Recursive Nearest Neighbors Methods in Recommender Systems

Stylianos Tsesmetzis
Supervisor: Nicholas Ampazis

University of the Aegean
School of Engineering
Department of Financial and Management Engineering

October 31, 2017

Table of Contents

Table of Contents

Recommender Systems Overview

- What is a recommender system and what does it do?

Recommender Systems Overview

- What is a recommender system and what does it do?
  1. Predict how much a user may like a certain item

Recommender Systems Overview

- What is a recommender system and what does it do?
  1. Predict how much a user may like a certain item
  2. Compose a list of N best items for a user

Recommender Systems Overview

- What is a recommender system and what does it do?
  1. Predict how much a user may like a certain item
  2. Compose a list of N best items for a user
  3. Compose a list of N best users for a specific item

Recommender Systems Overview

- What is a recommender system and what does it do?
  1. Predict how much a user may like a certain item
  2. Compose a list of N best items for a user
  3. Compose a list of N best users for a specific item
  4. Explain to the users why these items are recommended them

Recommender Systems Overview

- What is a recommender system and what does it do?
  1. Predict how much a user may like a certain item
  2. Compose a list of N best items for a user
  3. Compose a list of N best users for a specific item
  4. Explain to the users why these items are recommended them
  5. Adjust the prediction and recommendation based on user's and other people feedback

Recommender Systems Overview

- What is a recommender system and what does it do?
- Why we need a recommender system?

Recommender Systems Overview

- What is a recommender system and what does it do?
- Why we need a recommender system?
- How we build a recommender system?

Recommender Systems Overview

- What is a recommender system and what does it do?
- Why we need a recommender system?
- How we build a recommender system?
  1. **Collaborative Filtering**
     - Neighborhood-based
     - Model-based

Recommender Systems Overview

- What is a recommender system and what does it do?
- Why we need a recommender system?
- How we build a recommender system?
  1. **Collaborative Filtering**
     - Neighborhood-based
     - Model-based
  2. **Content-based**

Recommender Systems Overview

- What is a recommender system and what does it do?
- Why we need a recommender system?
- How we build a recommender system?
  1. **Collaborative Filtering**
     - Neighborhood-based
     - Model-based
  2. **Content-based**
  3. **Knowledge Based**

Recommender Systems Overview

- What is a recommender system and what does it do?
- Why we need a recommender system?
- How we build a recommender system?
  1. **Collaborative Filtering**
     - Neighborhood-based
     - Model-based
  2. **Content-based**
  3. **Knowledge Based**
  4. **Hybrid**

Motivation

Problems with collaborative filtering

1. **Scale**
   - New users and items enter a recommendation platform everyday.
2. **Sparse data**
   - A user has rated only one item or an item has been rated only once.
3. **Cold-Start**
   - New users and items have no history.

Approach

Objectives

1. Explain the Neighborhood-based Collaborative Filtering
2. Propose the Recursive Nearest Neighbors Algorithm
3. Present the Experimental results on the Epinions data set.

Table of Contents

Introduction

Collaborative Filtering

- **User-based**
  - Many People like "The Godfather" should I watch it too?
  - Choose a number of users who like things I like and decide based on how much they liked it
  - If we used to like similar items in the past, we will continue to like similar items in the future
- **Item-based**
  - Is "Jurassic Park" a good choice based on movies I usually see?
  - Choose a number of movies that I have seen and share similar audience with "Jurassic Park", then decide based on how much I liked the previous movies
  - If I liked these type of items in the past, I will probably also like those items

Introduction

Advantages of Collaborative Filtering

1. Simplicity
2. Justifiability
3. Efficiency
4. Stability

Cosine Similarity

User-based

$$cos(u,v) = \frac{\sum_{i \in \mathcal{I}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathcal{I}_u} r_{ui}^2} \sqrt{\sum_{i \in \mathcal{I}_v} r_{vi}^2}}$$

- $\mathcal{I}_u$: The set of items that have been rated by user u
- $\mathcal{I}_v$: The set of items that have been rated by user v
- $\mathcal{I}_{uv}$: The set of items that users u and v rated in common
- r_{ui} : The rating that user u gave to item i
- r_{vi} : The rating that user v gave to item i

Item-based

$$cos(i,j) = \frac{\sum_{u \in \mathcal{U}_{ij}} r_{iu} r_{ju}}{\sqrt{\sum_{u \in \mathcal{U}_i} r_{iu}^2} \sqrt{\sum_{u \in \mathcal{U}_j} r_{ju}^2}}$$

- $\mathcal{U}_i$: The set of users that have rated item i
- $\mathcal{U}_j$: The set of users that have rated item j
- $\mathcal{U}_{ij}$: The set of users that have both rated items i and j
- r_{iu} : The rating item i received from user u
- r_{ju} : The rating item j received from user u

Modified Cosine Similarity

User-based

$$MC(u,v) = \frac{\sum_{i \in \mathcal{I}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} r_{ui}^2} \sqrt{\sum_{i \in \mathcal{I}_{uv}} r_{vi}^2}}$$

- $\mathcal{I}_{uv}$: The set of items that users u and v rated in common
- r_{ui} : The rating that user u gave to item i
- r_{vi} : The rating that user v gave to item i

Item-based

$$MC(i,j) = \frac{\sum_{u \in \mathcal{U}_{ij}} r_{iu} r_{ju}}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} r_{iu}^2} \sqrt{\sum_{u \in \mathcal{U}_{ij}} r_{ju}^2}}$$

- $\mathcal{U}_{ij}$: The set of users that have both rated items i and j
- r_{iu} : The rating item i received from user u
- r_{ju} : The rating item j received from user u

Adjusted Cosine Similarity

User-based

$$AC(u,v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_i)(r_{vi} - \bar{r}_i)}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_i)^2} \sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{vi} - \bar{r}_i)^2}}$$

$$\bar{r}_i = \frac{\sum_{u \in \mathcal{U}_i} r_{iu}}{|\mathcal{U}_i|}$$

- $\mathcal{I}_{uv}$: The set of items that users u and v rated in common
- r_{ui} : The rating that user u gave to item i
- r_{vi} : The rating that user v gave to item i
- $\mathcal{U}_i$: The set of users that have rated item i
- $\bar{r}_i$: The mean rating of item i

Item-based

$$AC(i,j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{iu} - \bar{r}_u)(r_{ju} - \bar{r}_u)}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{iu} - \bar{r}_u)^2} \sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{ju} - \bar{r}_u)^2}}$$

$$\bar{r}_u = \frac{\sum_{i \in \mathcal{I}_u} r_{ui}}{|\mathcal{I}_u|}$$

- $\mathcal{U}_{ij}$: The set of users that have both rated items i and j
- r_{iu} : The rating item i received from user u
- r_{ju} : The rating item j received from user u
- $\mathcal{I}_u$: The set of items that have been rated by user u
- $\bar{r}_u$: The mean rating of user u

Modified Adjusted Cosine Similarity

User-based

$$MAC(u,v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_i)(r_{vi} - \bar{r}_i)}{\sqrt{\sum_{i \in \mathcal{I}_u} (r_{ui} - \bar{r}_i)^2} \sqrt{\sum_{i \in \mathcal{I}_v} (r_{vi} - \bar{r}_i)^2}}$$

$$\bar{r}_i = \frac{\sum_{u \in \mathcal{U}_i} r_{iu}}{|\mathcal{U}_i|}$$

- $\mathcal{I}_u$: The set of items that have been rated by user u
- $\mathcal{I}_v$: The set of items that have been rated by user v
- $\mathcal{I}_{uv}$: The set of items that users u and v rated in common
- r_{ui} : The rating that user u gave to item i
- r_{vi} : The rating that user v gave to item i
- $\mathcal{U}_i$: The set of users that have rated item i
- $\bar{r}_i$: The mean rating of item i

Item-based

$$MAC(i,j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{iu} - \bar{r}_u)(r_{ju} - \bar{r}_u)}{\sqrt{\sum_{u \in \mathcal{U}_i} (r_{iu} - \bar{r}_u)^2} \sqrt{\sum_{u \in \mathcal{U}_j} (r_{ju} - \bar{r}_u)^2}}$$

$$\bar{r}_u = \frac{\sum_{i \in \mathcal{I}_u} r_{ui}}{|\mathcal{I}_u|}$$

- $\mathcal{U}_i$: The set of users that have rated item i
- $\mathcal{U}_j$: The set of users that have rated item j
- $\mathcal{U}_{ij}$: The set of users that have both rated items i and j
- r_{iu} : The rating item i received from user u
- r_{ju} : The rating item j received from user u
- $\mathcal{I}_u$: The set of items that have been rated by user u
- $\bar{r}_u$: The mean rating of user u

Pearson Correlation Coefficient

User-based

$$PCC(u,v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

$$\bar{r}_u = \frac{\sum_{i \in \mathcal{I}_u} r_{ui}}{|\mathcal{I}_u|}$$

- $\mathcal{I}_{uv}$: The set of items that users u and v rated in common
- r_{ui} : The rating that user u gave to item i
- r_{vi} : The rating that user v gave to item i
- $\bar{r}_u$: The mean rating of user u

Item-based

$$PCC(i,j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{iu} - \bar{r}_i)(r_{ju} - \bar{r}_j)}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{iu} - \bar{r}_i)^2} \sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{ju} - \bar{r}_j)^2}}$$

$$\bar{r}_i = \frac{\sum_{u \in \mathcal{U}_i} r_{iu}}{|\mathcal{U}_i|}$$

- $\mathcal{U}_{ij}$: The set of users that have both rated items i and j
- r_{iu} : The rating item i received from user u
- r_{ju} : The rating item j received from user u
- $\bar{r}_i$: The mean rating of item i

Modified Pearson Correlation Coefficient 1

User-based

$$MPCC1(u,v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \tilde{r}_u)(r_{vi} - \tilde{r}_v)}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \tilde{r}_u)^2} \sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{vi} - \tilde{r}_v)^2}}$$

$$\tilde{r}_u = \frac{\sum_{i \in \mathcal{I}_{uv}} r_{ui}}{|\mathcal{I}_{uv}|}$$

- $\mathcal{I}_{uv}$: The set of items that users u and v rated in common
- r_{ui} : The rating that user u gave to item i
- r_{vi} : The rating that user v gave to item i
- $\tilde{r}_u$: The mean rating of user u

Item-based

$$MPCC1(i,j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{iu} - \tilde{r}_i)(r_{ju} - \tilde{r}_j)}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{iu} - \tilde{r}_i)^2} \sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{ju} - \tilde{r}_j)^2}}$$

$$\tilde{r}_i = \frac{\sum_{u \in \mathcal{U}_i} r_{iu}}{|\mathcal{U}_i|}$$

- $\mathcal{U}_{ij}$: The set of users that have both rated items i and j
- r_{iu} : The rating item i received from user u
- r_{ju} : The rating item j received from user u
- $\tilde{r}_i$: The mean rating of item i

Modified Pearson Correlation Coefficient 2

User-based

$$MPCC2(u,v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in \mathcal{I}_u} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in \mathcal{I}_v} (r_{vi} - \bar{r}_v)^2}}$$

$$\bar{r}_u = \frac{\sum_{i \in \mathcal{I}_u} r_{ui}}{|\mathcal{I}_u|}$$

- $\mathcal{I}_u$: The set of items that have been rated by user u
- $\mathcal{I}_v$: The set of items that have been rated by user v
- $\mathcal{I}_{uv}$: The set of items that users u and v rated in common
- r_{ui} : The rating that user u gave to item i
- r_{vi} : The rating that user v gave to item i
- $\bar{r}_u$: The mean rating of user u

Item-based

$$MPCC2(i,j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{iu} - \bar{r}_i)(r_{ju} - \bar{r}_j)}{\sqrt{\sum_{u \in \mathcal{U}_i} (r_{iu} - \bar{r}_i)^2} \sqrt{\sum_{u \in \mathcal{U}_j} (r_{ju} - \bar{r}_j)^2}}$$

$$\bar{r}_i = \frac{\sum_{u \in \mathcal{U}_i} r_{iu}}{|\mathcal{U}_i|}$$

- $\mathcal{U}_i$: The set of users that have rated item i
- $\mathcal{U}_j$: The set of users that have rated item j
- $\mathcal{U}_{ij}$: The set of users that have both rated items i and j
- r_{iu} : The rating item i received from user u
- r_{ju} : The rating item j received from user u
- $\bar{r}_i$: The mean rating of item i

Mean Squared Difference

User-based

$$MSD(u,v) = \frac{|\mathcal{I}_{uv}|}{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - r_{vi})^2}$$

- $\mathcal{I}_{uv}$: The set of items that users u and v rated in common
- r_{ui} : The rating that user u gave to item i
- r_{vi} : The rating that user v gave to item i

Item-based

$$MSD(i,j) = \frac{|\mathcal{U}_{ij}|}{\sum_{u \in \mathcal{U}_{ij}} (r_{iu} - r_{ju})^2}$$

- $\mathcal{U}_{ij}$: The set of users that have both rated items i and j
- r_{iu} : The rating item i received from user u
- r_{ju} : The rating item j received from user u

Mean Absolute Difference

User-based

$$MAD(u,v) = \frac{|\mathcal{I}_{uv}|}{\sum_{i \in \mathcal{I}_{uv}} |r_{ui} - r_{vi}|}$$

- $\mathcal{I}_{uv}$: The set of items that users u and v rated in common
- r_{ui} : The rating that user u gave to item i
- r_{vi} : The rating that user v gave to item i

Item-based

$$MAD(i,j) = \frac{|\mathcal{U}_{ij}|}{\sum_{u \in \mathcal{U}_{ij}} |r_{iu} - r_{ju}|}$$

- $\mathcal{U}_{ij}$: The set of users that have both rated items i and j
- r_{iu} : The rating item i received from user u
- r_{ju} : The rating item j received from user u

Jaccard Coefficient

User-based

$$J(u,v) = \frac{|\mathcal{I}_{uv}|}{|\mathcal{I}_u| + |\mathcal{I}_v| - |\mathcal{I}_{uv}|}$$

- $\mathcal{I}_u$: The set of items that have been rated by user u
- $\mathcal{I}_v$: The set of items that have been rated by user v
- $\mathcal{I}_{uv}$: The set of items that users u and v rated in common
- $|\mathcal{I}_u|$: The number of items in set $\mathcal{I}_u$
- $|\mathcal{I}_v|$: The number of items in set $\mathcal{I}_v$
- $|\mathcal{I}_{uv}|$: The number of items in set $\mathcal{I}_{uv}$

Item-based

$$J(i,j) = \frac{|\mathcal{U}_{ij}|}{|\mathcal{U}_i| + |\mathcal{U}_j| - |\mathcal{U}_{ij}|}$$

- $\mathcal{U}_i$: The set of users that have rated item i
- $\mathcal{U}_j$: The set of users that have rated item j
- $\mathcal{U}_{ij}$: The set of users that rated items i and j in common
- $|\mathcal{U}_i|$: The number of users in set $\mathcal{U}_i$
- $|\mathcal{U}_j|$: The number of users in set $\mathcal{U}_j$
- $|\mathcal{U}_{ij}|$: The number of users in set $\mathcal{U}_{ij}$

K-Nearest Neighbors Algorithm

The K-Nearest Neighbors algorithm

K-Nearest Neighbors Algorithm

Step 1: Select users that have rated $Item_B$.

K-Nearest Neighbors Algorithm

Step 1: Select users that have rated $Item_B$.

Step 2: Compute the similarities between $User_A$ and the users that have rated $Item_B$.

K-Nearest Neighbors Algorithm

Step 1: Select users that have rated $Item_B$.

Step 2: Compute the similarities between $User_A$ and the users that have rated $Item_B$.

Step 3: Sort the similarities in descending order.

K-Nearest Neighbors Algorithm

Step 1: Select users that have rated $Item_B$.

Step 2: Compute the similarities between $User_A$ and the users that have rated $Item_B$.

Step 3: Sort the similarities in descending order.

Step 4: Choose how many neighbors will contribute in the rating prediction by selecting the top $\mathcal{K}$ out of all the available neighbors($\mathcal{K}$ can be in range [1 - $\mathcal{N}$] where $\mathcal{N}$ is all the available neighbors).

K-Nearest Neighbors Algorithm

Step 1: Select users that have rated $Item_B$.

Step 2: Compute the similarities between $User_A$ and the users that have rated $Item_B$.

Step 3: Sort the similarities in descending order.

Step 4: Choose how many neighbors will contribute in the rating prediction by selecting the top $\mathcal{K}$ out of all the available neighbors($\mathcal{K}$ can be in range [1 - $\mathcal{N}$] where $\mathcal{N}$ is all the available neighbors).

Step 5: Use an aggregation formula to calculate the rating prediction of $User_A$ to $Item_B$. In this case the weighted sum is used.

$$\hat{r}(User_A, Item_B) = \frac{\sum_{u \in \mathcal{K}} similarity(User_A, User_u) * r(User_u, Item_B)}{\sum_{u \in \mathcal{K}} |similarity(User_A, User_u)|}$$

Table of Contents

Introduction

User \ Item	$Item_1$	$Item_2$	$Item_3$	$Item_4$	$Item_5$	$Item_6$
$User_1$	5	2	3	?	1	5
$User_2$	1	2	4	?	2	2
$User_3$	4	3	5	?	4	3
$User_4$	5	2	3	?	?	?
$User_5$	?	?	?	4	1	1
$User_6$	?	?	?	3	5	2
$User_7$	?	?	?	5	1	2
$User_8$	?	?	?	5	4	4

Table: Ratings Matrix

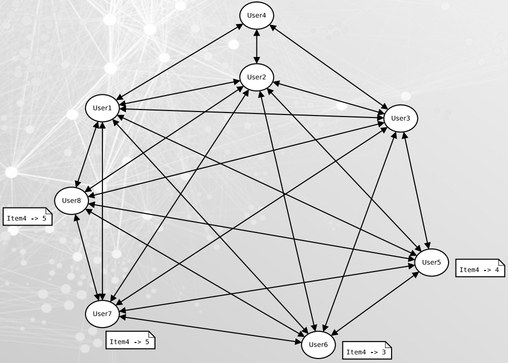


Figure: User Connections

Introduction

Item / User	$Item_1$	$Item_2$	$Item_3$	$Item_4$	$Item_5$	$Item_6$
$User_1$	5	2	3	4.3	1	5
$User_2$	1	2	4	4.15	2	2
$User_3$	4	3	5	4.12	4	3
$User_4$	5	2	3	?	2.35	3.42
$User_5$	3.35	2.35	4.02	4	1	1
$User_6$	3.21	2.4	4.15	3	5	2
$User_7$	3.46	2.32	3.94	5	1	2
$User_8$	3.36	2.35	4.03	5	4	4

Table: Ratings Matrix After KNN

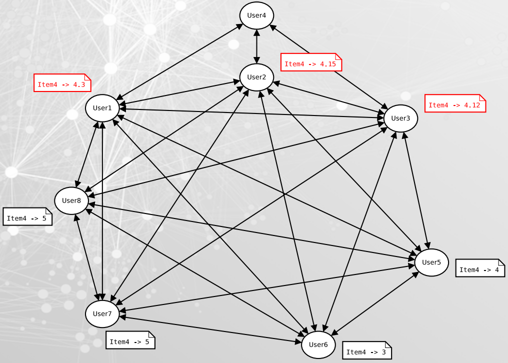


Figure: User Connections

The Recursive K-Nearest Neighbors Algorithm

The Recursive K-Nearest Neighbors algorithm

The Recursive K-Nearest Neighbors Algorithm

Step 1: Select users that have similarity connections with $User_A$, name it $Group_A$.

The Recursive K-Nearest Neighbors Algorithm

Step 1: Select users that have similarity connections with $User_A$, name it $Group_A$.

Step 2: Out of $Group_A$, select those users that have similarity connections with other users who have rated $Item_B$, name it $Group_B$.

The Recursive K-Nearest Neighbors Algorithm

Step 1: Select users that have similarity connections with $User_A$, name it $Group_A$.

Step 2: Out of $Group_A$, select those users that have similarity connections with other users who have rated $Item_B$, name it $Group_B$.

Step 3: Sort $Group_B$ in descending order.

The Recursive K-Nearest Neighbors Algorithm

Step 1: Select users that have similarity connections with $User_A$, name it $Group_A$.

Step 2: Out of $Group_A$, select those users that have similarity connections with other users who have rated $Item_B$, name it $Group_B$.

Step 3: Sort $Group_B$ in descending order.

Step 4: Choose how many neighbors from $Group_B$ will contribute in the rating prediction by selecting the top $\mathcal{K}$ out of all the available neighbors in this group, name it $Group_C$.

The Recursive K-Nearest Neighbors Algorithm

Step 1: Select users that have similarity connections with $User_A$, name it $Group_A$.

Step 2: Out of $Group_A$, select those users that have similarity connections with other users who have rated $Item_B$, name it $Group_B$.

Step 3: Sort $Group_B$ in descending order.

Step 4: Choose how many neighbors from $Group_B$ will contribute in the rating prediction by selecting the top $\mathcal{K}$ out of all the available neighbors in this group, name it $Group_C$.

Step 5: For each neighbor in $Group_C$, predict how this neighbor would rate $Item_B$ using the KNN algorithm. For convenience, call the number of recursive neighbors each neighbor in $Group_C$ uses, $\mathcal{M}$-Nearest Neighbors.

The Recursive K-Nearest Neighbors Algorithm

Step 1: Select users that have similarity connections with $User_A$, name it $Group_A$.

Step 2: Out of $Group_A$, select those users that have similarity connections with other users who have rated $Item_B$, name it $Group_B$.

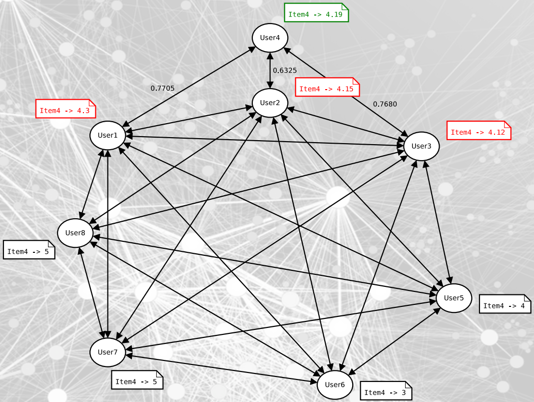
Step 3: Sort $Group_B$ in descending order.

Step 4: Choose how many neighbors from $Group_B$ will contribute in the rating prediction by selecting the top $\mathcal{K}$ out of all the available neighbors in this group, name it $Group_C$.

Step 5: For each neighbor in $Group_C$, predict how this neighbor would rate $Item_B$ using the KNN algorithm. For convenience, call the number of recursive neighbors each neighbor in $Group_C$ uses, $\mathcal{M}$-Nearest Neighbors.

Step 6: Perform the KNN algorithm for $User_A$ to $Item_B$ using the rating predictions applied on $Group_C$.

The Recursive K-Nearest Neighbors Algorithm



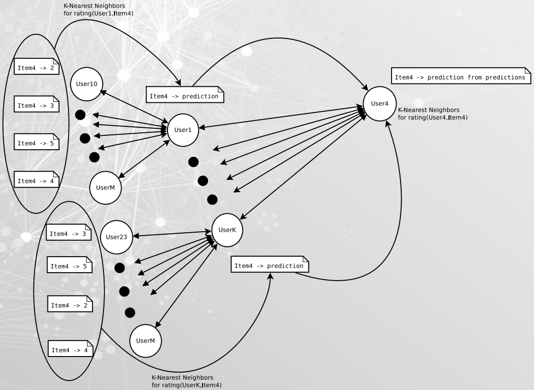


Table of Contents

Epinions data set

Table: Epinions Sample

user	item	Rating
36153	62461	5
427	38005	5
751	53361	4
11001	118950	4
1169	66176	5
9808	84459	2
85	7446	4
14717	3397	2

Table: Epinions Descriptive

	Ratings Matrix	Train	Test
count	664824	520203	144621
mean	3.9917	3.99	3.9975
std	1.2068	1.2072	1.2053
min	1	1	1
25%	3	3	3
50%	4	4	4
75%	5	5	5
max	5	5	5

Evaluation Metrics

Single Model

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in\mathcal{T}}(r_{u,i}-\hat{r}_{u,i})^2}{n}}$$

$$MAE = \frac{\sum_{(u,i)\in\mathcal{T}}|r_{u,i}-\hat{r}_{u,i}|}{n}$$

$$RMSUE = \frac{1}{n}\sum_{u\in\mathcal{T}}\sqrt{\frac{\sum_{i\in\mathcal{I}_u}(r_{u,i}-\hat{r}_{u,i})^2}{n_u}}$$

$$MAUE = \frac{1}{n}\sum_{u\in\mathcal{T}}\frac{\sum_{i\in\mathcal{I}_u}|r_{u,i}-\hat{r}_{u,i}|}{n_u}$$

Combined Models

$$RMSE_{Total} = \sqrt{\frac{n_{KNN} * RMSE_{KNN}^2 + n_{R-KNN} * RMSE_{R-KNN}^2}{n_{KNN} + n_{R-KNN}}}$$

$$MAE_{Total} = \frac{n_{KNN} * MAE_{KNN} + n_{R-KNN} * MAE_{R-KNN}}{n_{KNN} + n_{R-KNN}}$$

- $\mathcal{T}$ is the test set
- $r_{u,i}$ is the truth value of a rating for $user_u$ to $item_i$
- $\hat{r_{u,i}}$ is the prediction value of a rating for $user_u$ to $item_i$
- n is the number of rating predictions

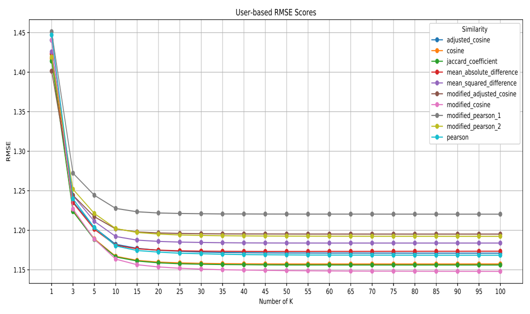
Volume of Predictions

Table: Ratings Predicted with KNN and Recursive-KNN

USERS			ITEMS			
KNN	R-KNN	TOTAL	KNN	R-KNN	TOTAL	SIMILARITY
88379	34392	122771	89800	33315	123115	Adjusted Cosine
100311	24122	124433	100311	24122	124433	Cosine
100311	24122	124433	100311	24122	124433	Jaccard
93924	29747	123671	94528	29203	123731	MAD
93924	29747	123671	94528	29203	123731	MSD
88379	34392	122771	89800	33315	123115	Modified Adjusted Cosine
100311	24122	124433	100311	24122	124433	Modified Cosine
59017	40470	99487	54644	34096	88740	Modified Pearson 1
89936	28183	118119	84511	24357	108868	Modified Pearson 2
89936	28183	118119	84511	24357	108868	Pearson

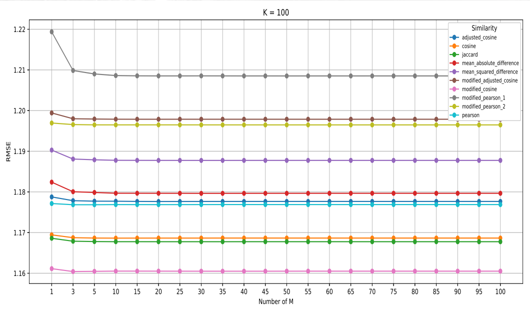
User-based KNN and Total RMSE

KNN



Modified cosine at K=100, RMSE=1.1479835373

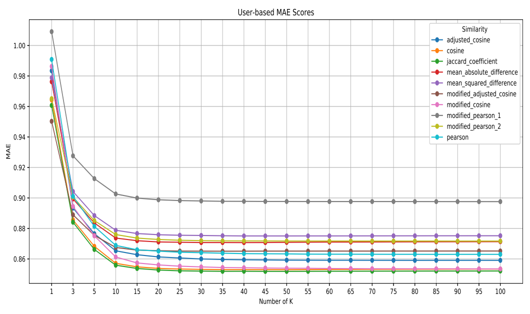
Total



Modified cosine at K=100 & M=3, RMSE=1.1604146071

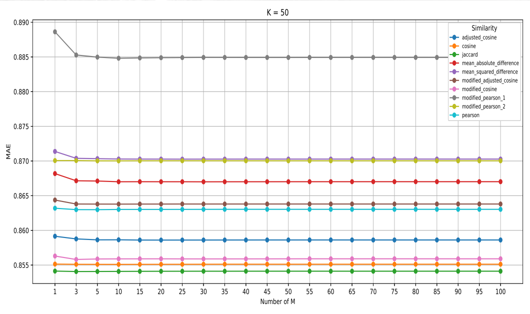
User-based KNN and Total MAE

KNN



Jaccard coefficient at K=50, MAE=0.8518433005

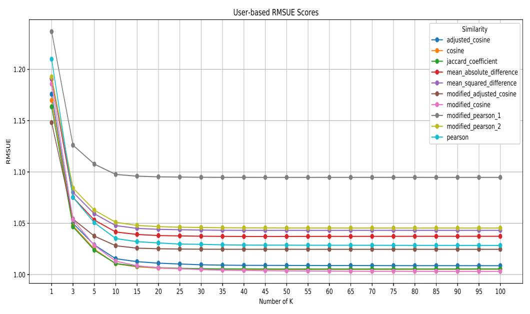
Total



Jaccard coefficient at K=50 & M=3, MAE=0.854066176

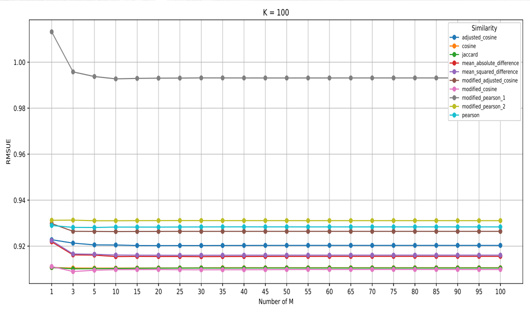
User-based KNN and Recursive-KNN RMSUE

KNN



Modified cosine at K=100, RMSUE=1.0031145695

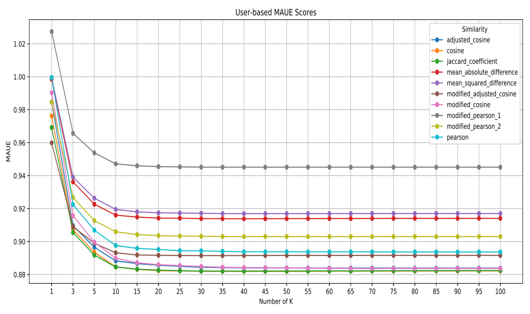
Recursive-KNN



Modified cosine at K=100 & M=3, RMSUE=0.9089525549

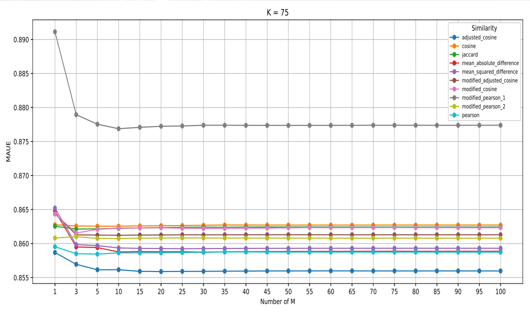
User-based KNN and Recursive-KNN MAUE

KNN



Cosine similarity at K=50, MAUE=0.8819077974

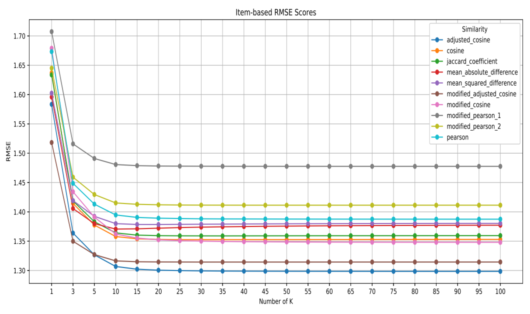
Recursive-KNN



Adjusted cosine at K=75 & M=20, MAUE=0.8558869566

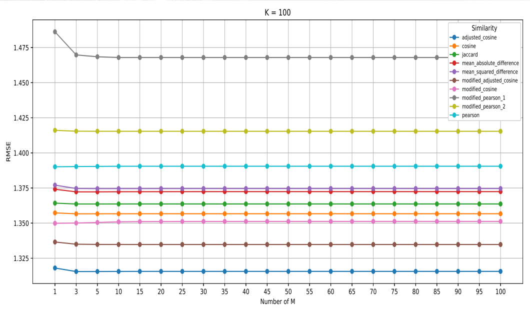
Item-based KNN and Total RMSE

KNN



Adjusted cosine at K=95, RMSE=1.2984970982

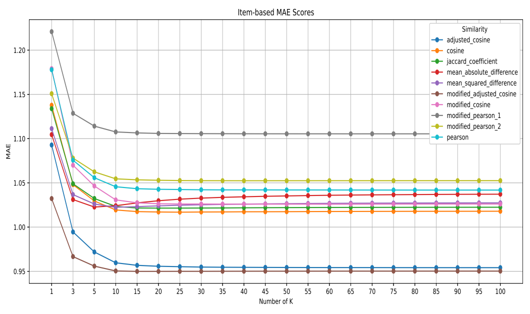
Total



Adjusted cosine at K=100 & M=3, RMSE=1.3155259043

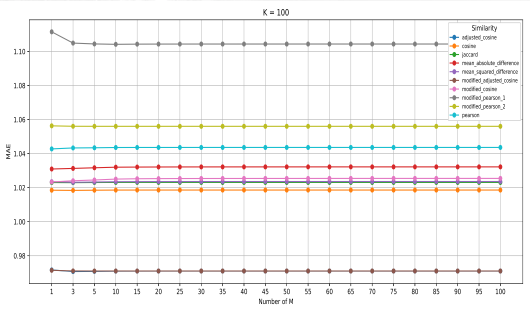
Item-based KNN and Total MAE

KNN



Modified adjusted cosine at K=15, MAE=0.9499219326

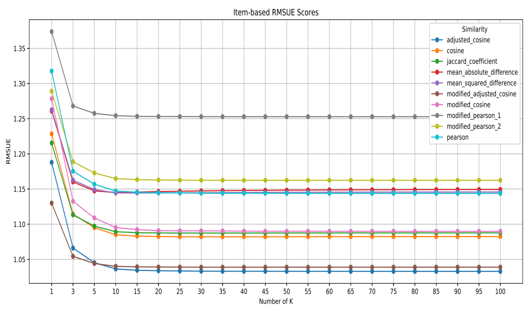
Total



Adjusted cosine at K=100 & M=3, MAE=0.9707211649

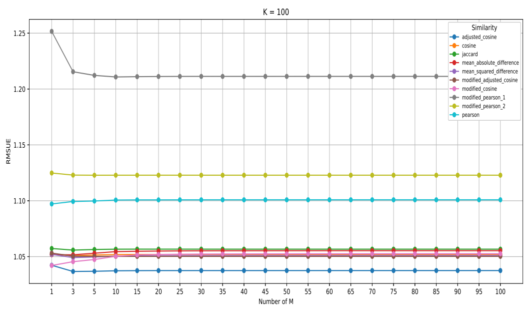
Item-based KNN and Recursive-KNN RMSUE

KNN



Adjusted cosine at K=100, RMSUE=1.0328968801

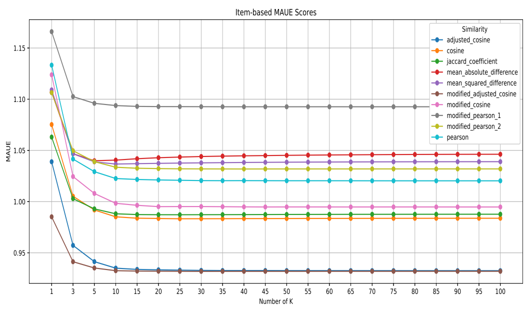
Recursive-KNN



Adjusted cosine at K=100 & M=3, RMSUE=1.0367245756

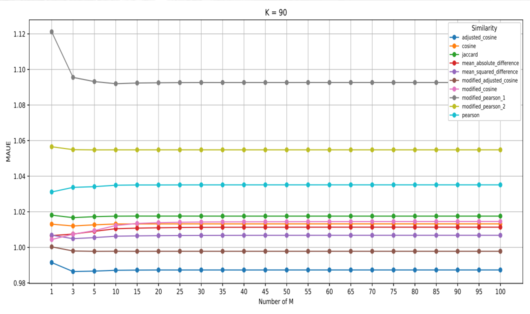
Item-based KNN and Recursive-KNN MAUE

KNN



Modified adjusted cosine at K=30, MAUE=0.9318609947

Recursive-KNN



Adjusted cosine at K=90 & M=3, MAUE=0.9864005988

Table of Contents

Conclusion

- A novel recursive approach for Neighborhood-based CF
- 25% increase in rating predictions
- Decent Results in User-based approach

Future Work

- Further use of similarity information
- Similarity re-computations including predictions
- Different prediction models

Table of Contents

Q & A

Any Questions?

Thank You!