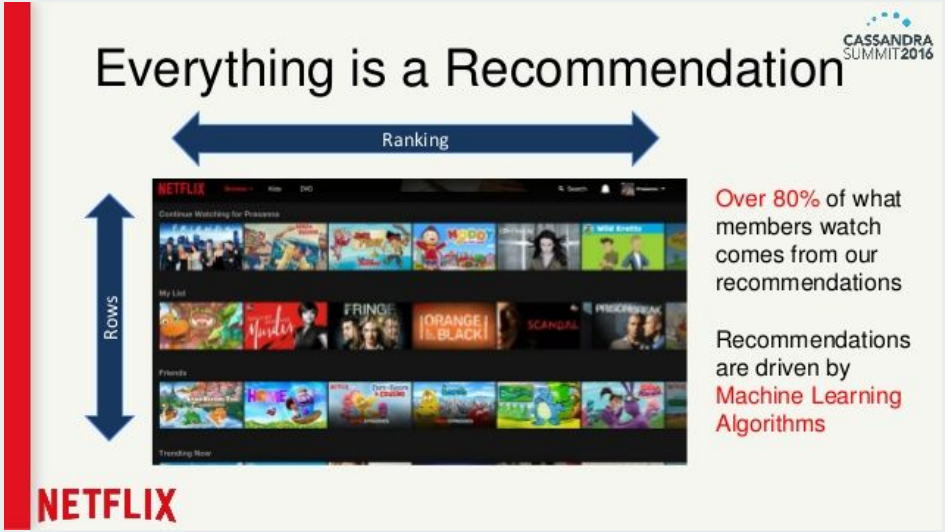
Project 4: Collaborative Filtering

An exercise in Algorithm Implementation and Evaluation

By: Xin Luo, Yi Zhang, Enrique Olivo, Wyatt Thompson
For: Applied Data Science STATGR5423, Professor Ying Liu

Outline

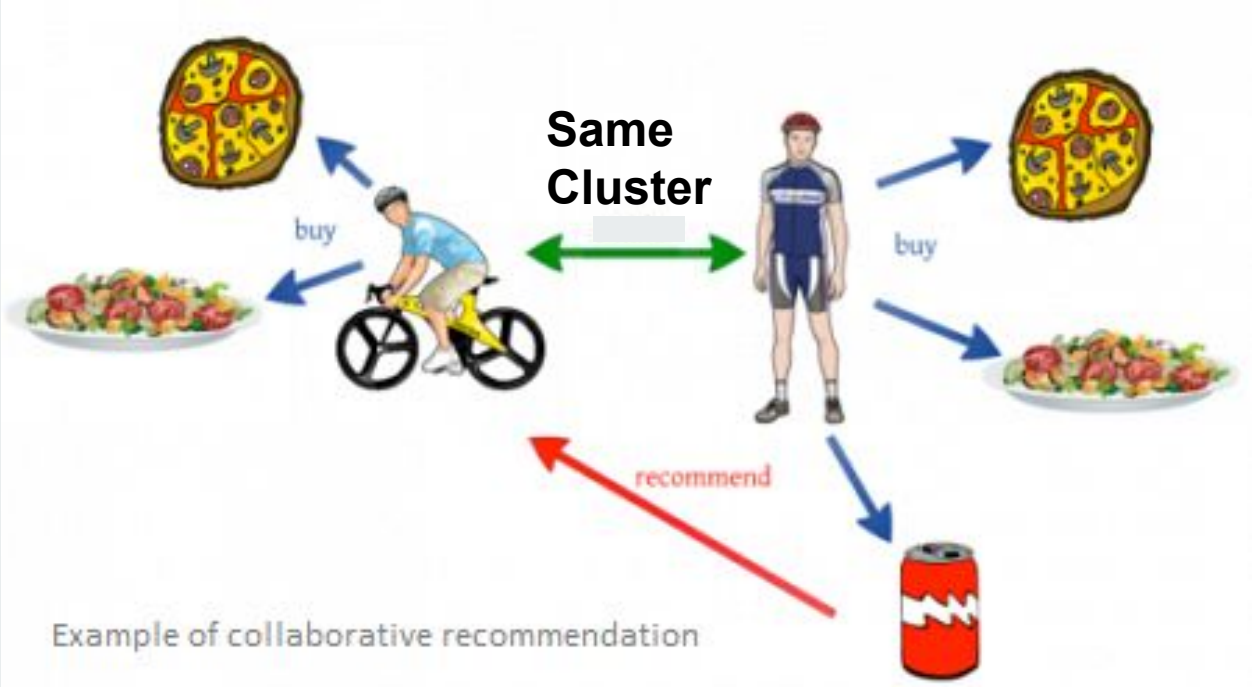
- Model Based
  - Cluster Model
- Memory Based
  - SimRank
    - Original
    - Variance Weighted
    - Neighbor Selection
    - Var + Neighbor Selection
  - Vector Similarity
    - Original
    - Variance Weighted
    - Neighbor Selection



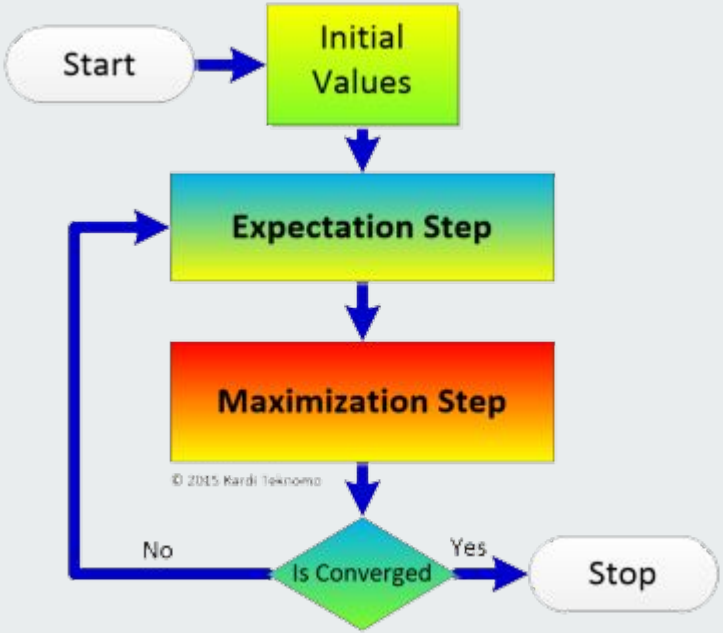
Cluster Model

Seeks to define clusters of users and use the distribution of Scores within clusters to estimate scores.

We use a Naive Bayes model to make initial guesses about the weights for being a member of a certain cluster and the distribution of scores within that cluster



Cluster Model (Continued)

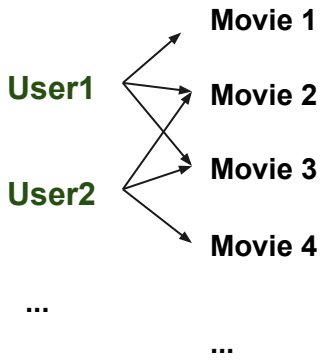


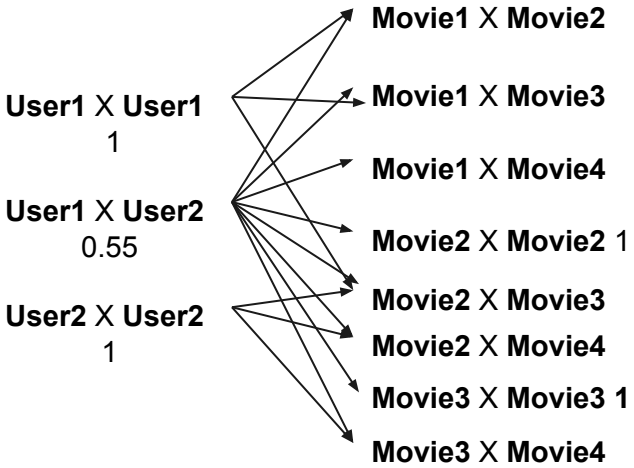
$$\hat{\mu}_c = \frac{\sum_{i=1}^{N} \hat{\pi}_i^c}{N}, \quad \text{for} \quad c = 1, ..., C$$

$$\hat{\gamma}_{c,j}^{(k)} = \frac{\sum_{i:j \in I(i)} \hat{\pi}_i^c \cdot \mathbb{I}\left(v_j^{(i)} = k\right)}{\sum_{i:j \in I(i)} \hat{\pi}_i^c}, \quad \text{for} \quad \forall c, j, k \tag{11}$$

SimRank: Movies

For the movie dataset, we apply **Bipartite SimRank** to it. But the SimRank algorithm only makes a graph for users and movies they rated, on the other hand, it cannot take the rates into consideration.





Original and Variance Weighted Results from SimRank

Algorithm	Normalization	MAE	ROC
SimRank	dev.from.mean	0.9354694	0.7505788
SimRank	z-score	0.9262371	0.7508749
SimRank+**Var.Weight**	dev.from.mean	1.1366397	0.6701729
SimRank+**Var.Weight**	z-score	1.1365447	0.6702932

Best One
0.065s/p/m

Selecting Neighbors Results from SimRank

Algorithm	Normalization	MAE	ROC
SimRank+Sel.Nei **50**	dev.from.mean	0.9370383	0.7496939
	z-score	0.9299381	0.7490959
SimRank+Sel.Nei **100**	dev.from.mean	0.9350571	0.7505077
	z-score	0.9263176	0.7503172
SimRank+Sel.Nei **200**	dev.from.mean	0.9346649	0.7506930
	z-score	0.9259002	0.7507565

Variance Weighted + Selecting Neighbors

Algorithm	Normalization	MAE	ROC
SimRank +**Var.Weight** +**Sel.Nei 200**	dev.from.mean	1.0658821	0.7059171
	z-score	1.0653316	0.7073118

Vector Similarity

Vector Similarity applied in dataset Movie and Web is similar to the way in information retrieval.

Users take the role of documents, Items take the role of words and votes is word frequencies.

$$w(a,i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}}$$

Original and Variance Weighted Results in *Movie DS*

Algorithm	Normalization	MAE	ROC
Vector Similarity	dev.from.mean	1.087116	0.8849790
Vector Similarity	z-score	1.084727	0.8856219
VectorS+**Var.Weight**	dev.from.mean	1.087145	0.8850208
VectorS+**Var.Weight**	z-score	1.084753	0.8856460

Selecting 50-neighbors in *Movie DS*

Algorithm	Normalization	MAE	ROC
VectorS+**Sel Nei 50**	dev.from.mean	1.027988	0.8929157
VectorS+**Sel Nei 50**	z-score	1.017934	0.8936828

Var Weighted + 50 Neighbors in *Movie DS*

Algorithm	Normalization	MAE	ROC
Vector Similarity **+Var.Weight +Sel.Nei 50**	dev.from.mean	1.044636	0.8903566
	z-score	1.037788	0.8908158

Incorporate Variance weighting does not have significant effect on increasing the accuracy.

Original and Variance Weighted Results in *Web* DS

Algorithm	Normalization	Rank Scoring
Vector Similarity	dev.from.mean	34.39748
Vector Similarity	z-score	33.77172
VectorS+**Var.Weight**	dev.from.mean	34.15969
VectorS+**Var.Weight**	z-score	33.58498