Recommender Systems

Instructor: Junghye Lee

Department of Industrial Engineering

junghyelee@unist.ac.kr

Contents

Content-based
Recommender Systems

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

What is the "content"?

- **The genre is actually not part of the content of a book**
- **Most CB-recommendation methods originate from Information Retrieval (IR) field:**
  - goal is to find and rank interesting text documents (news articles, web pages)
  - the item descriptions are usually automatically extracted (important words)
- **Fuzzy border between content-based and "knowledge-based" RS**
- **Here:**
  - classical IR-based methods based on keywords
  - no expert recommendation knowledge involved
  - user profile (preferences) are rather learned than explicitly elicited

Content representation and item similarities

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
...					

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

- **Simple approach**
  - Compute the similarity of an unseen item with the user profile based on the keyword overlap (e.g. using the Dice coefficient)
    - $sim(b_i, b_j) = \frac{2 * |keywords(b_i) \cap keywords(b_j)|}{|keywords(b_i)| + |keywords(b_j)|}$
  - Or combine multiple metrics in a weighted approach

Term-Frequency - Inverse Document Frequency (TF-IDF)

- **Simple keyword representation has its problems**
  - in particular when automatically extracted as
    - not every word has similar importance
    - longer documents have a higher chance to have an overlap with the user profile
- **Standard measure: TF-IDF**
  - Encodes text documents in multi-dimensional Euclidean space
    - weighted term vector
  - TF: Measures, how often a term appears (density in a document)
    - assuming that important terms appear more often
    - normalization has to be done in order to take document length into account
  - IDF: Aims to reduce the weight of terms that appear in all documents

TF-IDF

- **Compute the overall importance of keywords**
  - Given a keyword *i* and a document *j*

    $$TFIDF(i,j) = TF(i,j) * IDF(i)$$

- **Term frequency (TF)**
  - Let $freq(i,j)$ number of occurrences of keyword *i* in document *j*
  - Let $maxOthers(i,j)$ denote the highest number of occurrences of another keyword of *i'* in document *j*
  - $TF(i,j) = \frac{freq(i,j)}{maxOthers(i',j)}$

- **Inverse Document Frequency (IDF)**
  - N: number of all recommendable documents
  - $n(i)$: number of documents in which keyword *i* appears
  - $IDF(i) = log\,\frac{N}{n(i)}$

Example TF-IDF 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

Figure taken from http://informationretrieval.org

More on the vector space model

- **Vectors are usually long and sparse**
- **Improvements**
  - remove stop words ("a", "the", ..)
  - use stemming
  - size cut-offs (only use top n most representative words, e.g. around 100)
  - use additional knowledge, use more elaborate methods for feature selection
  - detection of phrases as terms (such as United Nations)
- **Limitations**
  - semantic meaning remains unknown
  - example: usage of a word in a negative context
    - "there is nothing on the menu that a vegetarian would like.."
- **Usual similarity metric to compare vectors: Cosine similarity (angle)**

Recommending items

- **Simple method: nearest neighbors**
  - Given a set of documents D already rated by the user (like/dislike)
    - Find the *k* nearest neighbors of a not-yet-seen item *i* in D
    - Take these ratings to predict a rating/vote for *i*
      - Variations: neighborhood size, lower/upper similarity thresholds
  - Good to model short-term interests / follow-up stories
  - Used in combination with method to model long-term preferences

Improvements

- **Side note: Conditional independence of events does in fact not hold**
  - "New York", "Hong Kong"
  - Still, good accuracy can be achieved
- **Boolean representation simplistic**
  - positional independence assumed
  - keyword counts lost
- **Other linear classification algorithms (machine learning) can be used**
  - e.g., logistic regression, support vector machine

Limitations of content-based recommendation methods

- **Keywords alone may not be sufficient to judge quality/relevance of a document or web page**
  - up-to-dateness, 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" (too similar items)

Knowledge-Based Recommender Systems

Knowledge-Based Recommendation

- **Explicit domain knowledge**
  - Sales knowledge elicitation from domain experts
  - System mimics the behavior of experienced sales assistant
  - Best-practice sales interactions
  - Can guarantee “correct” recommendations (determinism) with respect to expert knowledge

- **Conversational interaction strategy**
  - Opposed to one-shot interaction
  - Elicitation of user requirements
  - Transfer of product knowledge (“educating users”)

Limitations of knowledge-based recommendation methods

- **Cost of knowledge acquisition**
  - From domain experts
  - From users
  - From web resources

- **Accuracy of preference models**
  - Very fine granular preference models require many interaction cycles with the user or sufficient detailed data about the user
  - Preferences may depend on each other
  - Collaborative filtering models the preference of a user implicitly
    - Hybrid recommender systems

- **Instability of preference models**
  - e.g.) asymmetric dominance effects and decoy items

Recommender systems: technique comparison

	Pros	Cons
Collaborative	Nearly no ramp-up effort, serendipity of results, learns market segments	Requires some form of rating feedback, cold start for new users and new items
Content-based	No community required, comparison between items possible	Content-descriptions necessary, cold start for new users, no surprises
Knowledge-based	Deterministic recommendations, assured quality, no cold-start, can resemble sales dialogue	Knowledge engineering effort, basically static, does not react to short-term trends

Evaluation of Recommender Systems

Evaluating Recommender Systems

- **A myriad of techniques has been proposed, but**
  - Which one is the best in a given application domain?
  - What are the success factors of different techniques?
  - Comparative analysis based on an optimality criterion?

- **Research questions are:**
  - Is a RS efficient with respect to a specific criteria like accuracy, user satisfaction, response time, serendipity, online conversion, ramp-up efforts, ….
  - Do customers like/buy recommended items?
  - Do customers buy items they otherwise would have not?
  - Are they satisfied with a recommendation after purchase?

Evaluation in information retrieval (IR)

- **Recommendation is viewed as information retrieval task:**
  - Retrieve (recommend) all items which are predicted to be “good”.

- **Ground truth established by human domain experts**

		Reality	
		Actually Good	Actually Bad
Prediction	Rated Good	True Positive (TP)	False Positive (FP)
	Rated Bad	False Negative (FN)	True Negative (TN)

Metrics: Precision and Recall

- **Precision: a measure of exactness, determines the fraction of relevant items retrieved out of all items retrieved**
  - e.g. the proportion of recommended movies that are actually good

$$Precision = \frac{tp}{tp + fp} = \frac{|good\ movies\ recommended|}{|\text{all recommendations}|}$$

- **Recall: a measure of completeness, determines the fraction of relevant items retrieved out of all relevant items**
  - e.g. the proportion of all good movies recommended

$$Recall = \frac{tp}{tp + fn} = \frac{|good\ movies\ recommended|}{|all\ good\ movies|}$$

F_1 Metric

- **The F_1 Metric attempts to combine Precision and Recall into a single value for comparison purposes.**
  - May be used to gain a more balanced view of performance

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

- **The F_1 Metric gives equal weight to precision and recall**
  - Other F_β metrics weight recall with a factor of β.

Metrics: Rank Score – position matters Evaluation in RS

For a user:

Actually good
Item 237
Item 899

hit@k

here, k=3

Recommended (predicted as good)
Item 345
Item 237
Item 187

- **Rank Score extends recall and precision to take “the positions of correct items in a ranked list” into account**
  - Relevant items are more useful when they appear earlier in the recommendation list
  - Particularly important in recommender systems as lower ranked items may be overlooked by users

Metrics: Rank Score

- **Rank Score is defined as the ratio of the Rank Score of the correct items to best theoretical Rank Score achievable for the user, i.e.**

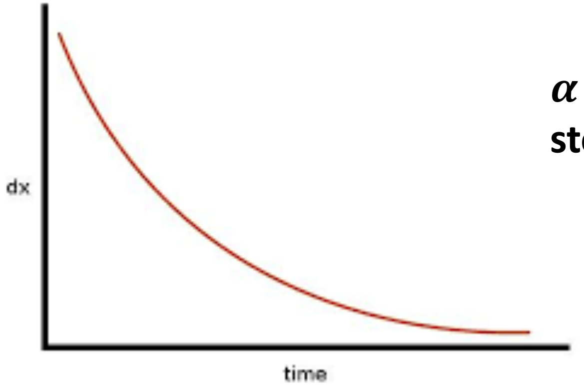
$$rankscore = \frac{rankscore_p}{rankscore_{\max}}$$

$$rankscore_p = \sum_{i \in h} 2^{-\frac{rank(i)-1}{\alpha}}$$

$$rankscore_{\max} = \sum_{i=1}^{|T|} 2^{-\frac{i-1}{\alpha}}$$

Where:

- *h* is the set of correctly recommended items, i.e. hits
- *rank* returns the position (rank) of an item
- *T* is the set of all items of interest
- α is the *ranking half life*, i.e. an exponential reduction factor



α decreases, steeper decay

Metrics: Liftindex

- **Assumes that ranked list is divided into 10 equal deciles S_i, where**

$$\sum_{i=1}^{10} S_i = |h|$$

  - Linear reduction factor

- **Liftindex:**

$$liftindex = \begin{cases} \dfrac{1 \times S_1 + 0.9 \times S_2 + \ldots + 0.1 \times S_{10}}{\sum_{i=1}^{10} S_i} & : \quad if\ |h| > 0 \\ 0 & : \quad else \end{cases}$$

» h **is the set of correct hits**

Metrics: Normalized Discounted Cumulative Gain

- **Discounted cumulative gain (DCG)**
  - Logarithmic reduction factor

$$DCG_{pos} = rel_1 + \sum_{i=2}^{pos} \frac{rel_i}{\log_2 i}$$

Where:

- *pos* denotes the position up to which relevance is accumulated
- rel_i returns the relevance of recommendation at position *i*

- **Idealized discounted cumulative gain (IDCG)**
  - Assumption that items are ordered by decreasing relevance

$$IDCG_{pos} = rel_1 + \sum_{i=2}^{|h|-1} \frac{rel_i}{\log_2 i}$$

- **Normalized discounted cumulative gain (nDCG)** $nDCG_{pos} \frac{DCG_{pos}}{IDCG_{pos}}$
  - Normalized to the interval [0..1]

Evaluation in RS

- **Datasets with items rated by users**
  - MovieLens datasets 100K-10M ratings
  - Netflix 100M ratings
- **Historic user ratings constitute ground truth**
- **Metrics measure error rate**
  - Mean Absolute Error (*MAE*) computes the deviation between predicted ratings and actual ratings

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - r_i|$$

  - Root Mean Square Error (*RMSE*) is similar to *MAE*, but places more emphasis on larger deviation

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2}$$

Establishing ground truth

- **IR measures are frequently applied, however**:

Offline experimentation	Online experimentation
Ratings, transactions	Ratings, feedback
Historic session (not all recommended items are rated)	Live interaction (all recommended items are rated)
Ratings of unrated items unknown, but interpreted as “bad” (default assumption, user tend to rate only good items)	“Good/bad” ratings of not recommended items are unknown
If default assumption does not hold: True positives may be too small False negatives may be too small	False/true negatives cannot be determined
Precision may increase Recall may vary	Precision ok Recall questionable

Results from offline experimentation have limited predictive power for online user behavior.

- **Offline/online: whether to run a new system on live users to collect new data**

Offline experimentation

- **Netflix competition**
  - Web-based movie rental
  - Prize of $1,000,000 for accuracy improvement (RMSE) of 10% compared to own Cinematch system.

- **Historical dataset**
  - ~480K users rated ~18K movies on a scale of 1 to 5
  - ~100M ratings
  - Last 9 ratings/user withheld
    - Probe set – for teams for evaluation
    - Quiz set – evaluates teams' submissions for leaderboard
    - Test set – used by Netflix to determine winner

Online experimentation

- **Effectiveness of different algorithms for recommending cell phone games**
  [Jannach, Hegelich 09]
- **Involved 150,000 users on a commercial mobile internet portal**
- **Comparison of recommender methods**

Details and results

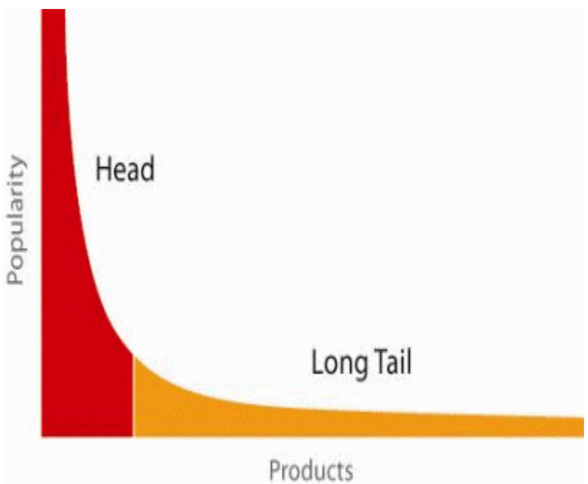
- **Recommender variants included:**
  - Item-based collaborative filtering
  - User-based collaborative filtering
  - Model-based collaborative filtering
  - Content-based recommendation
  - Hybrid recommendation
  - Top-rated items
  - Top-sellers

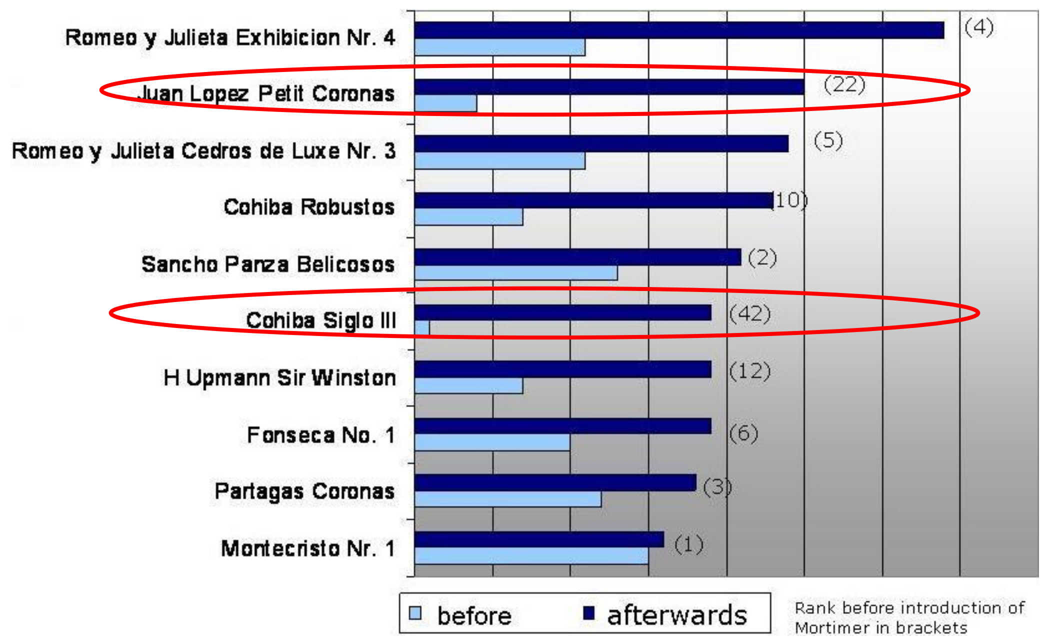
  } non-personalized

- **Findings:**
  - Personalized methods increased sales up to 3.6% compared to non-personalized
  - Choice of recommendation algorithm depends on user situation

Observational research

- **Increased demand in niches/long tail products**
  - Books ranked above 250.000 represent >29% of sales at Amazon, approx. 2.3 million books [Brynjolfsson et al., Mgt. Science, 2003]
  - Ex post from webshop data [Zanker et al., EC-Web, 2006]





Questions?