A Bottom-Up Approach to Job Recommendation System

(RecSys Competition 2016)

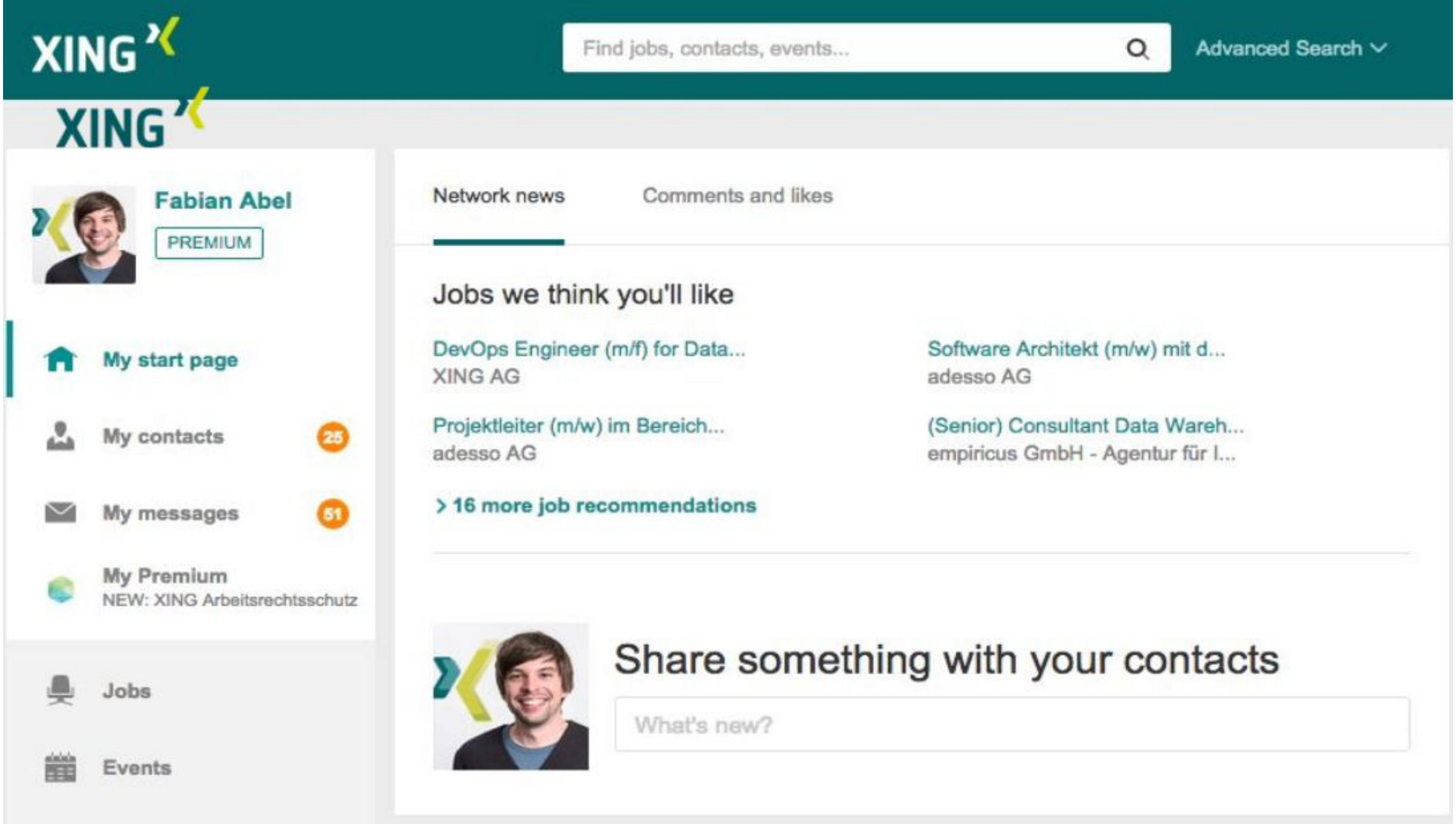
Stoyan Eftimov
Hrizantema Stancheva

Problem

Recommender Systems held during 15-19th September 2016, Boston @ MIT & IBM

RecSys Challenge:

- Build a job recommendation system for XING
- Given a XING user, the goal is to predict those job postings that a user will positively interact with (e.g. click, bookmark)



Jobs we think you'll like

DevOps Engineer (m/f) for Data...
XING AG

Software Architekt (m/w) mit d...
adesso AG

Projektleiter (m/w) im Bereich...
adesso AG

(Senior) Consultant Data Wareh...
empiricus GmbH - Agentur für I...

> 16 more job recommendations

Share something with your contacts

Datasets

Users

Job roles

Career level

Discipline

Industry

Country

Region

Work experience

Education

Items

Title

Discipline

Industry

Country

Region

Type of employment

Tags

Creation time

Interactions

User_ID

Item_ID

Time

Interaction_type

Interactions Data

- Purely focus on interactions

1 - user clicked on the item

2 - user bookmarked the item on XING

3 - user clicked the reply button or application form button

4 - user deleted the recommendation

- Issues:

Data is sparse and doesn’t include all users

Not every user has at least 30 positive interactions

Collaborative Filtering

- Leverages the notion

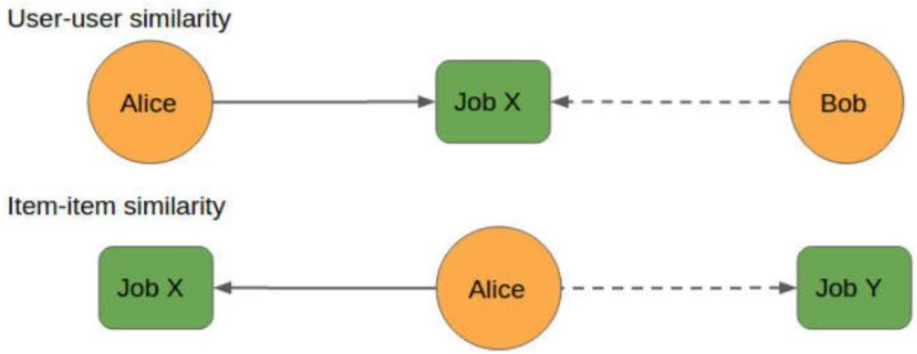
User-user similarity

Item-item similarity

- Challenges:

Sparsity

Definition of Similarity



Collaborative Filtering

- For similarity, we use K-Means clustering with cosine similarity
- Impute the interaction of user u with item i as average of interactions of other users v, in the same cluster, weighted by cosine similarity

Collaborative Filtering

How to we know if two items or users are similar?

K-means Clustering

- Number of clusters: 100, 1K, 5K
- Distance measure : Euclidean distance
- Library: SciPy.kmeans2

Collaborative Filtering

Cosine Similarity

- Similarity (U i , U j) = U i .U j / |U i | |U j |
- Similarity (I i , I j) = I i .I j / |I i | |I j |

Q: What are we using?

A: Both. Cosine similarity, but limited to the cluster

Scoring

- Given a user, score each job and finally rank based on score:

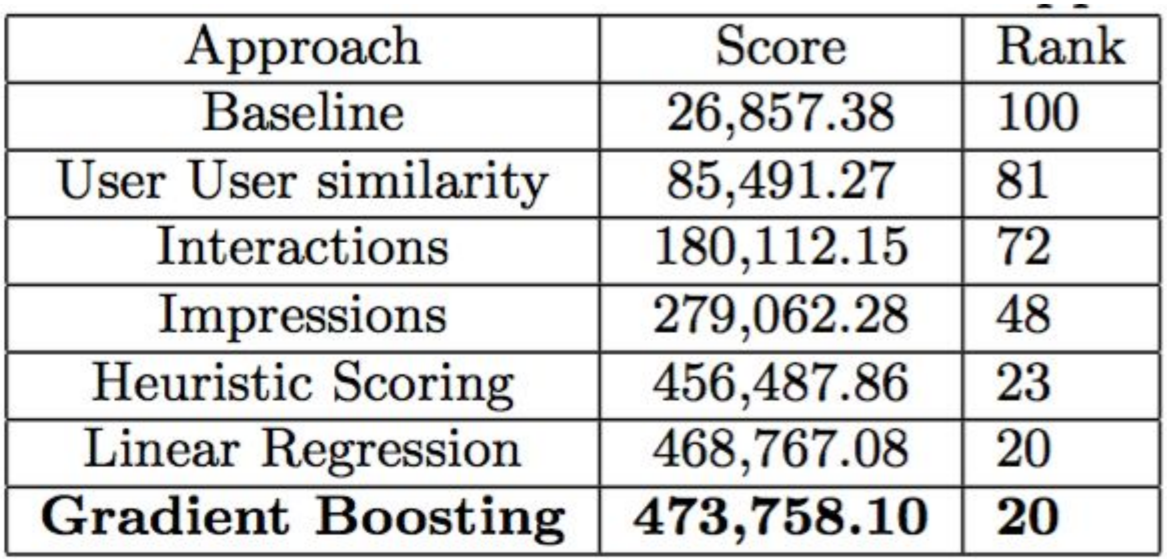
    Score **=** ***w1*****(Interaction score) +** ***w2*****(no. of overlaps in user job-roles and item-titles) +** ***w3*** **(I[career level match]) +** ***w4*** **(I[discipline IDs match]) +** ***w5*****(I[industry IDs match] + w6 (I[regions match])**

- Weights

    Heuristic: w1 = 1, w2 = 100, w3 = 10, w4 = 12, w5 = 10, w6 = 5 & w7 = 2

- Linear Regression: Learn the weights

Evaluations



Approach	Score	Rank
Baseline	26,857.38	100
User User similarity	85,491.27	81
Interactions	180,112.15	72
Impressions	279,062.28	48
Heuristic Scoring	456,487.86	23
Linear Regression	468,767.08	20
Gradient Boosting	**473,758.10**	**20**

Conclusion