UFMG UNIVERSIDADE FEDERAL DE MINAS GERAIS

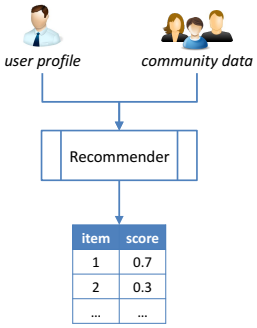
Recommender Systems

Similarity-based Recommendation

Rodrygo Santos
rodrygo@dcc.ufmg.br

How to recommend?

UFMG COMPUTER SCIENCE



Collaborative filtering

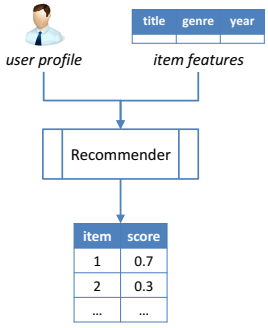
"tell me what's popular among my peers"

- Leverages item ratings
- Agnostic to item content

*Applicable to **any kind of item** (e.g., text, audio)*

How to recommend?

UFMG COMPUTER SCIENCE



Content-based filtering

"show me more of the same what I've liked"

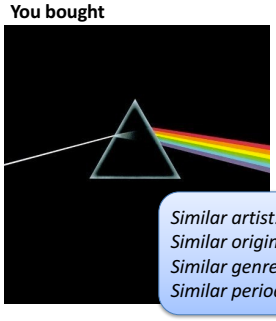
- Leverages item content
- Agnostic to item ratings

*Applicable for items with few or **no ratings***

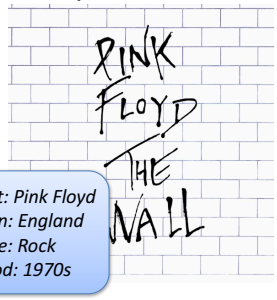
Content-based recommendation

UFMG COMPUTER SCIENCE

You bought



You may like



Similar artist: Pink Floyd
Similar origin: England
Similar genre: Rock
Similar period: 1970s

Vector representation

UFMG COMPUTER SCIENCE

- Each item is a vector
  - One component for each term
  - High dimensionality
- Each user is a vector
  - Some combination of item vectors

How to weigh term occurrences?

Presence-based weighing

UFMG COMPUTER SCIENCE

- Each item is a vector in $\{0, 1\}^{|V|}$

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
ANTHONY	1	1	0	0	0	1
BRUTUS	1	1	0	1	0	0
CAESAR	1	1	0	1	1	1
CALPURNIA	0	1	0	0	0	0
CLEOPATRA	1	0	0	0	0	0
MERCY	1	0	1	1	1	1
WORSER	1	0	1	1	1	0
...						

How representative is a term?

Count-based weighing

UFMG COMPUTER SCIENCE

- Each item is a vector in $N^{|V|}$

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
ANTHONY	157	73	0	0	0	1
BRUTUS	4	157	0	2	0	0
CAESAR	232	227	0	2	1	0
CALPURNIA	0	10	0	0	0	0
CLEOPATRA	57	0	0	0	0	0
MERCY	2	0	3	8	5	8
WORSER	2	0	1	1	1	5
...						

How discriminative is a term?

TF-IDF

UFMG COMPUTER SCIENCE

- Given a term *i* and an item *j*
  - TF_{ij}: term frequency of *i* in *j*
  - IDF_i: inverse document frequency of *i*

$$IDF_i = \log \frac{n}{n_i}$$

    - *n*: total number of items in the collection
    - n_i: number of items where term *i* appears
  - *TF*IDF*: weight given to each term

TF-IDF-based weighing

UFMG COMPUTER SCIENCE

- Each item is a vector in $R^{|V|}$

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
ANTHONY	5.25	3.18	0.0	0.0	0.0	0.35
BRUTUS	1.21	6.10	0.0	1.0	0.0	0.0
CAESAR	8.59	2.54	0.0	1.51	0.25	0.0
CALPURNIA	0.0	1.54	0.0	0.0	0.0	0.0
CLEOPATRA	2.85	0.0	0.0	0.0	0.0	0.0
MERCY	1.51	0.0	1.90	0.12	5.25	0.88
WORSER	1.37	0.0	0.11	4.15	0.25	1.95
...						

Vector representation

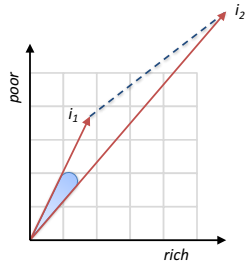
UFMG COMPUTER SCIENCE

- Each item is a vector
  - One component for each term
  - High dimensionality
- Each user is a vector
  - Some combination of item vectors

How to compute similarity?

Computing similarities

UFMG COMPUTER SCIENCE



i_1: "social inequality raises"
i_2: "rich-poor gap grows"

- Euclidean distance
  - Distance between vectors' endpoints
- Cosine
  - Angular distance between vectors

Which similarity?

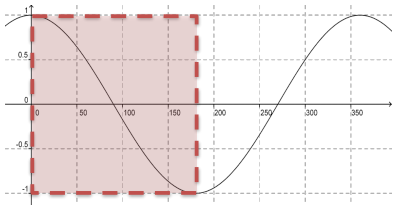
Angle vs. distance

UFMG COMPUTER SCIENCE

- Thought experiment
  - Take a term vector i_1
  - Append i_1 to itself, forming i_2 (i.e., $i_2 = 2\ i_1$)
- "Semantically", i_1 *and* i_2 are equivalent
  - The angle between the two vectors is 0…
    - High (actually, maximal) similarity
  - … the Euclidean distance can be quite large
    - Low similarity

From angles to cosines

- Cosine is a monotonically decreasing function of the angle for the interval [0º, 180º]



Smaller the angle, larger the cosine

vs.

Larger the angle, smaller the cosine

Cosine similarity

$$sim(\vec{u},\vec{v}) = cos(\vec{u},\vec{v}) = \frac{\vec{u}\cdot\vec{v}}{|\vec{u}|\times|\vec{v}|} = \frac{\sum_{i=1}^{|V|} u_i v_i}{\sqrt{\sum_{i=1}^{|V|} u_i^2}\sqrt{\sum_{i=1}^{|V|} v_i^2}}$$

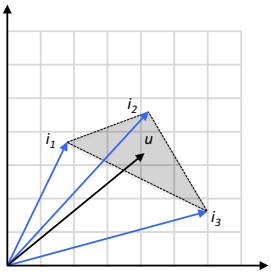
- u and v are term-weight vectors
  - u_i is the TF-IDF weight of term i in item u
  - v_i is the TF-IDF weight of term i in item v
  - $|u|$ and $|v|$ are the lengths of u and v

Quick recap

- We know how to represent items
  - Each item is a vector over terms
- We know how to compute vector similarities
  - Cosine of the angle between the vectors
- ***We can now produce recommendations***
  - Rank items by their similarity to the user

How to represent the user?

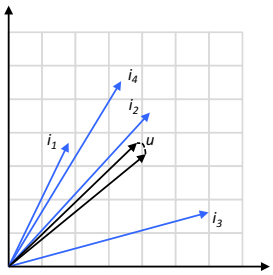
Representing the user



- User has rated
  - i_1: ★★★
  - i_2: ★★★★★
  - i_3: ★★★
- User prototype

$$\vec{u} = 3\vec{i}_1 + 5\vec{i}_2 + 3\vec{i}_3$$

Incremental updates



- User has rated
  - i_1: ★★★
  - i_2: ★★★★★
  - i_3: ★★★
  - i_4: ★★★★★
- User prototype

$$\vec{u} = 3\vec{i}_1 + 5\vec{i}_2 + 3\vec{i}_3 + 5\vec{i}_4$$

Relevance feedback

- The user marks items as relevant/nonrelevant
  - Relevant/nonrelevant can be viewed as classes
  - For each new item, the user decides which of these two classes is correct
- The system then uses these class assignments to build a better model of the user's need
  - ... in the hope of returning better items

Rocchio recommendation

UFMG COMPUTER SCIENCE

- Each item is a vector $\vec{i}$
  - One component for each term
- Each user is a vector $\vec{u}$

$$\vec{u} = \frac{1}{|R_u|} \sum_{j \in R_u} r_{uj}\, \vec{j}$$

R_u: items rated by user u
r_{uj}: rating of user u to item j

- Prediction score

$$sim(\vec{u}, \vec{i}) = cos(\vec{u}, \vec{i})$$

Rocchio example

UFMG COMPUTER SCIENCE

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	1		4		5		2			5
u_2		2		3		3	5		1	
u_3	4				4		3			2
u_4		2	1	5			1	3		3
u_5	3		4		3	5			4	

(input) utility matrix

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
t_1	2	0	1.5	0.5	1	2.5	0	7.5	3.2	1
t_2	1	2	10	1	0	16	20	0	8	5
t_3	0	2	1	1	3	5	1	2	1	3
t_4	4	1	0	2	1	0	1	2	1	0

(input) item feature matrix

Rocchio example

UFMG COMPUTER SCIENCE

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	1		4		5		2			5
u_2		2		3		3	5		1	
u_3	4				4		3			2
u_4		2	1	5			1	3		3
u_5	3		4		3	5			4	

Computing the user feature matrix

$$u_{11} = \frac{1\times2 + 4\times1.5 + 5\times1 + 2\times0 + 5\times1}{5} = 3.6$$

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
t_1	2	0	1.5	0.5	1	2.5	0	7.5	3.2	1
t_2	1	2	10	1	0	16	20	0	8	5
t_3	0	2	1	1	3	5	1	2	1	3
t_4	4	1	0	2	1	0	1	2	1	0

	u_1	u_2	u_3	u_4	u_5
t_1	u_{11}				
t_2					
t_3					
t_4					

Rocchio example

UFMG COMPUTER SCIENCE

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	1		4		5		2			5
u_2		2		3		3	5		1	
u_3	4				4		3			2
u_4		2	1	5			1	3		3
u_5	3		4		3	5			4	

Computing the user feature matrix

$$u_{12} = \frac{1\times1 + 4\times10 + 5\times0 + 2\times20 + 5\times5}{5} = 21.2$$

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
t_1	2	0	1.5	0.5	1	2.5	0	7.5	3.2	1
t_2	1	2	10	1	0	16	20	0	8	5
t_3	0	2	1	1	3	5	1	2	1	3
t_4	4	1	0	2	1	0	1	2	1	0

	u_1	u_2	u_3	u_4	u_5
t_1	3.6				
t_2	u_{12}				
t_3					
t_4					

Rocchio example

UFMG COMPUTER SCIENCE

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	1		4		5		2			5
u_2		2		3		3	5		1	
u_3	4				4		3			2
u_4		2	1	5			1	3		3
u_5	3		4		3	5			4	

Computing the user feature matrix

$$u_{13} = \frac{1\times0 + 4\times1 + 5\times3 + 2\times1 + 5\times3}{5} = 7.2$$

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
t_1	2	0	1.5	0.5	1	2.5	0	7.5	3.2	1
t_2	1	2	10	1	0	16	20	0	8	5
t_3	0	2	1	1	3	5	1	2	1	3
t_4	4	1	0	2	1	0	1	2	1	0

	u_1	u_2	u_3	u_4	u_5
t_1	3.6				
t_2	21.2				
t_3	u_{13}				
t_4					

Rocchio example

UFMG COMPUTER SCIENCE

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	1		4		5		2			5
u_2		2		3		3	5		1	
u_3	4				4		3			2
u_4		2	1	5			1	3		3
u_5	3		4		3	5			4	

Computing the user feature matrix

$$u_{14} = \frac{1\times4 + 4\times0 + 5\times1 + 2\times1 + 5\times0}{5} = 2.2$$

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
t_1	2	0	1.5	0.5	1	2.5	0	7.5	3.2	1
t_2	1	2	10	1	0	16	20	0	8	5
t_3	0	2	1	1	3	5	1	2	1	3
t_4	4	1	0	2	1	0	1	2	1	0

	u_1	u_2	u_3	u_4	u_5
t_1	3.6				
t_2	21.2				
t_3	7.2				
t_4	u_{14}				

Rocchio example

UFMG COMPUTER SCIENCE

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	1		4		5		2			5
u_2		2		3		3	5		1	
u_3	4				4		3			2
u_4		2	1	5			1	3		3
u_5	3		4		3	5			4	

How to recommend to user u_1?

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
t_1	2	0	1.5	0.5	1	2.5	0	7.5	3.2	1
t_2	1	2	10	1	0	16	20	0	8	5
t_3	0	2	1	1	3	5	1	2	1	3
t_4	4	1	0	2	1	0	1	2	1	0

	u_1	u_2	u_3	u_4	u_5
t_1	3.6	2.4	3.5	4.9	8.1
t_2	21.2	32.6	18.5	9.0	31.0
t_3	7.2	5.6	5.3	4.3	8.4
t_4	2.2	2.8	5.8	3.2	3.8

Rocchio example

UFMG COMPUTER SCIENCE

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	1	r_{12}	4		5		2			5
u_2		2		3		3	5		1	
u_3	4				4		3			2
u_4		2	1	5			1	3		3
u_5	3		4		3	5			4	

Recommending to u_1

$\hat{r}_{12} = cos(\vec{u}_1, \vec{i}_2)$

$$\hat{r}_{12} = \frac{3.6\times0 + 21.2\times2 + 7.2\times2 + 2.2\times1}{\sqrt{3.6^2+21.2^2+7.2^2+2.2^2}\sqrt{0^2+2^2+2^2+1^2}}$$

$\hat{r}_{12} = 0.86$

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
t_1	2	0	1.5	0.5	1	2.5	0	7.5	3.2	1
t_2	1	2	10	1	0	16	20	0	8	5
t_3	0	2	1	1	3	5	1	2	1	3
t_4	4	1	0	2	1	0	1	2	1	0

	u_1	u_2	u_3	u_4	u_5
t_1	3.6	2.4	3.5	4.9	8.1
t_2	21.2	32.6	18.5	9.0	31.0
t_3	7.2	5.6	5.3	4.3	8.4
t_4	2.2	2.8	5.8	3.2	3.8

Rocchio example

UFMG COMPUTER SCIENCE

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	1	.86	4	.61	5	.99	2	.25	.96	5
u_2		2		3		3	5		1	
u_3	4				4		3			2
u_4		2	1	5			1	3		3
u_5	3		4		3	5			4	

Recommending to u_1

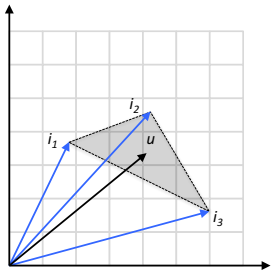
- $i_6 : 0.99$
- $i_9 : 0.96$
- $i_2 : 0.86$
- $i_4 : 0.61$
- $i_8 : 0.25$

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
t_1	2	0	1.5	0.5	1	2.5	0	7.5	3.2	1
t_2	1	2	10	1	0	16	20	0	8	5
t_3	0	2	1	1	3	5	1	2	1	3
t_4	4	1	0	2	1	0	1	2	1	0

	u_1	u_2	u_3	u_4	u_5
t_1	3.6	2.4	3.5	4.9	8.1
t_2	21.2	32.6	18.5	9.0	31.0
t_3	7.2	5.6	5.3	4.3	8.4
t_4	2.2	2.8	5.8	3.2	3.8

Representing the user

UFMG COMPUTER SCIENCE

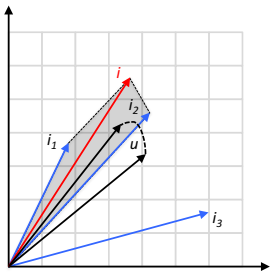


- User has rated
  - i_1: ★★★
  - i_2: ★★★★★
  - i_3: ★★★
- User prototype

$$\vec{u} = 3\vec{i}_1 + 5\vec{i}_2 + 3\vec{i}_3$$

Representing the user

UFMG COMPUTER SCIENCE



- User has rated
  - i_1: ★★★
  - i_2: ★★★★★
  - i_3: ★★★
- User prototype

$$\vec{u} = 3\vec{i}_1 + 5\vec{i}_2 + 3\vec{i}_3$$

k-NN recommendation

UFMG COMPUTER SCIENCE

- Each item is a vector $\vec{i}$
  - One component for each term
- Each user is a vector $\vec{u}$

$$\vec{u}_i = \frac{1}{|N_{ui}|}\sum_{j \in N_{ui}} r_{uj}\,\vec{j}$$

N_{ui}: neighbor items of i rated by user u
r_{uj}: rating of user u to item j

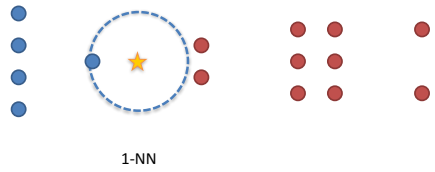
- Prediction score

$$sim(\vec{u}, \vec{i}) = cos(\vec{u}_i, \vec{i})$$

Summary

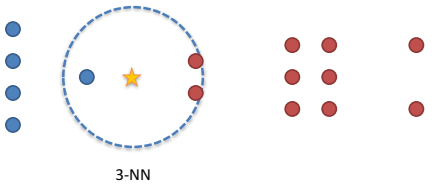
- Rocchio is a ***nearest centroid*** recommender
  - Items are matched against the user centroid(s)
  - Different items will use the same centroid(s)
- The former is an example of lazy, ***nearest neighbor*** content-based recommendation
  - Neighbors are chosen on-demand for each item
  - Different items will have different neighbors
    - *More effective, but way less efficient*

Closing example



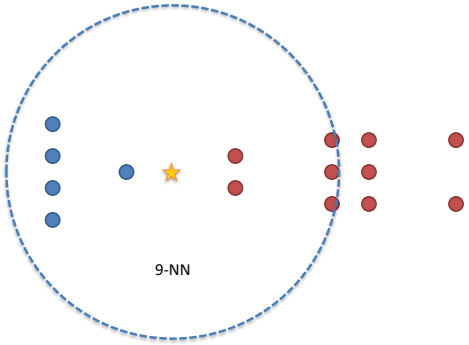
1-NN

Closing example



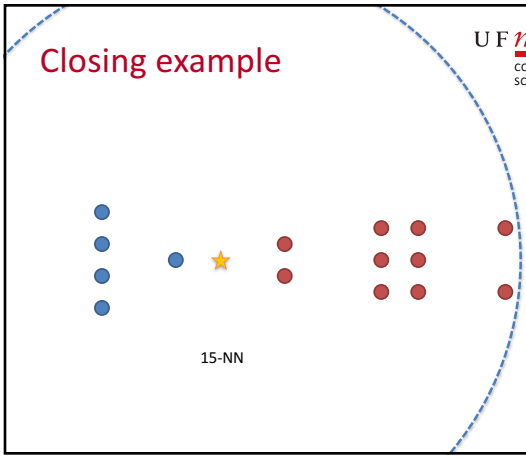
3-NN

Closing example



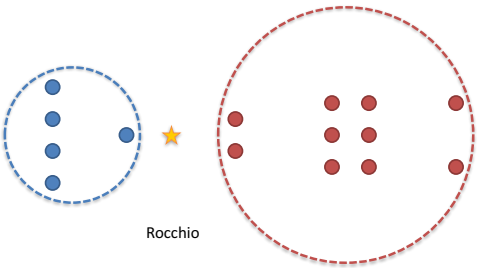
9-NN

Closing example



15-NN

Closing example



Rocchio