DATA ANALYTICS

Unit 4: Evaluation of Recommender Systems + Case Study + Other Applications

Gowri Srinivasa
Department of Computer Science and Engineering

Evaluation of Recommender Systems

Paradigms

1. User Studies: test subjects are actively recruited, and they are asked to interact with the recommender system to perform specific tasks (likes and dislikes inferred). Feedback can be collected from the user before and after the interaction.
   Results from user evaluations cannot be fully trusted.

2. Online Evaluation: A/B Testing (coming up in Unit 5)

3. Offline Evaluation: With historical datasets

Evaluation of Recommender Systems

Measures

1. Accuracy: MSE, RMSE, etc.

2. Coverage: The fraction of users for which at least *k* ratings may be predicted (user-space coverage); the fraction of items for which the ratings of at least *k* users can be predicted (item-space coverage); the fraction of items that are recommended to at least one user (catalog coverage)

3. Confidence and Trust: confidence measures the system's faith in the recommendation, trust measures the user's faith in the evaluation

4. Novelty: Likelihood of a system to recommend an item the user was not aware of (A *differential* accuracy between future and past predictions can be used to quantify this.)

Evaluation of Recommender Systems

Measures (contd.)

5. Serendipity – ‘lucky discovery’ or surprise factor (Ethiopian restaurant recommended to someone who likes Indian food is serendipitous; all that is novel is not serendipitous!)

6. Diversity: If three movies are recommended, they must not all be of the same genre; the changes a user will select one of them will then be higher (measured using content-centric similarity between pairs of items)

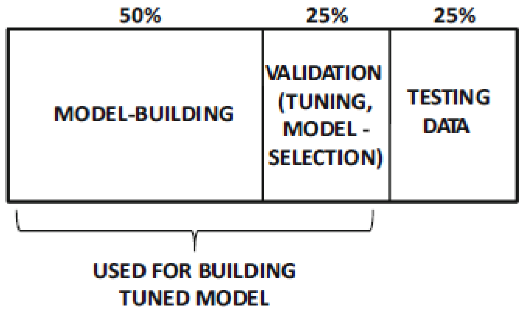
7. Robustness and stability: not significantly affected in the presence of attacks such as fake ratings or when the patterns in the data evolve significantly over time

8. Scalability: perform effectively and efficiently in the presence of large amounts of data (quantified based on training time, prediction time and memory requirements)

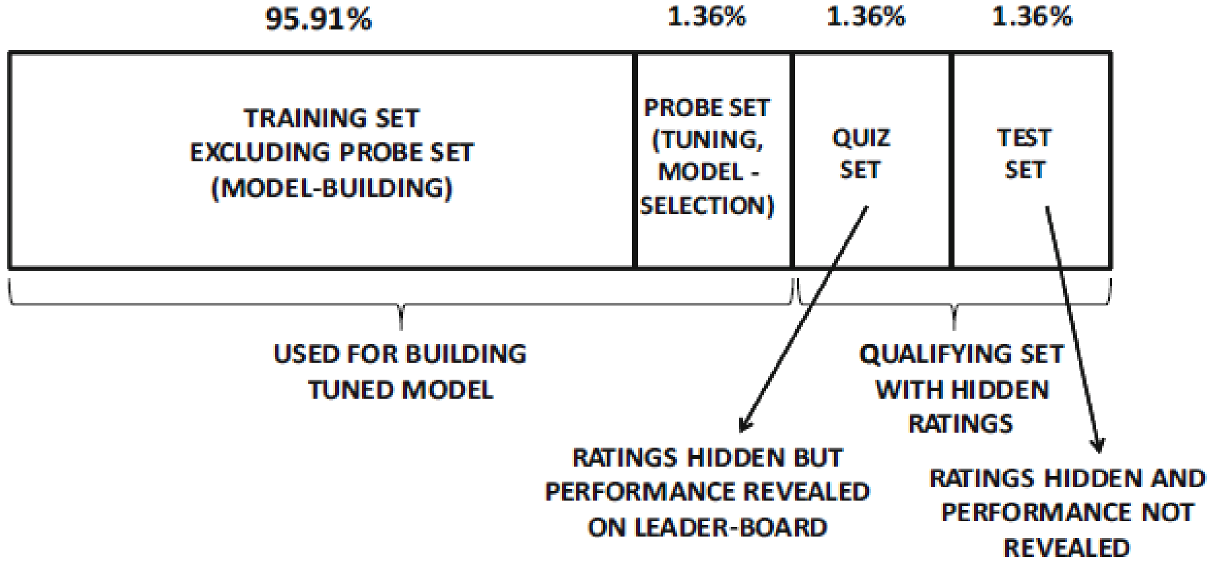
Case Study: The Netflix Prize - 1

The competition began on October 2, 2006: Prize awarded to the best collaborative filtering algorithm to predict user ratings based on training data <user, movie, date of grade, grade>

Usual partitioning of data in problems



Partitioning of Data for the Netflix Prize Challenge to penalize overfitting



Case Study: The Netflix Prize - 2

On September 21, 2009 "BellKor's Pragmatic Chaos" won the $1 million for their algorithm

This team used an ensemble of models; specifically, they used Gradient Boosted Decision Trees to combine 500 models! (Previous solutions had used linear regression for the combination).

Briefly, gradient boosted decision trees work by sequentially fitting a series of decision trees to the data; each tree is asked to predict the error made by the previous trees, and is often trained on slightly perturbed versions of the data.

Since GBDTs have a built-in ability to apply different methods to different slices of the data, we can add in some predictors:

- Number of movies each user rated
- Number of users that rated each movie
- Factor vectors of users and movies
- Hidden units of a restricted Boltzmann Machine

that help the trees make useful clusterings (For example, one thing that Bell and Koren found (when using an earlier ensemble method) was that RBMs are more useful when the movie or the user has a low number of ratings, and that matrix factorization methods are more useful when the movie or user has a high number of ratings.)

For the interested student: read more about this here (a summary) and here (links to the top papers). Crowd sourcing problem solving led to founding Kaggle and other similar organizations!

A few other application domains

1. *Query recommendation:* How can web logs can be used to recommend queries to users? Typically *session-specific* (i.e., dependent on the history of user behavior in a short session) and do not use *long-term* user behavior. This is because queries are often issued in scenarios in which user re-identification mechanisms are not available over multiple sessions.

2. *Portal content and news personalization:* Many online portals have strong user identification mechanisms by which returning users can be identified. In such cases, the content served to the user can be personalized. This approach is also used by news personalization engines, such as Google News, in which Gmail accounts are used for user identification. News personalization is usually based on implicit feedback containing user behavior (clicks), rather than explicit ratings.

3. *Computational advertising:* A form of recommendation, because it is desirable for companies to be able to identify advertisements for users based on a relevant context (Web page or search query). Therefore, many ideas from recommendation systems are directly used in the area of computational advertising.

4. *Reciprocal recommender systems:* In these cases, both the users and items have preferences (and not just the users). For example, in an online dating application, both parties have preferences, and a successful recommendation can be created only by satisfying both parties.

Additional References

R1 Data Mining: Concepts and Techniques by Han, Kamber and Pei (Morgan Kaufman)

Introduction to Data Mining by Tan, Steinbach and Kumar (Pearson – First Edition) Chapters 6 and 7

Recommender Systems – The Textbook by Charu C. Agarwal (Chapter 7)

THANK YOU

Gowri Srinivasa
Professor,
Department of Computer Science
gsrinivasa@pes.edu