Recommendation Systems

Anil Maheshwari

anil@scs.carleton.ca
School of Computer Science
Carleton University
Canada

Introduction

Recommendation Systems

Objective: Predict user responses to options

Examples:

1. Tailoring news depending on interest in an online newspaper
2. What to buy based on past history
3. Which book/show to recommend

Approach:

1. Content-based: If a user likes ‘Action’ Movies, recommend movies classified as of type ‘Action’
2. Collaborative Filtering: Recommend items based on similarity measures between users and/or items. Recommend items based on similar users.

Utility Matrix

Matrix consisting of m-users as rows and n-items as columns. The entries in the matrix are numerical values specifying the user-item preference.

Example: 4-users and 5-items. Ranking on the scale of $1 - 5$. Blanks refers to no ranking.

Utility Matrix:

	I_1	I_2	I_3	I_4	I_5
U_1	5	1		4	
U_2		1	2	3	4
U_3	1		4	1	
U_4		5		1	3

Problem: Predict the blank entries in Utility Matrix

Content-based RS

Problem: For each item construct a profile. Profile represents important characteristics of the item.

Examples:
Movies: Language, Country, Director, Genre, Year
Books: Author, Language, Year, Category

Techniques are adhoc
- look at item descriptor
- word counts for documents - usage of different words
- Jaccard distance between two documents based on the words they use
. . .

Collaborative Filtering

Collaborative Filtering RS

Key Principle: Two users (or items) are similar if their row (resp. column) vectors are similar in the utility matrix.

Collaborative Filtering: The process of identifying similar users/items and recommending what similar users like.

Problem: Find similar row/column vectors in the utility matrix.

Similarity Measures

Utility Matrix:

	I_1	I_2	I_3	I_4	I_5
U_1	5	1		4	
U_2		1	2	3	4
U_3	1		4	1	
U_4		5		1	3

1. Jaccard Distance: Treat any non-zero entry to mean 'membership' in the set: $JD(U_1, U_2) = 1 - JS(U_1, U_2) = 1 - \frac{2}{5} = \frac{3}{5}$
   $JD(U_1, U_4) = 1 - \frac{2}{4} = \frac{1}{2}$
   Does this makes sense?
2. Cosine Distance: Treat blank values as zeros and compute the Cosine Distance: Cos$(U_1, U_2) = \frac{U_1 \cdot U_2}{||U_1||||U_2||} = \frac{5 \cdot 0 + 1 \cdot 1 + 0 \cdot 2 + 4 \cdot 3 + 0 \cdot 4}{\sqrt{5^2+1^2+4^2}\sqrt{1^2+2^2+3^2+4^2}} =$
   $\frac{17}{\sqrt{42}\sqrt{46}} = 0.386 = 67^\circ$

   Cos$(U_1, U_4) = \frac{U_1 \cdot U_4}{||U_1||||U_4||} = \frac{9}{\sqrt{42}\sqrt{35}} = 0.23 = 76^\circ$

   (U_1, U_2) are similar as compared to (U_1, U_4)

Similarity Measures (contd.)

Fluctuations in Ratings: A user may rate everything higher compared to other users.

Normalization: Subtract the average rating of the user from all of its ratings:

	I_1	I_2	I_3	I_4	I_5	Average
U_1	5/3	-7/3		2/3		10/3
U_2		-2	-1	1	2	3
U_3	-1		2	-1		2
U_4		2		-2	0	3

Cos$(U_1, U_2) = 0.57 = 55°$

Cos$(U_1, U_4) = -.72 = 136°$

(U_1, U_2) are more similar as compared to (U_1, U_4)

UV Decomposition

UV-Decomposition

Approximate Utility Matrix M by the product of two low rank matrices U and V, i.e. $[M]_{m\times n} \approx [U]_{m\times d}[V]_{d\times n}$, where d is small.

Intuition: d represents the set of features for which the users respond. Matrix U corresponds to users preferences for those features and V corresponds to connection between items and features.

Examples: Movies: $d \sim 8 - 10$

$$M = \begin{bmatrix} 2 & 1 & 4 & 3 \\ 7 & 1 & 0 & 6 \\ 1 & 2 & 1 & 4 \\ 2 & 1 & 6 & 6 \end{bmatrix} \approx \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

How good is the approximation?

RMSE

Evaluate Root-Mean-Square-Error (RMSE)

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 1 & 1 & 2 \\ 1 & 3 & 2 & 2 \\ 1 & 2 & 1 & 0 \end{bmatrix}$$

$$\begin{aligned} RMSE &= \sqrt{\frac{1}{nm}\sum_{i,j}(M_{ij} - \sum_{k=1^d} U_{ik}V_{kj})^2} \\ &= \sqrt{\frac{1}{16}((2-1)^2 + (1-2)^2 + (4-1)^2 + \ldots + (6-0)^2)} \end{aligned}$$

Guessing missing entries in M

Steps:

1. Find U and V such that RMSE error between known entries in M and UV is small.
2. For a missing entry M_{ij} in M, let the guess be $\sum_{k=1}^{d} U_{ik}V_{kj}$

Guessing missing entries in M (contd.)

Let $M \approx UV$ be

$$M = \begin{bmatrix} 1 & 1 & \\ 2 & 1 & 3 \\ & 1 & 1 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$P = UV = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$RMSE = \sqrt{(M_{22} - P_{22})^2 + (M_{23} P_{24})^2} = \sqrt{2}$$

Guess for Missing Entries in M:

$$M_{13} = U_{11} V_{13} = 1 * 1 = 1$$
$$M_{31} = U_{31} V_{11} = 1 * 1 = 1$$

Computation of U and V

Main Steps:

1. Normalize Utility Matrix M
2. Initialize U and V
3. Compute each element of U and V by optimizing RMSE
4. Repeat the last step till convergence

We need to define convergence - for example the matrix entries don't change significantly, execute only for a fixed number of iterations, ...

Computation of U and V (contd.)

Let $M = \begin{bmatrix} 1 & 1 & \\ 2 & 1 & 3 \\ & 1 & 1 \end{bmatrix}$

Assume that we want to approximate M by rank 1 matrices U and V, so that $M \approx UV$

Initialize: $U = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ and $V = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

Let us optimize entry U_{11}

Problem: Find the best value of x so that the RMSE error between (the known entries) of M and $P = \begin{bmatrix} x \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ is minimum.

Computation of U and V (contd.)

$$P = \begin{bmatrix} x \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} x & x & x \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$RMSE(M, P) = \sqrt{(1-x)^2 + (1-x)^2 + 5}$$

To minimize RMSE, we need to choose $x = 1$

Now consider the best value for U_{21}:

$$P = \begin{bmatrix} 1 \\ x \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x & x & x \\ 1 & 1 & 1 \end{bmatrix}$$

$$RMSE(M, P) = \sqrt{(2-x)^2 + (1-x)^2 + (3-x)^2}$$

To minimize RMSE, we need to choose $x = 2$

Computation of U and V (contd.)

Now consider the best value for V_{13}:

$$P = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & x \end{bmatrix} = \begin{bmatrix} 1 & 1 & x \\ 2 & 2 & 2x \\ 1 & 1 & x \end{bmatrix}$$

$$RMSE(M, P) = \sqrt{(1-2)^2 + (3-2x)^2 + (1-x)^2}$$

To minimize RMSE, we need to choose $x = 7/5$

So far we have $M \approx UV$ as

$$M = \begin{bmatrix} 1 & 1 & \\ 2 & 1 & 3 \\ & 1 & 1 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 7/5 \end{bmatrix}$$

RMSE$(M, UV) \approx 1.09$

We may continue the improvement till RMSE converges.

Conclusions

1. Utility Matrix M: Users $\times$ Items ranking matrix
2. How to guess missing entries?
3. Content-based Filtering - Categorize items
4. Collaborative Filtering - Finding similar users/items
5. M as product of low rank matrices $M = UV$
6. Soon: We will express $M = U\Sigma V^T$ via Singular Value Decomposition. This has several interesting properties, including a low-rank approximation
7. Extra Reading: Netflix Challenge