IS590DT: Data Mining Applications

WEEK 14: Recommender System and Course Wrap-up

Spring 2019

Yingjun Guan
yingjun2@illinois.edu
http://ischool.illinois.edu/people/yingjun-guan

ILLINOIS | School of Information Sciences
The iSchool at Illinois

Contents today

- **What is recommender system?**
- **Collaborative vs Content Based Filtering**
- **R application of RS**
- **Wrap-up of data mining**
- **Q & A for the previous assignment and finals**

What is recommender system?

A **recommender system** or a **recommendation system** (sometimes replacing "**system**" with a synonym such as platform or engine) is a subclass of information filtering **system** that seeks to predict the "rating" or "preference" a user would give to an item.

Recommender system - Wikipedia
https://en.wikipedia.org/wiki/Recommender_system

What is recommender system?

- RS is everywhere: Amazon, Wayfair, Netflix, Google News, Pinterest, Spotify, Facebook, Linkedin, OkCupid ……
- A system that can automatically recommend items to users, which are likely to be of interest to the users, by utilizing historical information.
- Let's start with the famous Netflix competition...

Netflix competition

- Netflix, the world's largest on-line movie rental company.
  - October 2, 2006, publicly released a set of data for competition.
- Goal: develop a system to recommend movies to users based on personalized prediction of a user's preference of particular movies.
- Data: movie ratings (from 1 to 5).
  - Training: 100 million ratings, 480,000 users, 18,000 movies (user-movie
  - rating matrix is 99% missing).
  - Quiz: 1.5 million ratings but withheld.
  - Test: 1.5 million ratings but withheld.
  - However, the dataset is no longer available online anymore
- Evaluation: root mean squared error (RMSE).
- Overall average: 1.0528 for quiz, 1.0540 for testing.
- Cinematch (Netflix's algorithm): 0.9514 for quiz, 0.9525 for testing.

Netflix competition, cont'd

- Grand prize: RMSE 0.8572 (10%), or better, on the test set.
- Progress Prize: At least 1% better than previous year.
- In 2007 Progress Prize, singular value decomposition (SVD) and Restricted Boltzmann Machine (RBM) (Salakdutdinov, Mnih & Hinton, 02) are top performers.
- RBM is a building block for a deep learner.
- SVD: (0.8914), performs slightly better than RBM
- RBM (0.8990).

Netflix competition

- Netflix competition discontinued but other data competitions emerged.
- KDD Cup: the annual Data Mining and Knowledge Discovery competition organized by ACM Special Interest Group on Knowledge Discovery and Data Mining.
- KDD-2013, Yahoo music data for competition. Winner: A team from Taiwan using an ensemble method.
- On-going data competition: Alibaba–(on-line sales data over months).
- Goal: Seek methods to improve recommender systems.
- More data sets of this kind are available.
  - Movielens data, about 1% (10 Millions) ratings observed.
  - Yahoo music data, 300 million ratings.
  - Millions Song Dataset (MSD): 300GB of music data.

Let's recall how amazon recommend books for you.

- What do they recommend you based on?

Utility Matrix

- A matrix of users' ratings over items.
- Each row represents a user; each column represent an item.
- A utility matrix with 9 ratings looks like:

					
Tom	?	4	5	?	?
Jerry	?	?	?	3	1
Denny	2	5	?	?	?
Sarah	?	?	5	?	?
Edwin	?	?	?	?	4

Description

- Data are extremely high-volume (1M to 1B ratings).
- Extremely sparse.
- Algorithms need to be scalable.
- MovieLens 10M data:
  - 71,567 users over 10,681 movies
  - 10,000,054 observed ratings out of 764,407,127 possible ratings (1%)
- Goal: Complete the utility matrix.
- Note: Temporal or seasonal effects, rating times could be informative.

Concepts behind recommender systems

- Types
  - Collaborative filtering: measure similarity of users by their item preferences, e.g., Grooveshark.
    - Idea: If I liked {A; B}; others liked {A; B; J} –the system recommends J based on similar taste.
  - Content-based filtering: measures similarity by looking for common features of the items, e.g., Pandora.
    - Idea: build my profile based on content of items I like. For instance, if a Netflix user has watched many action movies, then recommend new action movies.
  - Hybrid: uses both types of information.
  - Mobile recommender systems requires real-time processing mobile data.

- Issues:
  - Data are extremely sparse.
  - Collaborative filtering: Cold start problem: fails for new items that have not been rated previously.
  - Content-based filtering: fails for new users who do not have their user profiles yet.
- Challenges:
  - Size: MoiveLens data, 105 users over 104 movies, with potentially 1 billion (109) ratings.
  - Missing: Movielens data, about 1% (10 Millions) ratings observed. Non-ignorable missing
  - Dependency: between items and users.
  - Key: (A) Prediction accuracy & (B) Scalability– MapReduce when data can not be fitted in memory.

Approaches and Objectives

- Major approaches:
  - Deep neural network (e.g., deep learning): train for prediction.
  - Neighborhood methods: content-based approaches applying certain similarity metric.
  - Matrix factorization: matrix completion through singular value decomposition.
- Objectives:
  - Design a method to achieve (A) & (B).
  - Leverage user-specific and content-specific predictors to enhance prediction.
  - Combine the strength of collaborative filtering with content-based filtering.

Evaluation

- Accuracy of the predicted scores:
  - Root mean squared error (RMSE):

$$\mathrm{RMSE} = \left\{ \frac{1}{n_{\mathrm{test}}} \sum_{\text{test data}} (r_{ui} - \hat{r}_{ui})^2 \right\}^{1/2}.$$

  - Mean absolute error:

$$\mathrm{MAE} = \left\{ \frac{1}{n_{\mathrm{test}}} \sum_{\text{test data}} |r_{ui} - \hat{r}_{ui}| \right\}.$$

  - Ranking: Match the true 10 with predicted top 10 preferences.

Collaborative filtering

- A user purchases item A. The system recommends to this user a top choice of other clients who have purchased item A.
- Preference (Rating) matrix: R (1 yes, 0 no):

	Book 1	Book 2	Book 3
User 1	0	0	0
User 2	0	1	1
User 3	0	1	1
User 4	0	0	1

- User 2 purchased Book 2, then examine all books by users who also purchases Book 2: P(Book 3|Book 2 purchased) = 1, P(Book 1|Book 2 purchased) = 0. Recommend Book 3.
- Cold start problem: Book 1 will never be recommended as none had purchased it yet.

Collaborative filtering

- Data collection
  - Explicit: Ask a user to rate an item (s), to create a preferred list of items.
  - Implicit: Observe the items that a user has viewed online, record items that a user has purchased online.
- Construction of users' feature profile:
  - User profile: browsing history (visit pattern); demographic information (age, gender, geo-location).
  - Item features (topics, keywords, category, entities); image tags.

Collaborative filtering

- Key advantage: Can accurately recommend complex items such as movies without requiring "understanding" the item itself.
  - Recommend based on user-item interaction: similar items based on user profiles.
  - Build a decision/classification rule given users' features as covariates.
  - Fail for new items.
- Disadvantages:
  - Cold start problem.
  - Scalability: require computing power.
  - Sparsity: involve a large number of missing values (not missing at random).

Collaborative filtering: real example

- Recommender “Audioscrobbler”: use in Last.fm, a music website.
- Data collection: Build a detailed profile of each user’s musical taste by recording details of the tracks the user listens to.
- From Internet radio stations, or user’s computer or portable music devices.
- This information is transferred to Last.fm’s database either via the music player (R-dio, Spotify, Clementine, Amarok, MusicBee) or a plugin installed into the user’s music player.
- The info is displayed on a user’s profile page and compiled to create reference pages for individual artists.
- Last.fm offers numerous social networking features & recommends artists similar to the user’s favorites.

Content-based filtering: example 2

- Transaction matrix: R (1 yes, 0 no):

	Book 1	Book 2	Book 3
User 1	0	0	0
User 2	0	1	1
User 3	0	1	1
User 4	0	0	1

- If a user purchased Book 2, then recommend Book 3 based on content.
- Because Book 2 and Book 3 are similar: 2/3 users purchased both—a similarity measure.
- Fail for new users. It can not recommend any book to User 1 (new user).

Content-based filtering

- Use an item's description (keywords) and a user's profile of preference
- Various candidate items are compared with items previously rated by a user and the best-matching items are recommended.
  - Model users' preference profile and a history of the user's interaction with the recommender.
  - Candidate items: movie features (genre, release date, cast); news article features (topic, word frequencies); image tags.

Content-based filtering

- Recommend items to a given user:
  - Recommend similar items based on item profiles.
  - Build a decision/classification rule given item features as covariates.
  - Fail for new users.
- Advantage & Disadvantages:
  - Does not require to collect much information about each user, therefore not personalized.

Content-based filtering: real example

- Recommender PandoraRadio: Pandora Internet Radio, uses a music streaming and automated music recommender.
- Pandora uses the properties of a song or artist (a subset of the 400 attributes provided by the Music Genome Project) to seed a “station” with music of similar properties.
- User feedback refines the station’s results, e.g., deemphasizing certain attributes when a user “dislikes” a song, and emphasizing other attributes when a user "likes" a song.
- Contrast:
  - Last.fm requires a large amount of information from a user for accurate recommendations. This is a problem called cold start, common in collaborative filtering systems.
  - Pandora needs very little information to get started, is more limited (e.g., can only recommend songs similar to the original seed).

Neighborhood-based filtering: Example 3

- Transaction matrix: R (1 yes, 0 no):

	Book 1	Book 2	Book 3
User 1	0	0	?
User 2	0	1	1
User 3	0	1	1

- To predict P(Book 3 purchased by User 1)
- Given Users 2 and 3 who purchased Book 3, do we expect that the prob is closer to that of User 2 or User 3?
- Define a similarity measure: s(1; 2) and s(1; 3).
- Similarity: inferred from overlapping items that they have already purchased: for instance, s(1; 2) = 1=3 and s(1; 3) = 0.
- Predict using a weighted average of the scores:

$$\frac{s(1,2)}{s(1,2)+s(1,3)} r_{23} + \frac{s(1,3)}{s(1,2)+s(1,3)} r_{33} = 1.$$

Neighborhood methods

- Key idea:
  - Recommend similar items, or items purchased by similar users.
  - Simple, efficient, and stable.
  - Flavors: User-based or Item-based.
- Challenges:
  - If # users or items is large, requires extensive computation with pairwise comparison.
  - If data are highly sparse, then difficult to recommend.

Talking about similarity...

- Give me some similarity measures.

Hybrid recommender systems

- Combining collaborative and content-based filtering could be more effective.
- How? (1) Make content-based and collaborative-based predictions separately and then combine them; (2) Add content-based capabilities to a collaborative-based approach, and vice versa; (3) Unify approaches into one model.
- Example: Netflix uses a hybrid system: Recommends by comparing and searching habits of similar users (collaborative filtering) and by offering movies sharing characteristics with films that a user has rated highly (content-based filtering).

Hybrid recommender systems

- "Hybrid" refers to combining multiple recommendation techniques to produce its output.
- Four major recommender systems: collaborative filtering, content-based, demographic, and knowledge-based.
- Some useful hybridizations:
  - Weighted: Combine using scores of different recommendation components, e.g., ensemble method.
  - Switching: Chooses a recommender and applies the selected one given a certain condition
  - Mixed: Merging multiple rankings into one.
  - Feature combination: Give a single recommendation using features derived from different knowledge sources.
  - Feature Augmentation: Use a set of features as a part of the input to another recommender.

R application of RS

- Dataset: MovieLens
- https://grouplens.org/datasets/movielens/1m/

Some other analysis of RS.

Review of data mining

Review of assignments

Last words

- Data mining is of bright future.
- Practice makes perfect.