UFMG UNIVERSIDADE FEDERAL DE MINAS GERAIS

Recommender Systems

Evaluation Methods

Rodrygo Santos
rodrygo@dcc.ufmg.br

Why evaluate?

UFMG COMPUTER SCIENCE

- Gazillions of algorithms
  - Collaborative, content-based, hybrid…
  - *Which one to choose?*
- Evaluation enables an informed choice
  - Rigor of science
  - Efficiency of practice

Why evaluate?

UFMG COMPUTER SCIENCE

- RS as an applied *scientific* discipline
  - Evaluation is a critical component
- RS has become plagued with weak experimentation, causing
  - Outsiders to think of RS as non-scientific
  - Minor improvements vs. weak baselines
  - Difficulty in defining the "state-of-the-art"

Why evaluate?

UFMG COMPUTER SCIENCE

For researchers

- It allows you to convince others (e.g., reviewers, researchers, funders) that your work is meaningful
- Without a strong evaluation, your paper will (probably) be rejected
- Empirical evaluation helps guide meaningful research directions

For practitioners

- It allows you to convince others (e.g., company VPs, investors, clients) that your work is meaningful
- Without a strong evaluation, your code will (probably) not be deployed
- Empirical evaluation helps guide meaningful development directions

What to evaluate?

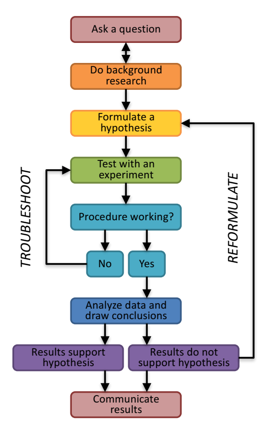
UFMG COMPUTER SCIENCE

- Three fundamental types of RS research
  - Systems (efficiency)
  - Methods (effectiveness)
  - Applications (user utility)
- Evaluation plays a critical role for all three
  - Our primary focus is on "methods" research
  - Same principles can be applied elsewhere (including other disciplines)

How to evaluate?

- *Scientifically, of course!*
  - *We have known how for 3,600+ years!*

Edwin Smith Papyrus, the oldest known surgical treatise



A pragmatic recipe

UFMG COMPUTER SCIENCE

- Four major steps
  - Ask a question
    - After observations
  - Formulate a hypothesis
    - After further observations
  - Perform an experiment
    - Test the hypothesis
  - Analyze the results
    - Accept or refute the hypothesis

Asking questions

UFMG COMPUTER SCIENCE

- What problem are you trying to solve?
  - Or in RS parlance, what ***task***?
- Are you solving a well-known task? *e.g., movie recommendation?*
  - Review the literature!
- Is your task unlike anything done before?
  - Try to characterize it (see class #2)
  - How do you define ***success***?

Formulating hypotheses

UFMG COMPUTER SCIENCE

- A hypothesis must be falsifiable
  - *e.g., "SVD improves CF"*
- It either holds or does not…
  - … with respect to the considered data (scope)
  - … perhaps under certain conditions (extent)
- It concerns some component of a method
  - Can it be tested in ***isolation***?

Research questions

UFMG COMPUTER SCIENCE

- *Hypotheses turned into questions*
  - *e.g., "does SVD improve CF?"*
- Open-ended "hypotheses"
  - *e.g., "how does SVD impact CF?"*

Performing experiments

UFMG COMPUTER SCIENCE

- Key components
  - Experimental setup
  - Analysis of results
- Key concern: ***reproducibility***
  - Must specify each and every detail needed for reproducing our method and the experiment

Experimental methodology

UFMG COMPUTER SCIENCE

- Key components
  - Research questions
  - Evaluation methodology
  - Evaluation benchmarks
  - Reference comparisons
  - Parameter tuning
  - Evaluation metrics

Research questions

UFMG COMPUTER SCIENCE

- We've talked about it before
  - But it's worth stressing
- Methods are ***not devised arbitrarily***
  - *We always have a hypothesis (whether implicit or explicit) for why our work should improve*
  - *Even the best results are useless if nobody understands what you are trying to solve*
- So, ***spell out*** your research questions!

Evaluation methodology

UFMG COMPUTER SCIENCE

Offline evaluation

- Retrospective experiments
  - How well can we predict (hidden) ***past preferences***?
- Benchmarked using static test collections
  - Highly reproducible
  - Poorly realistic

Online evaluation

- Prospective experiments
  - How well can we predict ***future preferences***?
- Benchmarked using live user interactions
  - Poorly reproducible
  - Highly realistic

Offline evaluation

UFMG COMPUTER SCIENCE

- Goal is to *estimate* the recommender's quality
  - High-throughput evaluation
  - Answer important research questions
- Often can't answer if recommender really works
  - User-based evaluation needed
  - Link to business metrics is weak
- Protocols inspired by related research areas
  - Machine learning
  - Information retrieval

Public test collections

UFMG COMPUTER SCIENCE

- For search
  - TREC has collections on Web, blog, tweet, video, question-answering, legal documents, medical records, chemicals, genomics, … search
- Check out
  - http://trec.nist.gov/tracks.html
  - http://trec.nist.gov/data.html

Public test collections

UFMG COMPUTER SCIENCE

- For recommendation
  - Many available test collections for movies, music, books, food, papers, jokes, tags, dates, healthcare
- Check out
  - http://www.recsyswiki.com/wiki/Category:Dataset
  - https://gist.github.com/entaroadun/1653794

You can build your own

UFMG COMPUTER SCIENCE

- Three core components
  - Users, items, user-item associations

For search

- A set of users' queries
- A corpus of documents
- A map of users' relevance assessments

For recommendation

- A set of users' profiles
- A catalog of items
- A map of users' preferences

You can build your own

UFMG COMPUTER SCIENCE

- Document corpus / item catalog
  - Go crawl it!
- Queries / user profiles
  - The more the better (e.g., at least 50)
  - Representative of the population (e.g., from a log)
- Relevance assessments / preferences
  - Lab studies, crowdsourcing
  - Must be unbiased (don't do it yourself!)

Reference comparisons

UFMG COMPUTER SCIENCE

- "My method achieves 0.9 precision"
  - Is it good or bad?
- Evaluation is often meaningless without a reference comparison (aka baseline)
  - Rephrasing: is it **better** or **worse**?
- Choice depends on the original hypothesis
  - Key question: *what are we trying to show?*

Choosing baselines

UFMG COMPUTER SCIENCE

- Vanilla baselines
  - Have the proposed effect turned off
    e.g., CF without dimensionality reduction
- Competing baselines
  - Exploit the proposed effect in a different manner
    e.g., probabilistic dimensionality reduction
- Analytical baselines
  - Can shed light on the tested hypothesis
    e.g., SVD with a varying number of factors

Choosing baselines

UFMG COMPUTER SCIENCE

- Try to stay "within the same framework"
  - In our example using SVD: collaborative filtering
  - Should we compare to a content-based approach?
- Aim for the state-of-the-art
  - In our case, probabilistic dimensionality reduction
- What if no baseline exists (e.g., for new tasks)?
  - Try to adapt methods proposed for a related task
  - As a last resort, use an appropriate vanilla baseline

Parameter tuning

UFMG COMPUTER SCIENCE

- Your method may have parameters
  - Your baselines may also have parameters
  - Example for SVD
    - k: number of latent factors
    - λ: regularization term
    - γ: learning rate
- Which parameters need tuning?
  - Which can stay fixed?
- How to tune?

k-fold cross validation

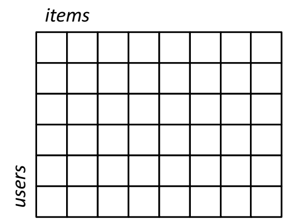
UFMG COMPUTER SCIENCE

- 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?
  - Small values → more efficient
    - k = 2 is a special case (train-then-test)
  - Large values → more training data
    - k = n is a special case (leave-one-out)
  - k = 5 and k = 10 are common

Splitting data

UFMG COMPUTER SCIENCE

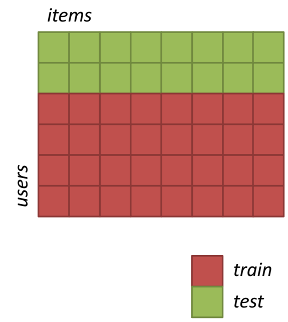
- Split users
- Split items
- Split ratings



Splitting data

UFMG COMPUTER SCIENCE

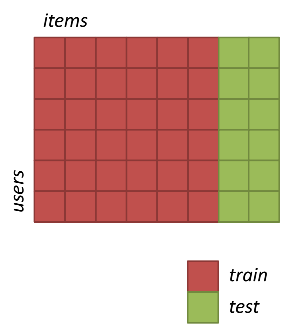
- Split users
  - Learn from some users
  - Predict for others
- Unsuitable for collaborative filtering
  - Useful for cold-start recommendations



Splitting data

UFMG COMPUTER SCIENCE

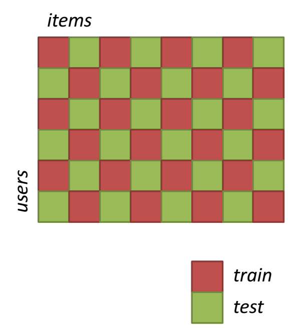
- Split items
  - Learn from some items
  - Predict for others
- Unsuitable for collaborative filtering
  - Useful for cold-start recommendations



Splitting data

UFMG COMPUTER SCIENCE

- Split ratings
  - Learn from some users or from some items
  - Predict for the same users or items
- Suitable for standard collaborative filtering



Splitting data

UFMG COMPUTER SCIENCE

- Split randomly
  - Very common
  - Use to compare with existing results
- Split by time
  - More accurate simulation of user experience
  - Results often worse
- Multiple splits by time
  - Train up to the time of test
  - Best, but expensive

Implicit feedback data

UFMG COMPUTER SCIENCE

- Many recommender contexts have no ratings or other form of explicit data
  - Implicit data may be a good replacement
- Implicit data is cheap and abundant
  - Lots of unary data (view, click, buy)
  - Sometimes implicit non-unary data
    - Clicked vs. saw but skipped

Problems

- No negative examples
  - e.g., log of song plays, clicks
- How do we know if we got it wrong?
  - Or if the user just didn't know about the item?
  - Put differently: how do we avoid punishing the recommender for doing its job?

Mitigation strategies

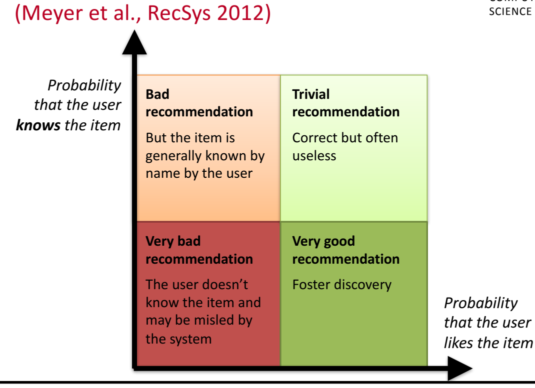
- Record negative feedback as well
  - Skipped music tracks, ranked documents
- Simulate negative feedback
  - For binary metrics (e.g., precision, recall)
    - 1-3 stars: negative
    - 4-5 stars: positive
  - For graded metrics (e.g., nDCG)
    - 1-3 stars: negative
    - 4 stars: positive
    - 5 stars: highly positive

Analyzing results

- Many metrics introduced
  - See next class
- Alternative approaches typically compared based upon their average performance
  - Comparing averages is not enough
- Results must be significant
  - Statistically significant
  - But also ***practically*** significant

An impact model

(Meyer et al., RecSys 2012)



Summary

- Evaluating recommenders is hard
  - Offline evaluation doubly so
  - No substitute for real user-centered testing
- Systems with real users not always available
  - Offline evaluation provides an estimate
- Need to design tests around goals
  - Different methods can achieve different results
- Whatever you do
  - Be aware of limitations

Writing assignment #2

- Choose one of the papers listed below and write a one-page summary describing it:
  - An algorithmic framework for performing collaborative filtering (SIGIR 1999)
  - Item-based collaborative filtering recommendation algorithms (WWW 2001)
- Due Mon, Apr 10 @ 23:55 via Moodle