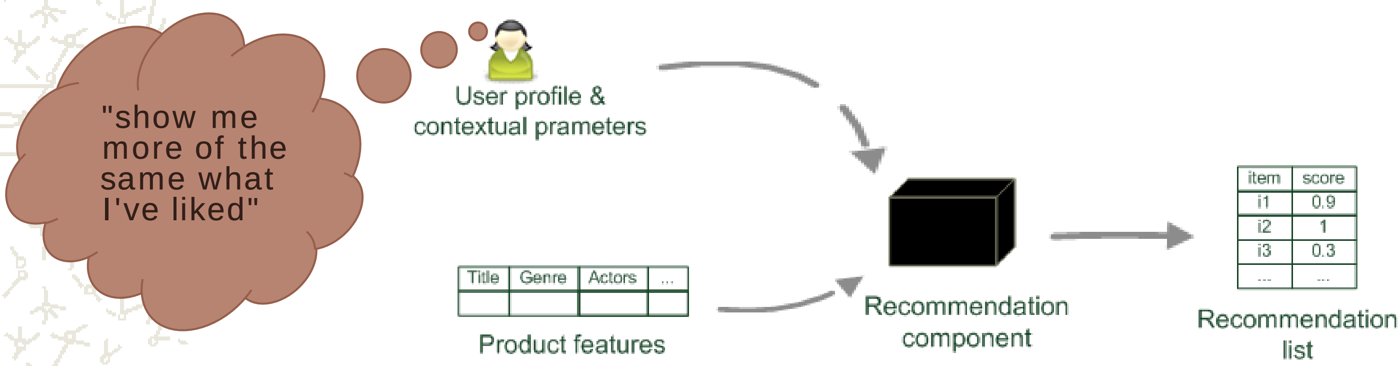
Implementing a Content-based Recommendation Engine

Costas Mourlas
Associate Professor
Univ. of Athens

Content-based recommendation

- **While CF – methods do not require any information about the items,**
  - it might be reasonable to exploit such information; and
  - recommend fantasy novels to people who liked fantasy novels in the past
- **What do we need:**
  - some information about the available items such as the genre ("content")
  - some sort of *user profile* describing what the user likes (the preferences)
- **The task:**
  - learn user preferences
  - locate/recommend items that are "similar" to the user preferences



How to Create a Content Based Recommender?

Similarity-Based Retrieval

1. Decide for an Item and a User Representation
2. Select a suitable Similarity / Distance Function
3. Compute the Distances between all the Items
4. Find the neighborhood of every Item
   k-nearest-neighbor method (kNN)
5. Compute the Recommendations
   make the neighbors “vote” for the unseen Items
   select Items with the higher values

What is the "content"?

- Content of items can also be represented as text documents.
  - With textual descriptions of their basic characteristics.
  - Structured: Each item is described by the same set of attributes

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism

  - Unstructured: free-text description.

Representing multi-valued attributes as set of Keywords

- **Item representation**

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism

- **User profile**

Title	Genre	Author	Type	Price	Keywords
…	Fiction	Brunonia, Barry, Ken Follett	Paperback	25.65	Detective, murder, New York

Representing Users / Users Profile

- Simple Approach

Title	Genre	Author	Type	Price	Keywords
...	Fiction, Suspense	Brunonia, Barry, Ken Follett	Paperback	25.65	Detective, murder, New York

1. Explicitly ask users for a desired price range or a set of preferred genres.
2. Asking Users to rate a set of items and then construct a preference profile for the user.

Compute Similarity between Items and Users

- **Item representation**

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism

- **User profile**

Title	Genre	Author	Type	Price	Keywords
...	Fiction	Brunonia, Barry, Ken Follett	Paperback	25.65	Detective, murder, New York

$keywords(b_j)$ describes Book b_j with a set of keywords

- Simple approach
  - Compute the similarity of an unseen item with the user profile based on the keyword overlap (e.g. using the Dice coefficient)
  - Or use and combine multiple metrics

$$\frac{2 \times |keywords(b_i) \cap keywords(b_j)|}{|keywords(b_i)| + |keywords(b_j)|}$$

Computing Similarity of Multi-valued Attributes of Items

- Dice's coefficient

$$2\frac{|Q \cap D|}{|Q| + |D|}$$

- Jaccard's coefficient

$$\frac{|Q \cap D|}{|Q \cup D|}$$

Moving from the Set Space (where a set of Keywords can be the value of Item's Multi-Valued Attributes) to the Vector Space

From similarity of sets to the similarity of vectors

Rpresenting Items with Keywords as Attributes –Binary Values

Doc-ID	recommender	intelligent	learning	school
1	1	1	1	0
2	0	0	1	1
3	1	1	0	0
4	1	0	1	1
5	0	0	0	1
6	1	1	0	0

Items as vectors
Doc-ID1<1,1,1,0> , Doc-ID2<0,0,1,1>, ...

Rpresenting Items with Keywords as Attributes –Real Values

TF.IDF Representation of Documents -> From Unstructured Representation of Documents (Text) to Structural Representation

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

Items as vectors:
Antony_and_Cleopatra<5.25,1.21,8.59,0,2.85,1.51,1.37>

Representing Items with Attributes/ Characteristics – Single valued Attributes

Structural Representation of Items							
Video / Attribute	**Action**	**Drama**	**Humor**	**Romantic**	**Violence**	**Suspense**	**Musical**
(A) Silence of the Lambs	**0**	**7**	**3**	**1**	**9**	**10**	**0**
(B) Seven	**5**	**5**	**1**	**2**	**10**	**9**	**5**
(C) Cape Fear	**5**	**7**	**4**	**5**	**9**	**9**	**3**
(D) Casablanca	**2**	**10**	**5**	**0**	**1**	**8**	**0**
(E) Waterboy	**4**	**2**	**6**	**3**	**4**	**3**	**1**
(F) L.A. Confidential	**8**	**9**	**6**	**6**	**9**	**9**	**6**
(G) West Side Story	**3**	**5**	**4**	**0**	**1**	**3**	**1**

Items as vectors:
A<0,7,3,1,9,10,0>
B<5,5,1,2,10,9,5>
C<5,7 ,4,5,9,9,3>
D<2,10,5,0,1,8,0>, …..

How to represent the User Profile

- In the vector space approach, Users are asked to rate a set of items. The history of this rating represents the profile of the User.

Utility Matrix

	King Kong	LOTR	Matrix	Nacho Libre
Alice	1		0.2	
Bob		0.5		0.3
Carol	0.2		1	
David				0.4

1. Decide for an Item and a User Representation
2. Select a suitable Similarity / Distance Function
   (suitable for the vector space approach)
3. Compute the Distances between all the Items
4. Find the neighborhood of every Item
   k-nearest-neighbor method (kNN)
5. Compute the Recommendations
   make the neighbors "vote" for the unseen Items
   select Items with the higher values

How to Compute Similarity of Items in this case?

Euclidean distances:

- Calculates the shortest path between two points.

$$d = |x - y| = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

or

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_1)^2}$$

- Sum of distances along each dimension (Manhattan Distance)

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

- **Usual similarity metric to compare vectors: Cosine similarity (angle)**
  - Cosine similarity is calculated based on the angle between the vectors
    - $sim(\vec{a}, \vec{b}\,) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| * |\vec{b}|}$

Compute the distances between Items

An example of content-based filtering							
Video / Attribute	Action	Drama	Humor	Romantic	Violence	Suspense	Musical
(A) Silence of the Lambs	0	7	3	1	9	10	0
(B) Seven	5	5	1	2	10	9	5
(C) Cape Fear	5	7	4	5	9	9	3
(D) Casablanca	2	10	5	0	1	8	0
(E) Waterboy	4	2	6	3	4	3	1
(F) L.A. Confidential	8	9	6	6	9	9	6
(G) West Side Story	3	5	4	0	1	3	1

Euclidean Distance:

	A	B	C	D	E	F	G
A		7,810249676	7,211102551	9,273618	11,35782	11,78983	11,35782
B			5,196152423	12,68858	11,13553	8,246211	12,24745
C				10,86278	9,949874	5,196152	11,7047
D					10,63015	13,22876	7,28011
E						12,64911	5,656854
F							14,3527
G							

1. Decide for an Item and a User Representation
2. Select a suitable Similarity / Distance Function
   (suitable for the vector space approach)
3. Compute the Distances between all the Items
4. Find the neighborhood of every Item
   k-nearest-neighbor method (kNN)
5. Compute the Recommendations
   make the neighbors "vote" for the unseen Items
   select Items with the higher values

Find the neighborhood of every Item

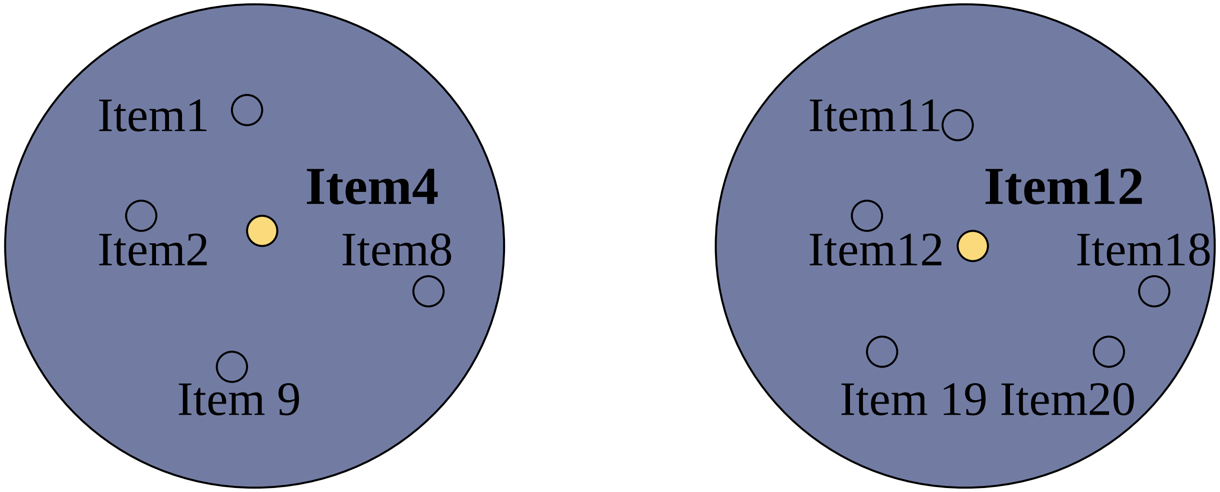
K-nearest-neighbor method (kNN)

Simple method: nearest neighbors

- Given a set of documents D already rated by the user (like/dislike)
  - Either explicitly via user interface
  - Or implicitly by monitoring user's behavior
- Find the n nearest neighbors of an not-yet-seen item i in D
  - Use similarity measures (like cosine similarity) to capture similarity of two documents
- Take these neighbors to predict a rating for i
  - e.g. $k = 5$ most similar items to i.
    4 of k items were liked by current user ➡ item i will also be liked by this user

Variations of kNN method

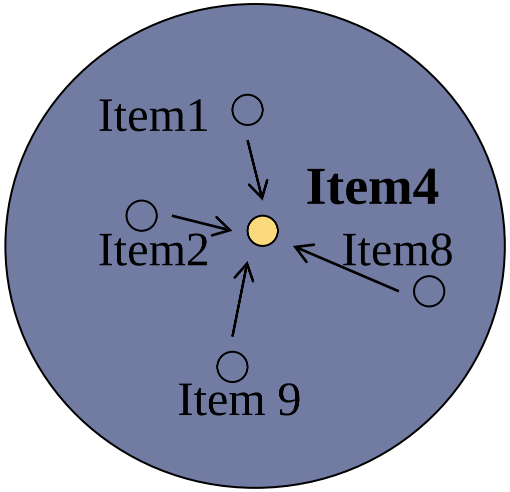
- Variations:
  - Varying neighborhood size k
  - lower/upper similarity thresholds to prevent system from recommending items the user already has seen
- Good to model short-term interests / follow-up stories
- Used in combination with method to model long-term preferences



1. Decide for an Item and a User Representation
2. Select a suitable Similarity / Distance Function
   (suitable for the vector space approach)
3. Compute the Distances between all the Items
4. Find the neighborhood of every Item
   k-nearest-neighbor method (kNN)
5. Compute the Recommendations
   make the neighbors “vote” for the unseen Items
   select Items with the higher values

Compute the Recommendations

- make the neighbors "vote" for the unseen Items



Three approaches:

1. Take the average of the rated items that belong to the neighborhood

2. Weight the votes based on the degree of similarity

3. Let the latest (more recent) ratings to vote -> user's short term interest

Other Ideas???

On feature selection

- **process of choosing a subset of available terms**
- **different strategies exist for deciding which features to use**
  - feature selection based on domain knowledge and lexical information from WordNet (Pazzani and Billsus 1997)
  - frequency-based feature selection to remove words appearing "too rare" or "too often" (Chakrabarti 2002)
- **Not appropriate for larger text corpora**
  - Better to
    - evaluate value of individual features (keywords) independently and
    - construct a ranked list of "good" keywords.
- **Typical measure for determining utility of keywords: e.g. X^2, mutual information measure or Fisher's discrimination index**

Limitations of content-based recommendation methods

- Keywords alone may not be sufficient to judge quality/relevance of a document or web page
  - up-to-date-ness, usability, aesthetics, writing style
  - content may also be limited / too short
  - content may not be automatically extractable (multimedia)
- Ramp-up phase required
  - Some training data is still required
  - Web 2.0: Use other sources to learn the user preferences
- Overspecialization
  - Algorithms tend to propose "more of the same"
  - Or: too similar news items

Discussion & summary

- In contrast to collaborative approaches, content-based techniques do not require user community in order to work
- Presented approaches aim to learn a model of user's interest preferences based on explicit or implicit feedback
  - Deriving implicit feedback from user behavior can be problematic
- Evaluations show that a good recommendation accuracy can be achieved with help of machine learning techniques
  - These techniques do not require a user community
- Danger exists that recommendation lists contain too many similar items
  - All learning techniques require a certain amount of training data
  - Some learning methods tend to overfit the training data
- Pure content-based systems are rarely found in commercial environments

Literature

- [Michael Pazzani and Daniel Billsus 1997] Learning and revising user profiles: The identification of interesting web sites, Machine Learning 27 (1997), no. 3, 313-331.
- [Soumen Chakrabarti 2002] Mining the web: Discovering knowledge from hyper-text data, Science & Technology Books, 2002.