Recommender Systems

Nivio Ziviani

Departamento de Ciência da Computação da UFMG

Junho de 2012

Introduction

Chapter 1 of Recommender Systems Handbook
Ricci, Rokach, Shapira and Kantor (editors), 2011.

Introduction

- *Recommender Systems* are software tools and techniques providing suggestions for items to be of use to a user.
- The sugestions relate to various decision-making processes, such as what items to buy, what music to listen, or what news online to read.
- Item: general term used to denote what the system recommend to users.
- In their simplest form, personalized recommendations are offered as ranked lists of items. In performing this ranking, RSs try to predict what the most suitable products or services are, based on the users preferences and constraints.
- RSs collect from users their preferences, which are either explicitly expressed, e.g., as ratings for products, or are inferred by interpreting user actions.

Introduction

- RSs development initiated from a rather simple observation: individuals often rely on recommendations provided by others in making routine, daily decisions.
- It is common to rely on what ones peers recommend when selecting a book to read.
- Employers count on recommendation letters in their recruiting decisions.
- When selecting a movie to watch, individuals tend to read and rely on the movie reviews that a film critic has written and which appear in the newspaper they read.

Introduction

- In seeking to mimic this behavior, the first RSs applied algorithms to leverage recommendations produced by a community of users to deliver recommendations a user looking for suggestions. The recommendations were for items that users with similar tastes had liked.
- This approach is termed *collaborative-filtering* and its rationale is that if the active user agreed in the past with some users, then the other recommendations coming from these similar users should be relevant as well and of interest to the active user.

Introduction

- The study of RSs is relatively new compared to research into other classical information system tools and techniques (e.g., search engines).
- RSs emerged as an independent research area in the mid-1990s, but in the recent years the interest in RSs has dramatically increased:
  - RSs play an important role in such higly rated internet sites as Amazon.com, YouTube, Netflix, Yahoo, TripAdvisor, Last.fm, IMDb.
  - There are dedicated conferences and workshops, where ACM Recommender Systems (RecSys) stablished in 2007 is now the premier annual event in recommender technology research and applications. Others: ACM SIGIR and ACM SIGMOD.

Recommender Systems Function

Various reasons as to why one may want to exploit RS technology:

- *Increase the number of items sold*: This is probably the most important function for a commercial RSs.
  - To be able to sell an additional set of items compared to those usually sold without any kind of recommendation.
  - From the service provider's point of view, the primary goal is to increase the conversion rate.
- *Sell more diverse items*: Select items that might be hard to find without a precise recommendation. For instance, in a movie RS such as Netflix, suggests or advertises unpopular movies to the right users.

Recommender Systems Function

- *Increase the user satisfaction*: The combination of accurate recommendations and a usable interface will increase the users subjective evaluation of the system. This in turn will increase system usage and the likelihood that the recommendations will be accepted.
- *Increase user fidelity*: A system recognizes the old customer and treats him as a valuable visitor, leveraging the information acquired from the user in previous interactions.
- *Better understand what the user wants*: Description of the users preferences, either collected explicitly or predicted by the system is important. For instance, advertise a particular type of promotional message derived by analyzing the data collected from transactions of the users.

Eleven Popular Tasks

Popular tasks by Herlocker, Konstan and Riedl, "Explaining Collaborative Filtering Recommendations", 2000

- *Find some good items*: Recommend some items as a ranked list along with predictions of how much the user would like them (e.g., on a one to five star scale).
- *Find all good items*: Recommend all the items that can satisfy some user needs. This is important when the number of items is relatively small or when the RS is mission-critical, such as in medical or financial applications.
- *Annotation in context*: Given an existing context, e.g., a list of items, emphasize some of them depending on the users long-term preferences. For example, a TV RS might annotate which TV shows displayed in the electronic program guide are worth watching.

Recommender Systems Function

- *Recommend a sequence*: Instead of focusing on the generation of a single recommendation, recommend a sequence of items that is pleasing as a whole. Typical examples include recommending a TV series; a book on RSs after having recommended a book on data mining; or a compilation of musical tracks.
- *Recommend a bundle*: Suggest a group of items that fits well together. For instance, a travel plan may be composed of various attractions, destinations, and accommodation services that are located in a delimited area.
- *Just browsing*: The user browses the catalog without any imminent intention of purchasing an item. The task of the recommender is to help the user to browse the items that are more likely to fall within the scope of the users interests.

Recommender Systems Function

- *Find credible recommender*: Some users do not trust recommender systems thus they play with them to see how good they are in making recommendations. Hence, some system may also offer specific functions to let the users test its behavior in addition to those just required for obtaining recommendations.
- *Improve the profile*: Capability of the user to provide information to the recommender system about what he likes and dislikes. This is a fundamental task that is strictly necessary to provide personalized recommendations.
- *Express itself*: Some users may not care about the recommendations at all. Rather, what it is important to them is that they be allowed to contribute with their ratings and express their opinions and beliefs.

Recommender Systems Function

- *Help others*: Some users are happy to contribute with information, e.g., their evaluation of items (ratings), because they believe that the community benefits from their contribution.
- *Influence others*: There are users whose main goal is to explicitly influence other users into purchasing particular products. There are also some malicious users that may use the system just to promote or penalize certain items.

Data and Knowledge Sources

- In RSs data is primarily about the *items* to suggest and the *users* who will receive these recommendations.
- In general, there are recommendation techniques that are knowledge poor, i.e., they use very simple and basic data, such as user ratings/evaluations for items.
- Other techniques are much more knowledge dependent, e.g., using ontological descriptions of the users or the items, or constraints, or social relations and activities of the users.
- As a general classification, data used by RSs refers to three kinds of objects: items, users, and transactions, i.e., relations between users and items.

Data and Knowledge Source: Items

- Items are the objects that are recommended.
- Items may be characterized by their complexity and their value or utility.
- The value of an item may be positive if the item is useful for the user, or negative if the item is not appropriate and the user made a wrong decision when selecting it.
- We note that when a user is acquiring an item she will always incur in a cost, which includes the cognitive cost of searching for the item and the real monetary cost eventually paid for the item.

Data and Knowledge Source: Items

- Items with low complexity and value are: news, Web pages, books, CDs, movies.
- Items with larger complexity and value are: digital cameras, mobile phones, PCs, etc.
- The most complex items that have been considered are insurance policies, financial investments, travels, jobs.

Data and Knowledge Source: Items

- RSs can use a range of properties and features of the items.
- For example in a movie recommender system, the genre (such as comedy, thriller, etc.), as well as the director, and actors can be used to describe a movie and to learn how the utility of an item depends on its features.
- Items can be represented using various information and representation approaches, e.g., as a single id code, or in a richer form, as a set of attributes.

Data and Knowledge Source: Users

- Users of a RS may have very diverse goals and characteristics.
- In order to personalize the recommendations and the human-computer interaction, RSs exploit a range of information about the users.
- This information can be structured in various ways and again the selection of what information to model depends on the recommendation technique.
- For instance, in collaborative filtering, users are modeled as a simple list containing the ratings provided by the user for some items.
- In a demographic RS, attributes such as age, gender, profession, and education, are used.

Data and Knowledge Source: Users

- User data is said to constitute the user model and encodes her preferences and needs.
- Various user modeling approaches have been used and a RS generates recommendations by building and exploiting user models.
- Since no personalization is possible without a convenient user model, then the user model will always play a central role.
- For instance, considering again a collaborative filtering approach, the user is either profiled directly by its ratings to items or, using these ratings, the system derives a vector of factor values, where users differ in how each factor weights in their model.

Data and Knowledge Source: Transactions

- A transaction is a recorded interaction between a user and the RS.
- Transactions are log-like data that store important information generated during the human-computer interaction and which are useful for the recommendation generation algorithm that the system is using.
- For instance, a transaction log may contain a reference to the item selected by the user and a description of the context for that particular recommendation.
- If available, that transaction may also include an explicit feedback the user has provided, such as the rating for the selected item.
- In fact, ratings are the most popular form of transaction data that a RS collects.

Data and Knowledge Source: Transactions

Ratings can take on a variety of forms:

- Numerical ratings such as the 1-5 stars provided in the book recommender associated with Amazon.com.
- Ordinal ratings, such as strongly agree, agree, neutral, disagree, strongly disagree, the user is asked to select the term that best indicates her opinion regarding an item.
- Binary ratings that model choices in which the user is simply asked to decide if a certain item is good or bad.
- Unary ratings can indicate that a user has observed or purchased an item, or otherwise rated the item positively.
- Another form consists of tags associated by the user with the items the system presents. For instance, in Movielens tags represent how MovieLens users feel about a movie, e.g.: too long, or acting.

Data and Knowledge Source: Transactions

Ratings can take on a variety of forms:

- In transactions collecting implicit ratings, the system aims to infer the users opinion based on the users actions.
- For example, if a user enters the keyword Yoga at Amazon.com she will be provided with a long list of books.
- In return, the user may click on a certain book on the list in order to receive additional information. At this point, the system may infer that the user is somewhat interested in that book.

Recommendation Techniques

- In order to implement its core function, identifying the useful items for the user, a RS must predict that an item is worth recommending.
- In order to do this, the system must be able to predict the utility of some of them, or at least compare the utility of some items, and then decide what items to recommend based on this comparison.
- The prediction step may not be explicit in the recommendation algorithm but we can still apply this unifying model to describe the general role of a RS.
- Here our goal is to provide the reader with a unifying perspective rather than an account of all the different recommendation approaches that will be illustrated in this handbook.

Recommendation Techniques

- To illustrate the prediction step of a RS, consider, for instance, a simple, nonpersonalized, recommendation algorithm that recommends just the most popular songs.
- The rationale for using this approach is that in absence of more precise information about the users preferences, a popular song, i.e., something that is liked (high utility) by many users, will also be probably liked by a generic user, at least more than another randomly selected song.
- Hence the utility of these popular songs is predicted to be reasonably high for this generic user.

Recommendation Techniques

Taxonomy by Burke, “Hybrid Web Recommender Systems”, In: The Adaptive Web, 2007

- *Content-based*:
  - The system learns to recommend items that are similar to the ones that the user liked in the past.
  - The similarity of items is calculated based on the features associated with the compared items.
  - For example, if a user has positively rated a movie that belongs to the comedy genre, then the system can learn to recommend other movies from this genre.
  - For example, Pandora uses the properties of a song or artist in order to seed a ”station” that plays music with similar properties. User feedback is used to refine the station’s results, deemphasizing certain attributes when a user ”dislikes” a particular song and emphasizing other attributes when a user ”loves” a song.

Recommendation Techniques

- *Collaborative filtering*:
  - The simplest and original implementation of this approach recommends to the active user the items that other users with similar tastes liked in the past.
  - The similarity in taste of two users is calculated based on the similarity in the rating history of the users.
  - This is the reason why some authors refer to collaborative filtering as people-to-people correlation.
  - For example, Last.fm creates a ”station” of recommended songs by observing what bands and individual tracks that the user has listened to on a regular basis and comparing those against the listening behavior of other users. Last.fm will play tracks that do not appear in the user’s library, but are often played by other users with similar interests.

Recommendation Techniques

- *Demographic*:
  - This type of system recommends items based on the demographic profile of the user. The assumption is that different recommendations should be generated for different demographic niches.
  - For example, users are dispatched to particular Web sites based on their language or country. Or suggestions may be customized according to the age of the user.
  - While these approaches have been quite popular in the marketing literature, there has been relatively little proper RS research into demographic systems.

Recommendation Techniques

- *Knowledge-based*:
  - Knowledge-based systems recommend items based on specific domain knowledge about how certain item features meet users needs and preferences and, ultimately, how the item is useful for the user.
  - Notable knowledge-based RSs are case-based: a similarity function estimates how much the user needs (problem description) match the recommendations (solutions of the problem).
  - Here the similarity score can be directly interpreted as the utility of the recommendation for the user.
  - Knowledge-based systems tend to work better than others at the beginning of their deployment but if they are not equipped with learning components they may be surpassed by other shallow methods that can exploit the logs of the human/computer interaction (as in CF).

Recommendation Techniques

- *Community-based*:
  - Recommends items based on the preferences of the users friends.
  - This technique follows the epigram “Tell me who your friends are, and I will tell you who you are”.
  - Evidence suggests that people tend to rely more on recommendations from their friends than on recommendations from similar but anonymous individuals. This observation, combined with the growing popularity of open social networks, is generating a rising interest in social recommender systems.
  - The recommendation is based on ratings that were provided by the users friends.

Recommendation Techniques

- *Hybrid*:
  - These RSs are based on the combination of the above mentioned techniques.
  - A hybrid system combining techniques A and B tries to use the advantages of A to fix the disadvantages of B.
  - For instance, CF methods suffer from new-item problems, i.e., they cannot recommend items that have no ratings.
  - This does not limit content-based approaches since the prediction for new items is based on their description (features) that are typically easily available. Given two (or more) basic RSs techniques, several ways have been proposed for combining them to create a new hybrid system.

Algorithms (Wikipedia)

Hundreds of algorithms have been used in the design of recommender systems. The following subsections highlight a few examples.

- *K-Nearest Neighbor*:
  - One of the most commonly used algorithms in recommender systems is the k-nearest neighborhood (k-NN) approach.
  - The k-NN algorithm is a method for classifying objects based on the properties of its closest neighbors in the feature space.
  - In k-NN, an object is classified through a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small).
  - If k = 1, then the object is simply assigned to the class of its nearest neighbor.

Algorithms (Wikipedia)

- *Pearson Correlation*:
  - The Pearson Correlation is a measure of the correlation (linear dependence) between two variables X and Y, giving a value between +1 and -1 inclusive.
  - In a social network, a particular user's neighborhood with similar taste or interest can be found by calculating the Pearson correlation coefficient.
  - By collecting the preference data of top-N nearest neighbors of a particular user (weighted by similarity), the user's preference can be predicted.

Algorithms (Wikipedia)

- *Rocchio Relevance Filtering*:
  - Method of relevance feedback dating back to the 1970s.
  - Rocchio makes use of the VSM and is based on the assumption that most users have a general conception of which items should be denoted as relevant or non-relevant.
  - User feedback is used to refine a search query by emphasizing or deemphasizing certain terms (similar to how Pandora refines its user recommendations).
  - Through feedback, the user's search query is revised to include an arbitrary percentage of relevant and non-relevant terms as a means of increasing the search engine's recall, and possibly the precision as well.
  - The number of relevant and non-relevant terms allowed to enter a query is dictated by a series of weights in the central equation.

Cold Start Problem

- It concerns the issue that the system cannot draw any inferences for users or items about which it has not yet gathered sufficient information.
- In the CF approach, the RS would identify users who share the same rating patterns with the active user, and propose items which the like-minded users favoured (and the active user has not yet seen). Due to the cold start problem, this approach would fail to consider items which no-one in the community has rated previously.
- The cold start problem is often reduced by adopting a hybrid approach between content-based matching and collaborative filtering. New items would be assigned a rating automatically, based on the ratings assigned by the community to other similar items (according to the items' content-based characteristics).

Sparsity Problem

- The number of items sold on major e-commerce sites is extremely large.
- The most active users will only have rated a small subset of the overall database.
- Thus, even the most popular items have very few ratings.
- Bessa, A., Laender, A.H.F., Veloso, A. and Ziviani, N. “Alleviating the Sparsity Problem in Recommender Systems by Exploring Underlying User Communities”. *6th Alberto Mendelzon International Workshop on Foundations of Data Management*, Ouro Preto, Brazil, June 28-30, 2012.

Metrics

- *Precision*
- *Recall*
- $F1 = 2 \times \frac{precision \times recall}{precision + recall}$

Metrics

- *Accuracy*: Root-mean-square error (RMSE) is a frequently used measure of the differences between values predicted by a model or an estimator and the values actually observed. RMSD is a good measure of accuracy.
  $RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - r_i)}{n}}$
  where p_i is the ith-prediction and r_i is the actual value of the ith-prediction.
- *Diversity*
- *Novelty*
- Ribeiro, M.T., Lacerda, A., Veloso, A. and Ziviani, N. "Pareto-Efficient Hybridization for Multi-Objective Recommender Systems", *6th ACM Conference on Recommender Systems*, Dublin, Ireland, 9-13th Sept, 2012.

Amazon.com Recommendations

G. Linden, B. Smith and J. York, Item-to-Item Collaborative Filtering, IEEE Internet Computing, 2003

- Use recommendation algorithms to personalize the online store for each customer.
- It is based on customer interests, showing programming titles to a software engineer and baby toys to a new mother.

E-Commerce Environment

E-commerce recommendation algorithms often operate in a challenging environment. For example:

- A large retailer might have huge amounts of data, tens of millions of customers and millions of distinct catalog items.
- Many applications require the results set to be returned in realtime, in no more than half a second, while still producing high-quality recommendations.
- New customers typically have extremely limited information, based on only a few purchases or product ratings.
- Older customers can have a glut of information, based on thousands of purchases and ratings.
- Customer data is volatile: Each interaction provides valuable customer data, and the algorithm must respond immediately to new information.

Amazon Algorithm: Item-to-Item Collaborative Filtering

- Online computation scales independently of the number of customers and number of items in the product catalog.
- Produces recommendations in realtime.
- Scales to massive data sets.
- Generates high quality recommendations.

Common Approaches to Solving the Recommendation Problem

Approach 1: Find similar customers

- Find a set of customers whose purchased and rated items overlap the users purchased and rated items;
- Aggregates items from these similar customers;
- Eliminates items the user has already purchased or rated;
- Recommends the remaining items to the user.

Examples: *collaborative filtering* and *cluster models*

Common Approaches to Solving the Recommendation Problem

Approach 2: Find similar items

- For each of the users purchased and rated items, find similar items.
- Aggregates the similar items and recommends them.

Examples: *search-based methods* and Amazon's *item-to-item collaborative filtering*.

Traditional Collaborative Filtering

- Represents a customer as an N-dimensional vector of items, where N is the number of distinct catalog items.
- The components of the vector are positive for purchased or positively rated items and negative for negatively rated items.
- To compensate for best-selling items, the algorithm typically multiplies the vector components by the inverse of the number of customers who have purchased or rated the item, making less well-known items much more relevant.
- For almost all customers, this vector is extremely sparse.

Traditional Collaborative Filtering

- It generates recommendations based on a few customers who are most similar to the user.
- It can measure the similarity of two customers, A and B, in various ways; a common method is to measure the *cosine* of the angle between the two vectors.
- The algorithm can select recommendations from the similar customers items using various methods.
- A common technique is to rank each item according to how many similar customers purchased it.

Traditional Collaborative Filtering

- It is $O(MN)$ in the worst case, where M is the number of customers and N is the number of items (it examines M customers and up to N items for each customer).
- Because the average customer vector is sparse, it tends to be closer to $O(M + N)$.
- Scanning every customer is approximately $O(M)$, not $O(MN)$, because almost all customer vectors contain a small number of items.
- But there are a few customers who have purchased or rated a significant percentage of the catalog, requiring $O(N)$ processing time.
- For large data sets such as 10 million or more customers and 1 million or more catalog items the algorithm encounters severe performance and scaling issues.

Traditional Collaborative Filtering

- It is possible to partially address these scaling issues by reducing the data size.
- We can reduce M by randomly sampling the customers or discarding customers with few purchases, and reduce N by discarding very popular or unpopular items.
- It is also possible to reduce the number of items examined by a small, constant factor by partitioning the item space based on product category or subject classification.
- Dimensionality reduction techniques such as clustering and principal component analysis can reduce M or N by a large factor.

Traditional Collaborative Filtering

- Unfortunately, all these methods also reduce recommendation quality in several ways.
- First, if the algorithm examines a small customer sample, selected customers will be less similar to the user.
- Second, item-space partitioning restricts recommendations to a specific product or subject area.
- Third, if the algorithm discards most popular or unpopular items, they will never appear as recommendations.
- Dimensionality reduction applied to the item space tend to have the same effect by eliminating low-frequency items.
- Dimensionality reduction applied to the customer space effectively groups similar customers into clusters; as we now describe, such clustering can also degrade recommendation quality.

Cluster Models

- To find customers who are similar to the user, cluster models divide the customer base into many segments and treat the task as a classification problem.
- The algorithms goal is to assign the user to the segment containing the most similar customers.
- It then uses the purchases and ratings of the customers in the segment to generate recommendations.
- The segments typically are created using a clustering or other unsupervised learning algorithm, although some applications use manually determined segments.

Cluster Models

- Using a similarity metric, a clustering algorithm groups the most similar customers together to form clusters or segments.
- Because optimal clustering over large data sets is impractical, most applications use various forms of greedy cluster generation.
- These algorithms typically start with an initial set of segments, which often contain one randomly selected customer each.
- They then repeatedly match customers to the existing segments, usually with some provision for creating new or merging existing segments.
- For very large data sets especially those with high dimensionality sampling or dimensionality reduction is also necessary.

Cluster Models

- Once the algorithm generates the segments, it computes the users similarity to vectors that summarize each segment, then chooses the segment with the strongest similarity and classifies the user accordingly.
- Some algorithms classify users into multiple segments and describe the strength of each relationship.
- Cluster models have better online scalability and performance than collaborative filtering because they compare the user to a controlled number of segments rather than the entire customer base.
- The complex and expensive clustering computation is run offline.

Cluster Models

- However, recommendation quality is low.
- Cluster models group numerous customers together in a segment, match a user to a segment, and then consider all customers in the segment similar customers for the purpose of making recommendations.
- Because the similar customers that the cluster models find are not the most similar customers, the recommendations they produce are less relevant.
- It is possible to improve quality by using numerous finegrained segments, but then online usersegment classification becomes almost as expensive as finding similar customers using collaborative filtering.

Search-Based Methods

- Search- or content-based methods treat the recommendations problem as a search for related items.
- Given the users purchased and rated items, the algorithm constructs a search query to find other popular items by the same author, artist, or director, or with similar keywords or subjects.
- If a customer buys the Godfather DVD Collection, for example, the system might recommend other crime drama titles, other titles starring Marlon Brando, or other movies directed by Francis Ford Coppola.

Search-Based Methods

- If the user has few purchases or ratings, searchbased recommendation algorithms scale and perform well.
- For users with thousands of purchases, however, its impractical to base a query on all the items.
- The algorithm must use a subset or summary of the data, reducing quality.
- In all cases, recommendation quality is relatively poor. The recommendations are often either too general (such as best-selling drama DVD titles) or too narrow (such as all books by the same author).
- Recommendations should help a customer find and discover new, relevant, and interesting items. Popular items by the same author or in the same subject category fail to achieve this goal.

Item-to-Item Collaborative Filtering

Amazon.com Algorithm

- Amazon.com uses recommendations as a targeted marketing tool in email campaigns and on its Web sites pages.
- Users can filter their recommendations by product line and subject area, rate the recommended products, rate their previous purchases, and see why items are recommended.
- Offers customers product suggestions based on the items in their shopping cart.
- The feature is similar to the impulse items in a supermarket checkout line, but our impulse items are targeted to each customer.

Item-to-Item Collaborative Filtering

How it Works

- Rather than matching the user to similar customers, item-to-item collaborative filtering matches each of the users purchased and rated items to similar items, then combines those similar items into a recommendation list.
- To determine the most-similar match for a given item, the algorithm builds a similar-items table by finding items that customers tend to purchase together.

Item-to-Item Collaborative Filtering

Similar-Items Table

The algorithm builds a similar-items table by calculating the similarity between a single product and all related products:

```
For each item I1 in product catalog
  For each customer C who purchased I1
    For each item I2 purchased by customer C
      Record that a customer purchased I1 and I2
  For each item I2
    Compute the similarity between I1 and I2
```

Similarity between two items given by the cosine measure:

- Each vector corresponds to an item;
- Vector's M dimensions correspond to customers who have purchased that item.

Item-to-Item Collaborative Filtering

Similar-Items Table

Complexity of the offline computation:

- $O(N^2M)$ in the worst case.
- $O(NM)$ in practice, as most customers have very few purchases.
- Sampling customers who purchase best-selling titles reduces runtime even further, with little reduction in quality.

Item-to-Item Collaborative Filtering

The Algorithm

Given a similar-items table:

- The algorithm finds items similar to each of the users purchases and ratings;
- Aggregates those items;
- Recommends the most popular or correlated items.

Complexity: This computation is very quick, depending only on the number of items the user purchased or rated.

Item-to-Item Collaborative Filtering

Scalability

- The key to scalability and performance is that it creates the expensive similar-items table offline.
- Looking up similar items for the users purchases and ratings scales independently of the catalog size or the total number of customers.
- It is dependent only on how many titles the user has purchased or rated.
- Because the algorithm recommends highly correlated similar items, recommendation quality is excellent.
- Unlike traditional collaborative filtering, the algorithm also performs well with limited user data, producing high-quality recommendations based on as few as two or three items.

Item-to-Item Collaborative Filtering

Comparison to Tradition Collaborative Filtering

- Traditional collaborative filtering does little or no offline computation.
- Its online computation scales with the number of customers and catalog items.
- It is impractical on large data sets, unless it uses dimensionality reduction, sampling, or partitioning all of which reduce recommendation quality.

Item-to-Item Collaborative Filtering

Comparison to Cluster models

- Cluster models can perform much of the computation offline, but recommendation quality is relatively poor.
- To improve it, its possible to increase the number of segments, but this makes the online usersegment classification expensive.

Item-to-Item Collaborative Filtering

Comparison to Search-Based Models

- Search-based models build keyword, category, and author indexes offline, but fail to provide recommendations with interesting, targeted titles.
- They also scale poorly for customers with numerous purchases and ratings.

E-Commerce Environment

- x