CHAPTER 6: RECOMMENDATION SYSTEMS

Subject: Introduction to data science

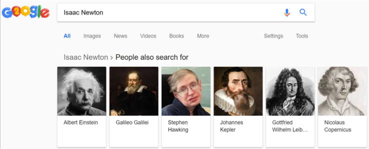
Instructor: Đinh Xuân Trường
Posts and Telecommunications Institute of Technology

Hanoi, 2024

What is recommender systems?

- **A recommender system (RS)** helps users that have no sufficient competence or time to evaluate the, potentially overwhelming, number of alternatives offered by a web site.
  - In their simplest form, RSs recommend to their users personalized and **ranked lists of items**

(a) Book recommender system.



(b) Google search.

What is recommender systems?

- Systems for recommending items (e.g. books, movies, CD's, web pages, newsgroup messages) to users based on examples of their preferences.
- Many websites provide recommendations (e.g. Amazon, NetFlix, Pandora).
- Recommenders have been shown to substantially increase sales at on-line stores.
- There are two basic approaches to recommending:
  - Collaborative Filtering (a.k.a. social filtering)
  - Content-based

Content

Concepts

- Collaborative Filtering.
  - The process of information filtering by collecting human judgments (ratings)
  - “word of mouth”
- User: Any individual who provides ratings to a system
- Items: Anything for which a human can provide a rating

Concepts

	Star Wars	Hoop Dreams	Contact	Titanic
Joe	5	2	5	4
John	2	5		3
Al	2	2	4	2
Nathan	5	1	5	?

The problem of collaborative filtering is to predict how well a user will like an item that he has not rated given a set of historical preference judgments for a community of users.

Uses for CF: User tasks

What tasks users may wish to accomplish

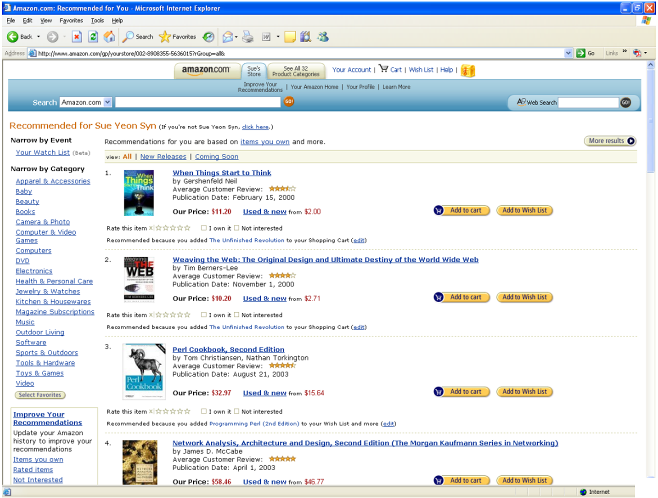
- Help me find new items I might like
- Advise me on a particular item
- Help me find a user (or some users) I might like
- Domain-specific tasks
- Help me find an item, new or not

Uses for CF: System tasks

What CF systems support

- Recommend items
  - Eg. Amazon.com
- Predict for a given item
- Constrained recommendations
  - Recommend from a set of items

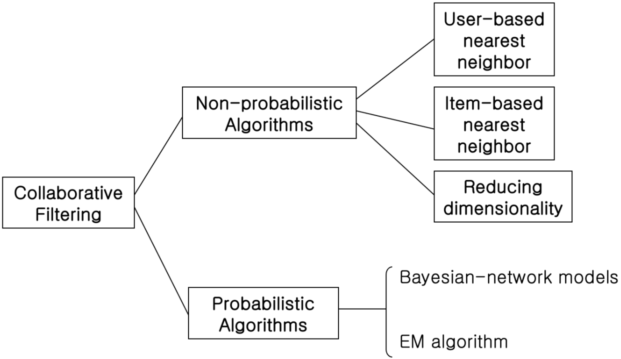
Uses for CF: Amazon.com



Uses for CF: Domains

- Many items
- Many ratings
- Many more users than items recommended
- Users rate multiple items
- For each user of the community, there are other users with common needs or tastes
- Item evaluation requires personal taste
- Items persists
- Taste persists
- Items are homogenous

Algorithms


Collaborative Filtering
Non−probabilistic Algorithms
User−based nearest neighbor
Item−based nearest neighbor
Reducing dimensionality
Probabilistic Algorithms
Bayesian−network models
EM algorithm

Algorithms: Non-probabilistic

- **User-based Nearest Neighbor**
  - Neighbor = similar users
  - Generate a prediction for an item i by analyzing ratings for i from users in u's neighborhood

$$pred(u,i) = \bar{r}_u + \frac{\sum_{n \subset neighbor(n)} sim(u,n).(r_{ni} - \bar{r}_n)}{\sum_{n \subset neighbor(n)} sim(u,n)} \quad (1)$$

Algorithms: Non-probabilistic

Item-based Nearest Neighbor

- Generate predictions based on similarities between items.
- Prediction for a user u and item i is composed of a weighted sum of the user u's ratings for items most similar to i.

$$pred(u,i) = \frac{\sum_{j \in ratedItems(u)} sim(i,j).(r_{ui})}{\sum_{j \in ratedItems(u)} sim(i,j)} \quad (2)$$

Algorithms: Non-probabilistic

- Dimensionality Reduction
  - Reduce domain complexity by mapping the item space to a smaller number of underlying dimensions.
  - Dimension may be latent topics or tastes.
  - Vector-based techniques
    - Vector decomposition
    - Principal component analysis
    - Factor analysis

Algorithms: Probabilistic

- Represent probability distributions
- Given a user u and a rated item i, the user assigned the item a rating of r : $p(r|u,i)$.

$$E(r|u,i) = \sum_r r.p(r|u,i) \tag{3}$$

- Bayesian-network models, Expectation maximization (EM) algorithm

Practical issues: Rating

- Explicit vs. Implicit ratings
  - Explicit ratings
    - Users rate themselves for an item
    - Most accurate descriptions of a user's preference
    - Challenging in collecting data
  - Implicit ratings
    - Observations of user behavior
    - Can be collected with little or no cost to user
    - Ratings inference may be imprecise.

Practical issues: Rating

- Scalar ratings
  - Numerical scales.
  - 1-5, 1-7, etc.
- Binary ratings
  - Agree/Disagree, Good/Bad, etc.
- Unary ratings
  - Good, Purchase, etc.
  - Absense of rating indicates no information.

Practical issues: Cold start

- New user
  - Rate some initial items
  - Non-personalized recommendations
  - Describe tastes
  - Demographic info.
- New item
  - Non-CF : content analysis, metadata
  - Randomly selecting items
- New community
  - Provide rating incentives to subset of community
  - Initially generate non-CF recommendation
  - Start with other set of ratings from another source outside community

Evaluation metrics

- Accuracy
  - Predict accuracy
    - The ability of a CF system to predict a user's rating for an item
    - Mean absolute error (MAE)
  - Rank accuracy
    - Precision – percentage of items in a recommendation list that the user would rate as useful
    - Half-life utility – percentage of the maximum utility achieved by the ranked list in question

Evaluation metrics

- Novelty
  - The ability of a CF system to recommend items that the user was not already aware of.
- Serendipity
  - Users are given recommendations for items that they would not have seen given their existing channels of discovery.
- Coverage
  - The percentage of the items known to the CF system for which the CF system can generate predictions.

Evaluation metrics

- Learning rate
  - How quickly the CF system becomes an effective predictor of taste as data begins to arrive.
- Confidence
  - Ability to evaluate the likely quality of its predictions.
- User satisfaction
  - By surveying the users or measuring retention and use statistics

Concepts

- CBR systems recommends an item to user.
- Maintains a profile of user's interests.
- Being used in variety of domains (web-pages, new articles, restaurants, TV programs)
- Main utility of CBR systems is in
  - Selecting a subset of items to be displayed
  - Determining an order to display the items

Concepts: Item representation

- Items are stored in db table.
- Each item has a unique identifier key.
- Data can be structured or unstructured.
  - Structured – a list of restaurants.
  - Unstructured – news articles
  - Unstructured data is more complex
  - Data can also be semi-structured
  - Unrestricted text can be converted to structured representation.

User profiles

- Most recommendation systems use profile of user's interests.
- Generally there are two types of information.
  - A model of user's preferences i.e., a description of the types of items that interest the user.
  - A history of the user's interactions with the recommendation system, it can also include queries typed by the user.
    - History in CBR also serves as training data for a machine learning algorithm that creates a user model.

User profiles

- User model can also be made by user customization.
  - System provides an interface to construct a representation of user's interests.
  - Often check boxes and forms are used.
  - There are many limitations of user customization, it can't capture changing preferences.
    - They do not provide a way to order the items.
    - They do not provide detailed personalized recommendation.

User profiles

- Learning a user model
  - Creating a model of the user's preference from the user history.
  - The training data is divided into categories.
    - The binary categories 'items the user likes'.
    - 'Items the user doesn't like'.
  - This is accomplished either by explicit feedback or by observing the users interactions with items.
  - Importance of classification algorithms is in providing an estimate of the probability that a user will like an unseen item.

Algorithm

There are several algorithm that can be used to train user model:

- Decision trees and rule induction
- Nearest Neighbor methods
- Relevance feedback and Rocchio's algorithm
- Linear classifiers
- Probabilistic methods and Naive Bayes

Algorithm

Decision tree and rule induction

- Ex. are ID3 (Quinlan, 1986) , RIPPER (Cohen, 1995)
- Decision tree learners like ID3, build tree by recursively partitioning training data.
- They are excellent with structured data.
- They are not ideal for unstructured text classification tasks (Pazzani and Billsus, 1997)
- Rule induction algorithm (RIPPER) are closely related to decision trees.
- RIPPER performs competitively with other state-of-the-art text classification algorithm.
- Ripper supports multi-valued attributes.
- This makes it a good for text classification tasks.

Algorithm

Nearest neighbor methods

- Nearest neighbor algorithm stores all of its training data, in memory.
- A new unlabeled item is compared to all stored items using a similarity function and determines the ‘nearest neighbor’ or the k nearest neighbors.
- Numeric score or class label is derived for the unseen item.
- Similarity function depends on the type of data.
- For structured data, a Euclidean distance metric is used.
- For vector space model, the cosine similarity is used.

Algorithm

Relevance feedback and Rocchio's algorithm

$$Q_{i+1} = \alpha Q_i + \beta \sum_{rel} \frac{D_i}{|D_i|} - \gamma \sum_{monrel} \frac{D_i}{|D_i|} \tag{4}$$

- Principle is to allow users to rate documents returned by the retrieval system.
- There are explicit and implicit means of collecting relevance feedback data.
- Rocchio's algorithm is widely used algorithm that operates in the vector space model.
- Based on the modification of initial query through differently weighted prototypes of relevant and non-relevant documents.

Algorithm

- **Linear classifiers**
  - These algorithms learn linear decision boundaries.
  - Hyper planes separating instances in a multi-dimensional space.
  - Suitable for text classification tasks (Lewis et.al. 1996)
  - Outcome of learning process is an n-dimensional weight vector.
  - Threshold is used to convert continuous predictions to discrete class labels.

Algorithm

Probabilistic methods and Naïve Bayes

- Naïve Bayes is an exceptionally well performing text classification algorithm.
- Two frequently used formulations of naïve Bayes
  - Multi-variate Bernoulli
  - Multinomial model
  - Both model assumes that text doc. Are generated by a parameterized mixture model

$$P(d_i|\theta) = \sum_{j=1}^{|C|} P(c_j|\theta)P(d_i|c_j;\theta) \tag{5}$$

Algorithm

- **Probabilistic methods and Naïve Bayes**
  - Multivariate Bernoulli formulation was derived with structured data in mind.
  - Multinomial formulation captures word frequency information.
  - Multinomial naïve Bayes formulation outperforms multivariate Bernoulli model.
    - This is noticeable particularly for large vocabularies (McCallum and Nigam, 1998)

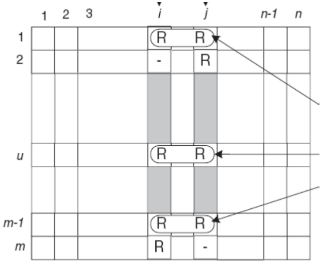
Limitations and extension of CBR system

- CBR can't give good recommendation if the content dos not contain enough information.
- Mainly to distinguish items the user likes from the items the user doesn't like.
- A better approach is to use collaborative and content-based features.
- The paper presented variety of learning algorithms for adapting user profiles, however the choice of algorithm depends upon the representation of content.

Hybrid recommender systems

- Mix of recommender systems
- Recommender system classification – knowledge source
  - Collaborative (CF)
    - User's ratings "only"
  - Content-based recommender (CBR)
    - Product features, user's ratings
    - Classifications of user's likes/dislikes
  - Demographic
    - User's ratings, user's demographics
  - Knowledge-based (KB)
    - Domain knowledge, product features, user's need/query
    - Inferences about a use's needs and preferences

Collaborative filtering (CF) vs. content-based recommenders (CBR)



- User-based CF: Searches for similar users in user-item "rating"matrix
- Item-based CF: searches for similar items in user-item "rating"matrix
- CN: Searches for similar items in item-feature matrix

Recommender system problems

- Cold-start problem
  - Learning based techniques
  - Collaborative, content-based, demographic
    $\Rightarrow$ **Hybrid techniques**
- Stability vs. plasticity problem
  - Difficulty to change established user's profile
    $\Rightarrow$ **Temporal discount - older rating with less influence**

$\Rightarrow$ KB – fewer cold start problem (no need of historical data)
$\Rightarrow$ CF/Demographic – cross-genre niches, jump outside of the familiar (novelty, serendipity)

Strategies for hybrid recommendation

- Combination of multiple recommendation techniques together for producing output
- Different techniques of different types
  - Most common implementations
  - Most promise to resolve cold-start problem
- Different techniques of the same type
  - Ex) NewsDude – naïve Bayes + kNN

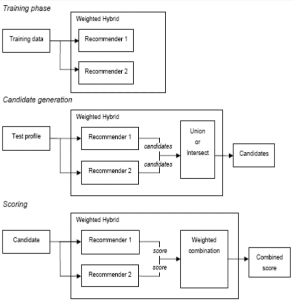
Seven type of recommender systems

- Weighted
- Switching
- Mixed
- Feature combination
- Feature augmentation
- Cascade
- Meta-level

Weighted hybrid

- Concept
  - Each component of the hybrid scores a given item and the scores are combined using a linear formula
  - When recommenders have consistent relative accuracy across the product space
  - Uniform performance among recommenders (otherwise $\Rightarrow$ other hybrids)

Weighted hybrid (Cont)

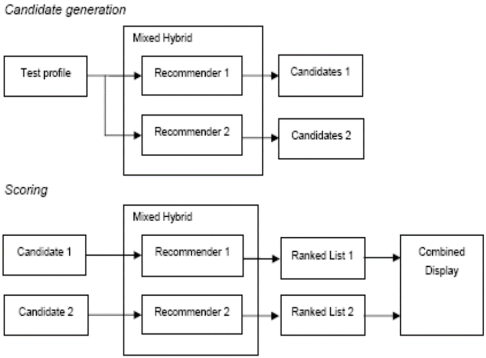


- Training
- Joint rating
  - Intersection - candidates shared between the candidates
  - Union - case with no possible rating, then neutral score
- Linear combination

Mixed hybrid

- Concept
  - Presentation of different components side-by-side in a combined list
  - If lists are to be combined, how are rankings to be integrated?
    - Merging based on predicted rating or on recommender confidence
  - Not fit with retrospective data
    - Cannot use actual ratings to test if right items ranked highly
- Example
  - CF_rank(3) + CN_rank(2) $\Rightarrow$ Mixed_rank(5)

Mixed hybrid (cont)

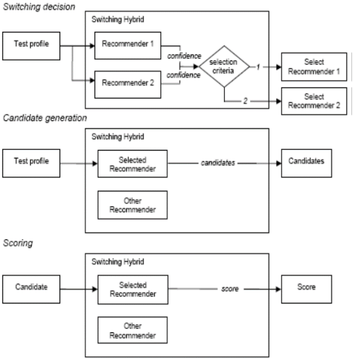


1. Candidate generation
2. Multiple ranked lists
3. Combined display

Switching hybrid

- Concept
  - Selects a single recommender among components based on recommendation situation
  - Different **profile** $\Rightarrow$ different **recommendation**
  - Components with different **performance** for some types of **users**
  - Existence of criterion for switching decision. Ex. Confidence value, external criteria

Switching hybrid (cont)



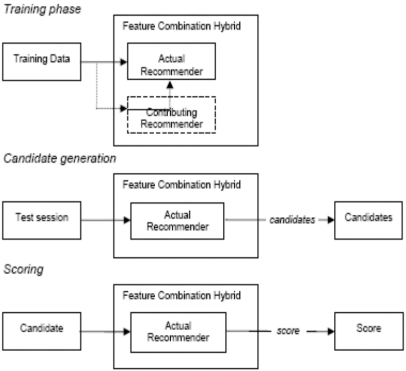
1. Switching decision
2. Candidate generation
3. Scoring

⇒ **No role for unchosen recommender**

Feature combination hybrid

- Concept
  - Inject features of one source into a different source for processing different data
  - Features of “contributing recommender” are used as a part of the “actual recommender”
  - **Adding new features** into the mix
  - Not combining components, just combining knowledge source

Feature combination hybrid (cont)

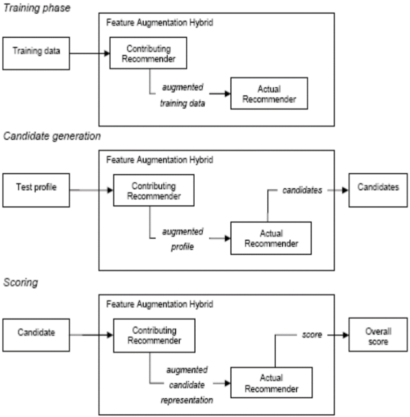


1. Feature combination
   - In training stage
2. Candidate generation
3. Scoring

Feature augmentation hybrid

- Concepts
  - Similar to Feature Combination
  - Generates new features for each item by contributing domain
  - Augmentation/combination – done offline
- Comparison with Feature Combination
  - Not raw features (FC), but the result of computation from contribution (FA)
  - More flexible to apply
  - Adds smaller dimension

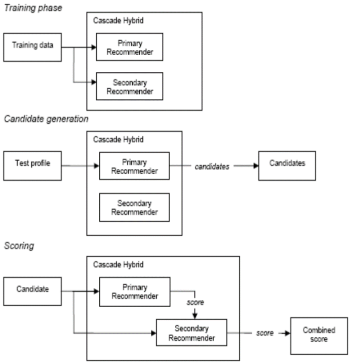
Feature augmentation hybrid (cont)



Cascade hybrid

- Concepts
  - **Tie breaker**
  - Secondary recommender
    - Just tie breaker
    - Do refinements
  - Primary recommender
    - Integer-valued scores – higher probability for ties
    - Real-valued scores – low probability for ties
    - Precision reduction

Cascade hybrid (cont)

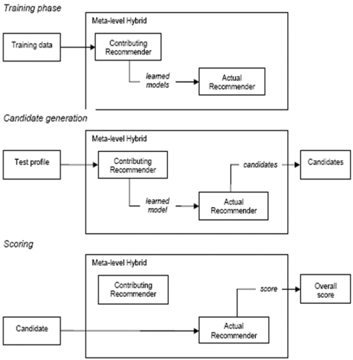


- Procedure
  1. Primary recommender
  2. Ranks
  3. Break ties by secondary recommender

Meta-level hybrid

- Concepts
  - A model learned by contributing recommender $\Rightarrow$ input for actual recommender
  - Contributing recommender completely **replaces** the original knowledge source with a learned model
  - Not all recommenders can produce the intermediary model

Meta-level hybrid (cont)



- Procedure
  1. Contributing recommender
  2. Knowledge Source Replacement
  3. Actual Recommender