE VARONA
33 / male
CALIFORNIA
Add to Friends

Mitch mcalister
34 / male
Los Angeles, California
Add to Friends

show more

Facebook



Recommendations

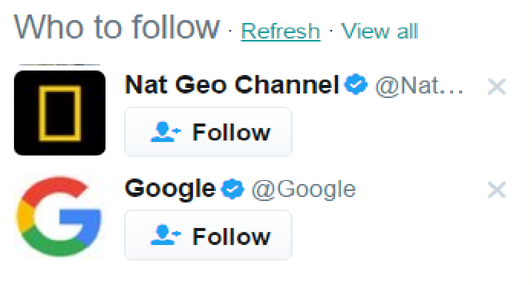
IT Blogger Tips - Blogging Tips, Web Tutorials and IT Solutions
8 people recommend this.

How to Block Adult Websites in Internet Explorer - IT Blogger Tips
One person recommends this.

Best Facebook Page Promoter Lightbox Popup WordPress Plugin - IT Blogger Tips
One person recommends this.

How to Check If a Comment Link is NoFollow or DoFollow? - IT Blogger Tips
One person recommends this.

Twitter



Who to follow · Refresh · View all

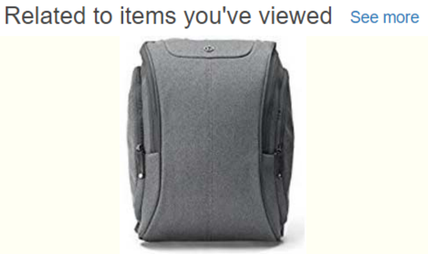
Nat Geo Channel @Nat…
Follow

Google @Google
Follow

Item/Content Recommender Systems (RS)

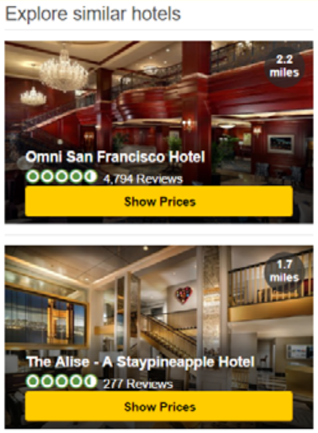
- Application areas

Amazon



Related to items you've viewed See more

Trip Advisor



Explore similar hotels

2.2 miles

Omni San Francisco Hotel

4,794 Reviews

Show Prices

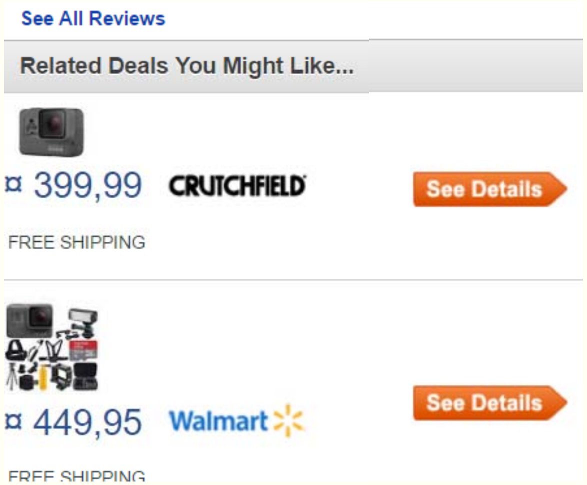
1.7 miles

The Alise - A Staypineapple Hotel

277 Reviews

Show Prices

Epinions



See All Reviews

Related Deals You Might Like...

¤ 399,99 CRUTCHFIELD

See Details

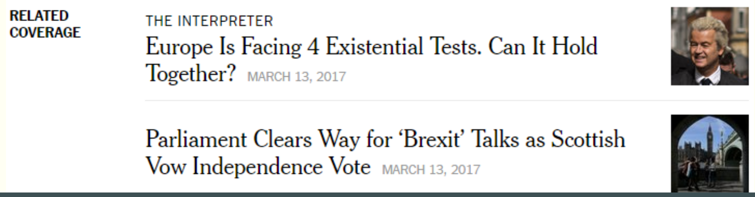
FREE SHIPPING

¤ 449,95 Walmart

See Details

FREE SHIPPING

The New York Times



RELATED COVERAGE

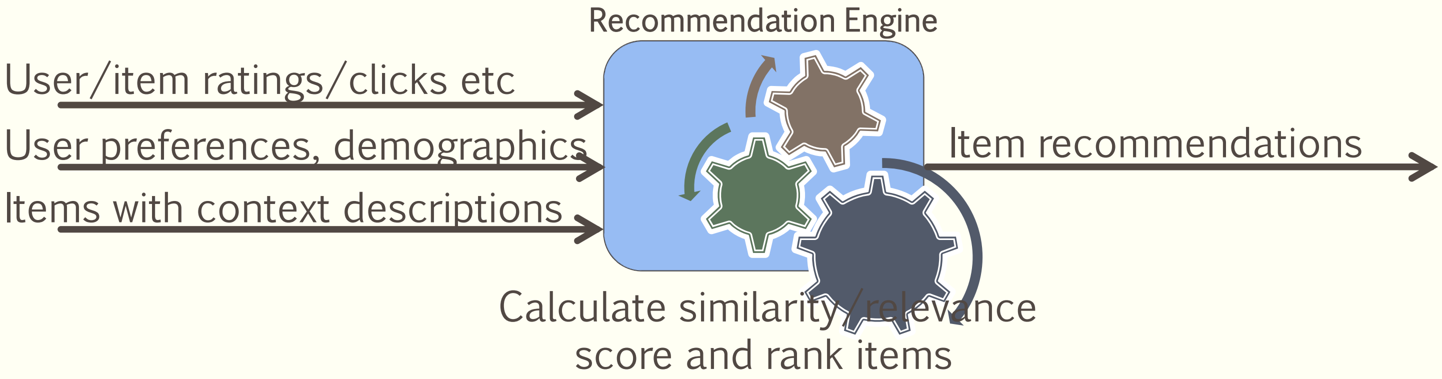
THE INTERPRETER

Europe Is Facing 4 Existential Tests. Can It Hold Together? MARCH 13, 2017

Parliament Clears Way for 'Brexit' Talks as Scottish Vow Independence Vote MARCH 13, 2017

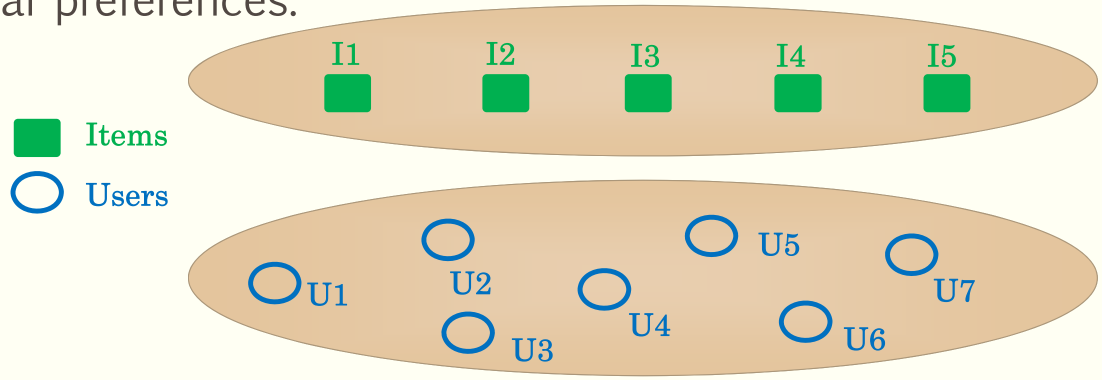
Role of Recommender Systems

- ❑ An item recommender system
  - ❑ operates on data related to the behavior of a set of users and
  - ❑ is responsible for recommending items (e.g. products, articles etc.) to users, based on their previous activity.


Recommendation Engine
User/item ratings/clicks etc
User preferences, demographics
Items with context descriptions
Item recommendations
Calculate similarity/relevance
score and rank items

Collaborative Filtering (CF)

- Fundamental approach for recommending "similar" items.
  - Used by some of the biggest e-commerce sites
- Takes advantage of the "wisdom of crowds".
- Users evaluate the available items (usually in a numerical scale).
- User-Item ratings information is represented as a bipartite graph.
- Recommendations are based on collaboration of multiple users and filtered on those with similar preferences.



Current status in CF

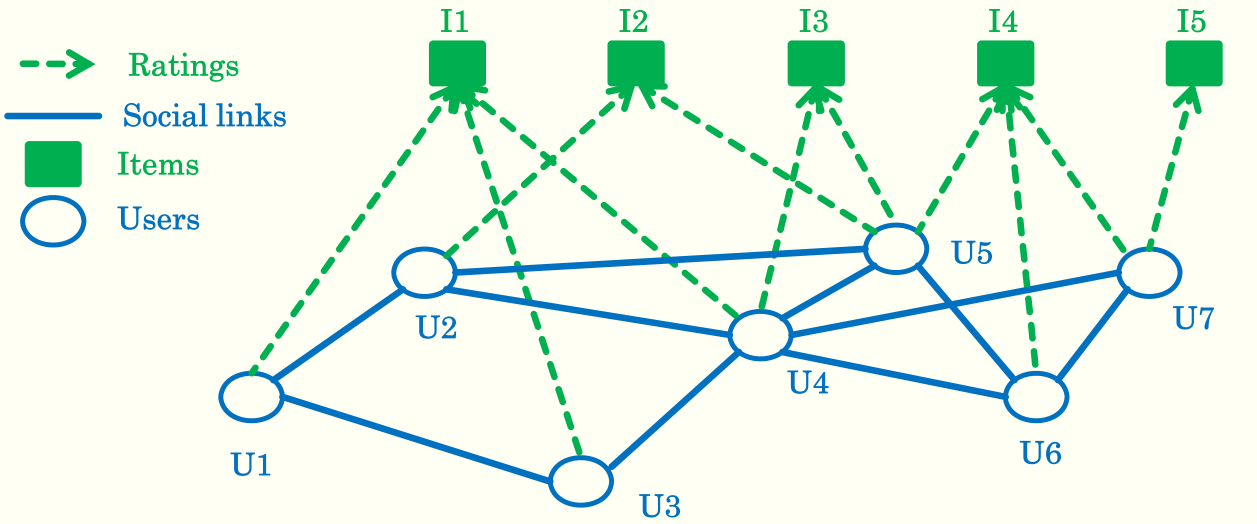
- Large-scale social networks: too many users ➔ generate very large graphs with different characteristics.
- In product-rating graphs, users connect with each other and rate items in tandem.
- In such bipartite graphs users and items are the nodes and ratings are the edges.
- Collaborative Filtering algorithms use these edges to suggest items of potential interest to users.

PROBLEM

- Existing algorithms can hardly scale up to the size of the entire product rating graph and require unlimited resources to finish.

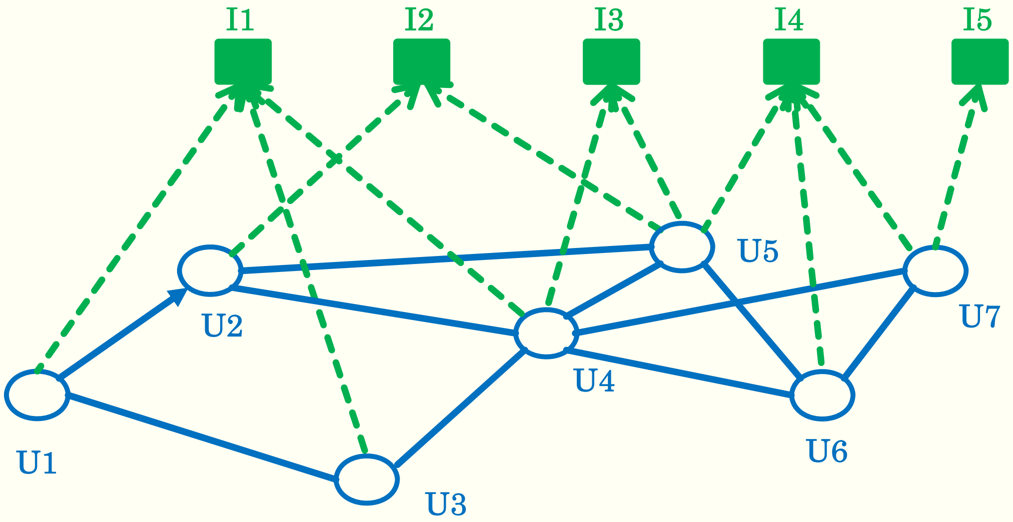
The case & Motivation

- The case: Recommender systems for social networks that combine bi-partite (e.g. item ratings) and uni-partite graphs (e.g. user social links).
- Motivation: Take advantage of the structure of social networks in order to scale-down the CF complexity. Use less memory, use parallel processing.



Scaling CF to large graphs: Our approach

- Collaborative filtering algorithms cannot perform on the complete graph
- Partition the graph into smaller subgraphs → increase performance without loosing in quality



Feasibility study

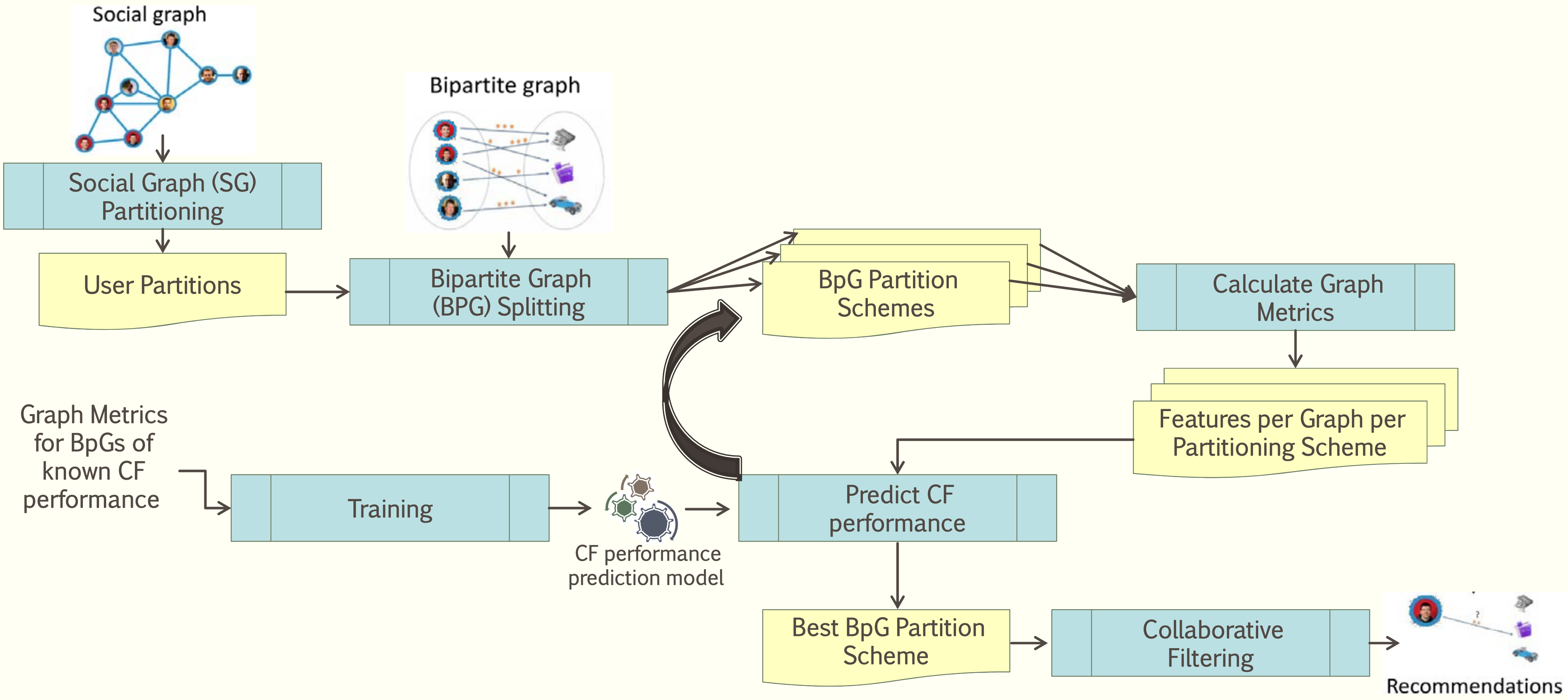
- Graph partitioning is a fast process.
- CF is the bottleneck for large graphs (slow and resource-demanding).
- If we can predict the performance of the CF algorithm in every partition beforehand,
- We can decide on the optimal partitioning scheme.

Q: Is it possible to find the best number of partitions by simply examining some features of the different schemes?

Finding the best graph partitioning scheme for CF

- The quality of CF recommendation is strongly related to the number of items that users have rated in common (user rating overlap).
- We extract several structural features for each subgraph in each partitioning scheme, in order to capture user rating overlap.
- We train a supervised model, to predict the quality of recommendations.

Proposed recommender system workflow



Social graph

Social Graph (SG) Partitioning

User Partitions

Bipartite graph

Bipartite Graph (BPG) Splitting

BpG Partition Schemes

Calculate Graph Metrics

Features per Graph per Partitioning Scheme

Graph Metrics for BpGs of known CF performance

Training

CF performance prediction model

Predict CF performance

Best BpG Partition Scheme

Collaborative Filtering

Recommendations

Structural features of bipartite graph partitions

- We measure the overlap between user provided ratings.

- We also measure node related features that capture node centrality.

centrality

- We have employed two types of graph metrics:
  1. Graph specific metrics: give a single value (or set of values) for the subgraph as a whole
  2. Node specific metrics: give a different value for each node
- For the node specific metrics, we employ the percentile values for the 10 percentile levels computed on the set of vertices of the graph
- We modified some features that were not currently defined for bipartite graphs and adapted them in our model.

Choosing our features

Graph specific features

OVERLAP

- Sparsity
- Gini
- Entropy

Node specific features

centrality

- Degree centrality
- Eigenvector centrality
- Pagerank centrality

OVERLAP

- Triads
- Clustering coefficient
- Redundancy coefficient

Experimental Setup

- To evaluate our approach we used the Epinions dataset.
- Epinions is one of the largest consumer product reviews site.

Graph Characteristics		
Social Graph	Num. of Distinct Users	132 Thousand
	Num. of Social Edges	842 Thousand
	Average Degree	13
Bipartite Graph	Num. of Distinct Users (raters)	121 Thousand
	Num. of Distinct Items	756 Thousand
	Num. of Ratings	14 Millions
	Avg. outDegree/User	114
	Avg. inDegree/Item	18

http://konect.uni-koblenz.de/networks

Experimental Setup

- We train our machine learning model using subgraphs from different partitioning schemes of the original graph.
- We evaluated 64 different partitioning schemes raging from 1500 to 4 partitions. 2000 training instances (graphs) in total.
- We predicted CF performance for the three main CF techniques: user-based, item-based and matrix factorization
- Initially, we evaluated our predictions using only the three Graph Features (i.e. Sparsity, Gini and Entropy). Then we compared with the Node Features. Finally, we compared with the full set of Graph and Node Features.
- For each technique, 2 different CF performance metrics were examined (RMSE and nDCG).
- All experiments were repeated using a random 90%-10% training-test split.
- We use Pearson correlation coefficient to measure the correlation between the actual and the predicted CF recommendation quality.

Pearson correlation between predicted and actual CF performance.

- We used two alternatives to predict the actual performance of CF in each subgraph for every partitioning scheme:
  - Linear Regression with feature selection
  - SMOreg (Sequential Minimal Optimizer for regression)

		3 Feat. (LR)	All feat. (LR)	3 Feat. (SMOreg)	All feat. (SMOreg)
User-User	RMSE	0.29	0.46	0.32	0.47
	RMSE by user	0.16	0.34	0.18	0.36
	nDCG	0.09	0.50	0.27	0.51
Item-Item	RMSE	0.24	0.38	0.27	0.38
	RMSE by user	0.08	0.23	0.08	0.23
	nDCG	0.08	0.50	0.27	0.51
SVD	RMSE	0.24	0.47	0.30	0.48
	RMSE by user	0.16	0.36	0.19	0.39
	nDCG	0.14	0.52	0.29	0.57

Binary classification model evaluation

- We repeated the experiments, training a binary classifier with the values of the RMSE by user metric.
- RandomForest classifier: 90.8% ±0.02 accuracy level (at 99% CI).
- The performance of the SVM classifier reaches 93.5% ±0.03 (at 99% CI).
- We decided to train an SVM classifier on the RMSE metric and evaluate all future partitions of a large bipartite graph using this classifier.

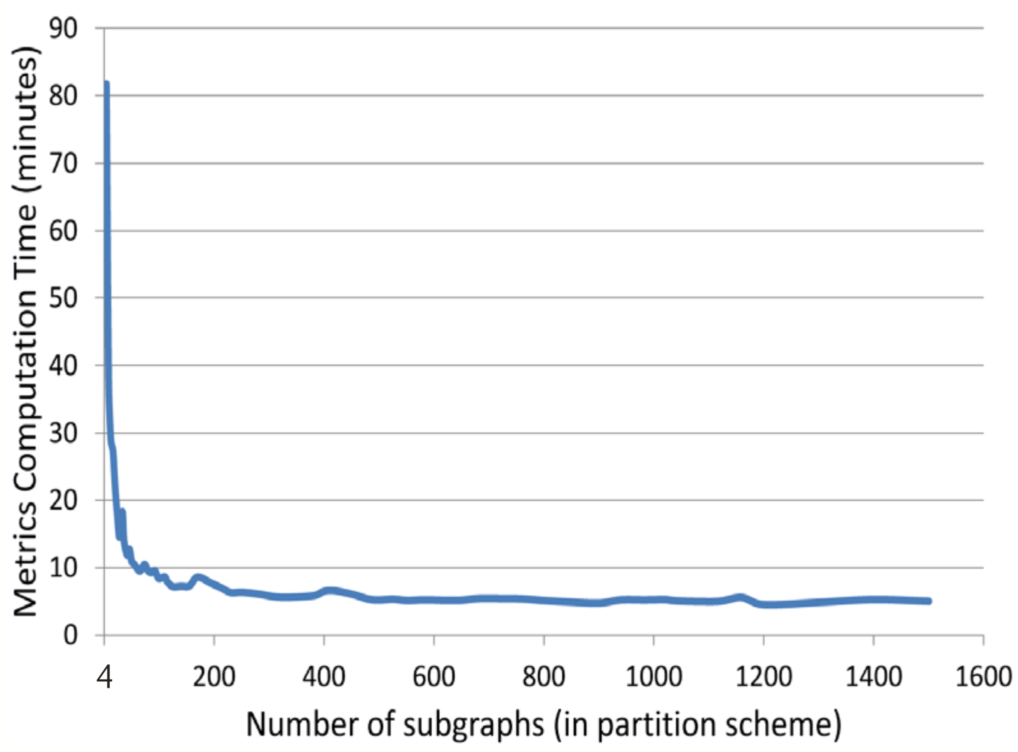
Experimental Setup

- All experiments were performed on a Dell PowerEdge R730 server with 24 CPU cores and 96GB of RAM in total, running Apache Spark.
- The implementation of CF algorithms provided by LensKit was used.
- Despite the large amount of memory available it was not possible to run the three CF algorithms on the whole graph not even on its 2 partitions.
- We split the graph repeatedly to 4 up to 1500 partitions and used the binary classifier to predict CF performance. The model predicted 65-partitions as the best scheme
- ~~At 65-partitions scheme the sequential execution of the three CF algorithms took 2 hours and 5 minutes.~~

Time performance

- The partitioning process, including the execution of Metis 64 times was completed in about 30 seconds (less than 0.5 seconds per partitioning scheme).
- In general, it takes 5 to 10 min for partitioning into subgraphs and testing whether these graphs can produce high quality recommendations.
- The time for predicting CF performance is less than 1sec for any partition in any scheme.
- Parallelizing this process we reduced the total execution time for the 65 partitions to 8 minutes.

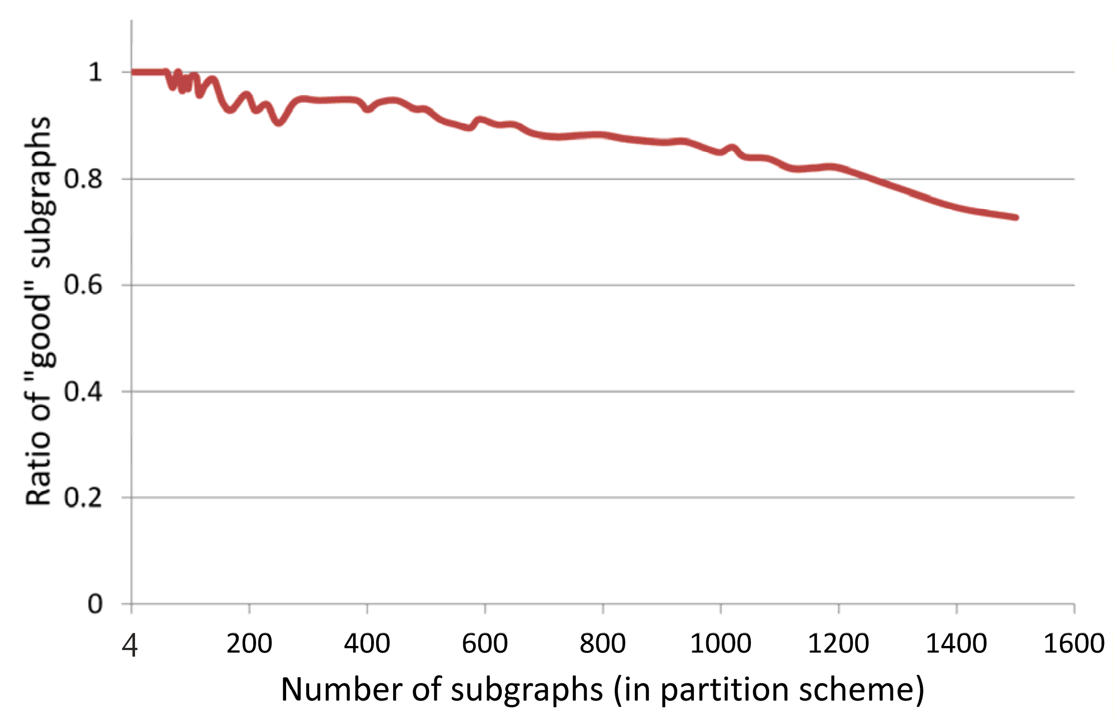
Execution time for every partitioning scheme (from 4 to 1500)



Binary classification outcomes

- Our goal was not to find how good did a CF algorithm performed in a specific dataset, but to identify if quality of CF keeps high as we keep splitting in smaller graphs.
- At 65 subgraphs the "good" graphs ratio is almost 0.97.
- As we keep splitting in more than 65 graphs, the quality of the created graphs decreases.

Ratio of good subgraphs per scheme (from 4 to 1500)



Conclusions

- We presented a methodology for predicting the performance of CF algorithms using the structural features of the bipartite graph.
- The proposed methodology incorporates several metrics that either apply to the whole graph or to each node separately
- Results show that we are able to generate different partitioning schemes, predict the performance of CF in each one of them and select the optimal partitioning scheme.
- Our methodology can handle huge graphs and perform faster if more resources are available to be used in parallel.

Future work

- Test our methodology in different datasets with varying characteristics.
- Improve bad sub-graphs by adding to them users (and their ratings) that rate many items.
- Test different partitioning criteria, as the partitioning process could take into account the structure of the product-rating graph itself.
- A Spark-based recommender system by incorporating our method.

Thank you for your time.