5-7: Experimental Protocols for Rating Data

Introduction

- We've discussed several evaluation metrics
- We now turn to experimental protocol design
  - How do we structure an evaluation using these metrics?

Learning Objectives

- Understand basic structure of a crossfold recommender evaluation
- Be able to design a plausible, repeatable evaluation using best practices
  - For rating or yes/no data

Goal of Offline Evaluation

- To *estimate* the recommender's quality
  - High-throughput evaluation
  - Answer important research questions
- Often cannot answer if recommender really works
  - User-based evaluation needed
  - Link to business metrics is weak

Background

- Offline protocols inspired by related research areas
  - Machine learning
  - Information retrieval

Machine Learning

- Hidden data
  - Hold out some data, try to predict/classify it
- Cross-validation
  - Split data into partitions, hold out each in turn
  - Average results
  - Mitigates effects of split in results
- Measure score or classification accuracy

Information Retrieval

- Measure accuracy in providing results for queries with known results
- Uses known preference judgements

Adapting to Recommenders

- Use ratings/purchases/clicks as relevance judgements or ground truth
- Measure recommendations or predictions

Basic Structure

- Partition data set into *k* partitions
- For *i* = 1 to *k*
  - train on all sets other than *i*
  - test on set *i*
- What *k* to use?
  - Large values → more training data
  - Small values → more efficient
  - 5 and 10 are common

Splitting data

- Split ratings
- Split users
  - Allows more control for measuring expected user experience
- Split items
  - Rarely, if ever, done

Splitting users

- Split user ratings randomly
  - Very common
  - Use to compare with existing results
- Split user ratings by time
  - More accurate simulation of user experience
  - Results often worse
- Best, but expensive: only train on ratings before time of test rating

Using log data

- Log data often unary (clicked, purchased), nothing known about absent items
- Basic structure is the same
- More discussion in next lecture

Good Practice

- Split users into *k* partitions (5 is common)
- Split user ratings by time
  - Use random to compare with previous results
- Include user query ratings in train data
- Document your protocol carefully
  - So you can run it again
  - So others can compare

5-7: Experimental Protocols for Rating Data