# Evaluation of Recommender Systems

Radek Pelánek

# Summary

Proper evaluation is important, but really difficult.

# Evaluation: Typical Questions

- Do recommendations work? Do they increase sales? How much?
- Which algorithm should we prefer for our application?
- Which parameter setting is better?

# Evaluation is Important

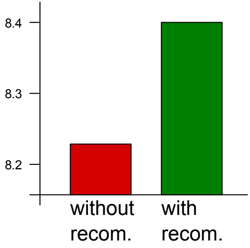
- many choices available: recommender techniques, similarity measures, parameter settings...
- personalization $\Rightarrow$ difficult testing
- impact on revenues may be high
- development is expensive
- intuition may be misleading

# Evaluation is Difficult

- hypothetical examples
- illustrations of flaws in evaluation

# Case I

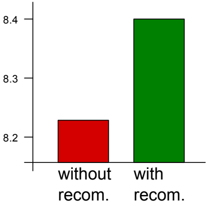
- personalized e-commerce system for selling foobars
- you are a manager
- I'm a developer responsible for recommendations
- this is my graph:




*I did good work. I want bonus pay.*

# Case I: More Details

- personalized e-commerce system for selling foobars
- recommendations available, can be used without recommendations
- comparison:
  - group 1: users using recommendations
  - group 2: users not using recommendations
- measurement: number of visited pages
- result: mean(group 1) $>$ mean(group 2)
- conclusion: recommendations work!




*really?*

# Issues

- what do we measure: number of pages vs sales
- division into groups: potentially biased (self-selection) vs randomized
- statistics: comparison of means is not sufficient
    - role of outliers in the computation of mean
    - statistical significance (p-value)
    - practical significance – effect size
- presentation: y axis

# Case II

- two models for predicting ratings of foobars (1 to 5 stars)
- comparison on historical data
- metric for comparison: how often the model predicts the correct rating
- Model 1 has better score than Model 2
- conclusion: using Model 1 is better than using Model 2

*probing questions?*
*potential flaws?*

# Issues

- over-fitting, train/test set division
- metric:
  - models usually give float; exact match not important
  - we care about the size of the error
- statistical issues again (significance of differences)
- better performance wrt metric $\Rightarrow$ better performance of the recommender system ?

# General Evaluation Problem

- what we care about:
  - long-term sales
  - user satisfaction, trust, happiness, learning, ...
  - fairness, equity, diversity, ...
- what we can measure (and that's still non-trivial):
  - short-term sales
  - ratings, clicks, response times
  - predictive accuracy

# Evaluation Methods

- experimental
  - "online experiments", A/B testing
  - ideally "randomized controlled trial"
  - at least one variable manipulated, units randomly assigned
- non-experimental
  - "offline experiments"
  - historical data
- simulation experiments
  - simulated data, limited validity
  - "ground truth" known, good (not only) for "debugging"

# Offline Experiments

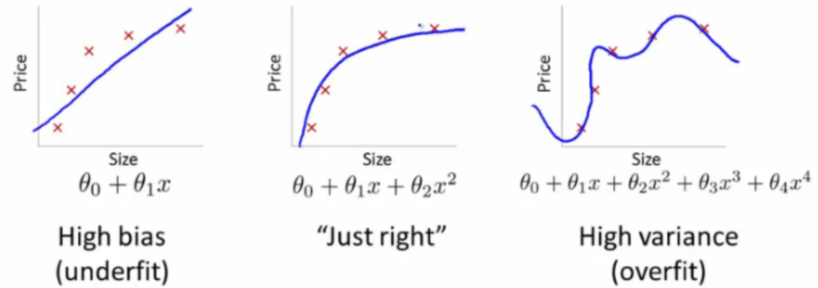
- data: "user, product, rating"
- overfitting, cross-validation
- performance of a model – difference between predicted and actual rating

| predicted | actual |
|---:|---:|
| 2.3 | 2 |
| 4.2 | 3 |
| 4.8 | 5 |
| 2.1 | 4 |
| 3.5 | 1 |
| 3.8 | 4 |

# Overfitting

- model performance good on the data used to build it; poor generalization
- too many parameters
- model of random error (noise)
- typical illustration: polynomial regression

# Overfitting – illustration




`http://kevinbinz.com/tag/overfitting/`

# Cross-validation

- aim: avoid overfitting
- split data: training, testing set
- training set – setting model "parameters" (includes selection of fitting procedure, number of latent classes, and other choices)
- testing set – evaluation of performance
- (validation set)

(more details: machine learning)

# Train and Test Set Division

typical setting (e.g., image classifiers):

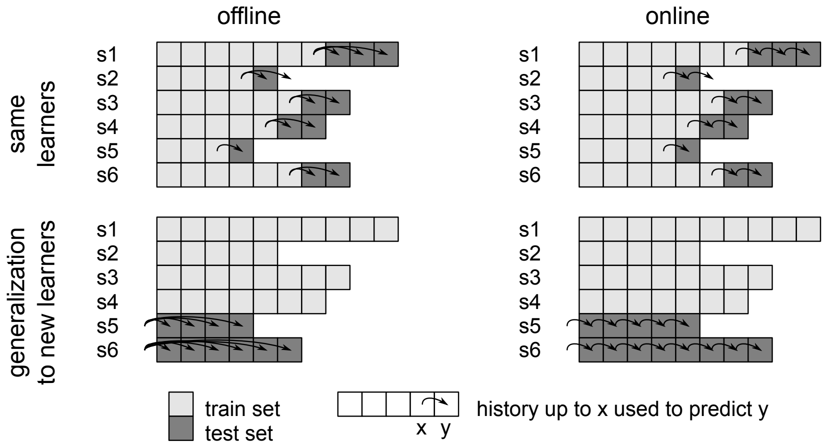
- 80 % train, 20 % test
- randomized selection
- $k$-fold cross validation: $k$ folds, in each turn one fold is the testing set

# Train and Test Set Division: RecSys

not so simple...

- data entries not independent, randomized selection not reasonable
- should the division respect user data? item data?
- temporal aspect: avoid "predicting past from future", respecting time information
- $k$-fold cross validation while respecting time?

# Train/Test Set Division



# Note on Experiments

- (unintentional) “cheating” is easier than you may think
- “data leakage”
  - training data corrupted by some additional information
- useful to separate test set as much as possible

# Metrics

| predicted | actual |
|---|---|
| 2.3 | 2 |
| 4.2 | 3 |
| 4.8 | 5 |
| 2.1 | 4 |
| 3.5 | 1 |
| 3.8 | 4 |

# Metrics

| predicted | actual |
|---|---|
| 2.3 | 2 |
| 4.2 | 3 |
| 4.8 | 5 |
| 2.1 | 4 |
| 3.5 | 1 |
| 3.8 | 4 |

- MAE (mean absolute error)

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |a_i - p_i|$$

- RMSE (root mean square error)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (a_i - p_i)^2}$$

- correlation coefficient

*higher is better? lower is better?*

# When do measures differ?

Describe specific cases of predictions and actual outcomes such that:

1. good RMSE, good correlation
2. good RMSE, poor correlation
3. poor RMSE, good correlation
4. poor RMSE, poor correlation

# Normalization

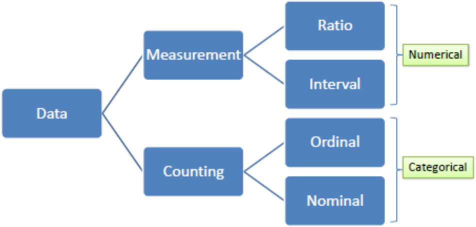
- used to improve interpretation of metrics
- e.g., normalized MAE

$$NMAE = \frac{MAE}{r_{max} - r_{min}}$$

# Note on Likert Scale

1 to 5 "stars" $\sim$ Likert scale (psychometrics)

what kind of data?




http://www.saedsayad.com/data_preparation.htm

# Note on Likert Scale

- 1 to 5 “stars” $\sim$ Likert scale (psychometrics)
  strongly disagree, disagree, neutral, agree, strongly agree
- ordinal data? interval data?
- for ordinal data some operation (like computing averages) are not meaningful
- in RecSys commonly treated as interval data

# Binary Predictions

- like
- click
- buy
- correct answer (educational systems)

prediction: probability $p$

notes:

- (bit surprisingly) more difficult to evaluate properly
- closely related to evaluation of models for weather forecasting (rain tomorrow?)

# Metrics for Binary Predictions

- do not use:
    - MAE: it can be misleading (not a "proper score")
    - correlation: harder to interpret
- reasonable metrics:
    - RMSE
    - log-likelihood

$$LL = \sum_{i=1}^{n} c_i \log(p_i) + (1 - c_i) \log(1 - p_i)$$

# Information Retrieval Metrics

| | | Reality | |
|---|---|---|---|
| | | Actually Good | Actually Bad |
| Prediction | Rated Good | True Positive (tp) | False Positive (fp) |
| | Rated Bad | False Negative (fn) | True Negative (tn) |

All recommended items

All good items

- accuracy
- precision $= \frac{TP}{TP+FP}$
  good items recommended / all recommendations
- recall $= \frac{TP}{TP+FN}$
  good items recommended / all good items
- $F1 = \frac{2TP}{2TP+FP+FN}$
  harmonic mean of precision and recall

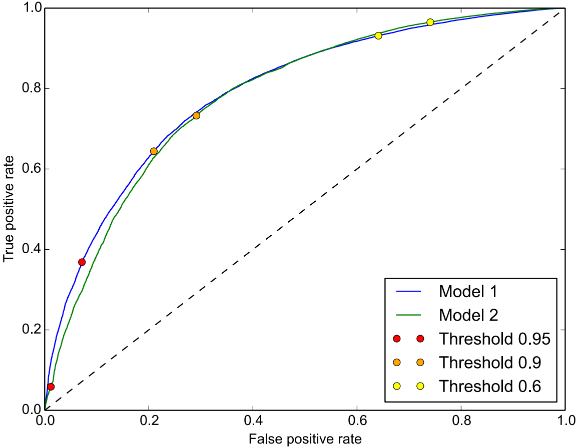
*skewed distribution of classes – hard interpretation (always use baselines)*

# Receiver Operating Characteristic

- to use precision, recall, we need classification into two classes
- probabilistic predictors: value $\in [0, 1]$
- fixed threshold $\Rightarrow$ classification
- what threshold to use? (0.5?)
- evaluate performance over different threshold $\Rightarrow$ Receiver Operating Characteristic (ROC)
- metrics: area under curve (AUC)

*AUC used in many domains, sometimes overused*

# Receiver Operating Characteristic



# Averaging Issues

(relevant for all metrics)

- ratings not distributed uniformly across users/items
- averaging:
  - global
  - per user?
  - per item?
- choice of averaging can significantly influence results
- suitable choice of approach depends on application

# Ranking

- typical output of RS: ordered list of items
- swap on the first place matters more than swap on the 10th place
- ranking metrics – extensions of precision/recall

# Ranking Metrics

- Spearman correlation coefficient
- half-life utility
- liftindex
- discounted cumulative gain
- average precision

specific examples for a case study later

# Metrics

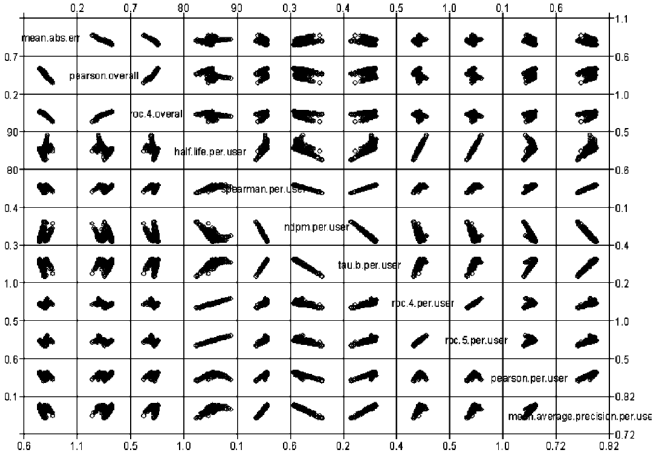
- which metric should we use in evaluation?
- does it matter?

# Metrics

- which metric should we use in evaluation?
- does it matter?

- it depends...
- my advice: use RMSE as the basic metric

# Accuracy Metrics – Comparison




Evaluating collaborative filtering recommender systems, Herlocker et al., 2004

# Beyond Accuracy of Predictions

harder to measure (user studies may be required) $\Rightarrow$ less used (but not less important)

- coverage
- confidence
- novelty, serendipity
- diversity
- utility
- robustness

# Coverage

- What percentage of items can the recommender form predictions for?
- consider systems X and Y:
  - X provides better accuracy than Y
  - X recommends only subset of "easy-to-recommend" items
- one of RecSys aims: exploit "long tail"

# Novelty, Serendipity

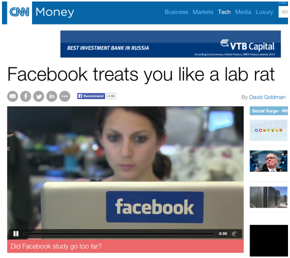
- it is not that difficult to achieve good accuracy on common items
- valuable feature: novelty, serendipity
- serendipity $\sim$ deviation from “natural” prediction
  - successful baseline predictor $P$
  - serendipity – good, but deemed unlikely by $P$

# Diversity

- often we want diverse results
- example: holiday packages
  - bad: 5 packages from the same resort
  - good: 5 packages from different resorts
- measure of diversity – distance of results from each other
- precision-diversity curve

# Online Experiments

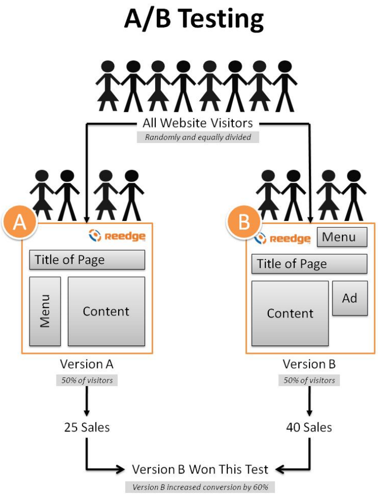
- randomized control trial
- AB testing



# AB Testing

- what is AB testing?
- what is a typical use case?

# AB Testing




https://receiptful.com/blog/ab-testing-for-ecommerce/

# Online Experiments – Comparisons

we usually compare averages (means)

- are data (approximately) normally distributed?
- if not, averages can be misleading
- specifically: presence of outliers $\rightarrow$ use median or log transform

# Statistics Reminder

- statistical hypothesis testing
  *Is my new version really better?*
- t-test, ANOVA, significance, p-value
  *Do I have enough data? Is the observed difference "real" or just due to random fluctuations?*
- error bars
  *How "precise" are obtained estimates?*

note: RecSys – very good opportunity to practice statistics

# Error Bars

Recommended article: Error bars in experimental biology (Cumming, Fidler, Vaux)

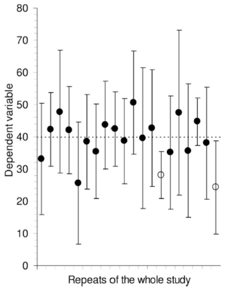



Table I. **Common error bars**

| Error bar | Type | Description | Formula |
|---|---|---|---|
| Range | Descriptive | Amount of spread between the extremes of the data | Highest data point minus the lowest |
| Standard deviation (SD) | Descriptive | Typical or (roughly speaking) average difference between the data points and their mean | $SD = \sqrt{\frac{\sum(X-M)^2}{n-1}}$ |
| Standard error (SE) | Inferential | A measure of how variable the mean will be, if you repeat the whole study many times | $SE = SD/\sqrt{n}$ |
| Confidence interval (CI), usually 95% CI | Inferential | A range of values you can be 95% confident contains the true mean | $M \pm t_{(n-1)} \times SE$, where $t_{(n-1)}$ is a critical value of $t$. If $n$ is 10 or more, the 95% CI is approximately $M \pm 2 \times SE$. |

# Warning

What you should never do:
*report mean value with precision up to 10 decimal places (just because that is the way your program printed the computed value)*

Rather:
*present only "meaningful" values, report "uncertainty" of your values*

# Practical Advice

Recommended:

- author Ron Kohavi
- paper *Seven rules of thumb for web site experimenters*
- lecture *Online Controlled Experiments: Lessons from Running A/B/n Tests for 12 Years*
  `https://www.youtube.com/watch?v=qtboCGd_hTA`

context: mainly search engines (but highly relevant for evaluation of recommender systems)

# Seven Rules of Thumb for Web Site Experimenters

1. Small changes can have a big impact to key metrics
2. Changes rarely have a big positive impact to key metrics
3. Your mileage will vary
4. Speed matters a lot
5. Reducing abandonment is hard, shifting clicks is easy
6. Avoid complex designs: iterate
7. Have enough users

# Number of Users and Detectable Differences

*How many users do I need for a meaningfull AB experiment?*

- hundreds of users – significantly different versions of the system
- tens of thousands of users – different parametrizations of one algorithm
- millions of user – “shades of blue”

# Comparing Algorithms Without AB Test

meaningful comparison can be achieved even without splitting users

example:

- two recommendation algorithms A, B
- each picks 3 items
- user is presented with all 6 items (in interleaved order)
- which items users choose more often?

Basic evaluation: this type of comparison, "on ourselves", compare to "random recommendations"

# Simulated Experiments

- simulate data according to a chosen model of users
- add some noise
- advantages:
  - known "ground truth"
  - simple, cheap, fast
  - very useful for testing implementation (bugs in models)
  - insight into behaviour, sensitivity analysis
- disadvantage: results are just consequence of used assumptions

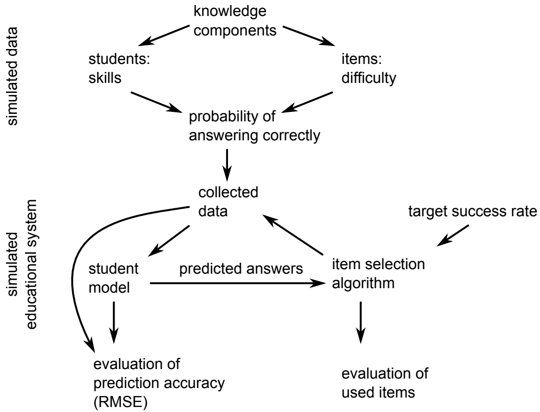
# Simple Simulated Setting: Personas

- very simple “manual” simulation
- artificial users (“personas”) with strong preferences
- clear expectations, for which we can check the behavior of the algorithm
- recipe recommendation setting: vegetarian, strict diet, nut alergy, strong preference for Indian food, ...

# Simulated Experiments: Simple Example

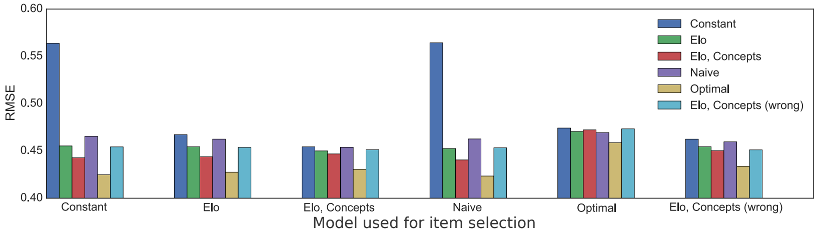
- setting: movies
- simulated users: each user likes some genres (randomly chosen)
- simulated ratings: based on the genre (1 or 4 stars) + random noise
- compute item-item similarity based on ratings
- do the results correspond to genres? how much data needed for convergence?

# Simulated Experiments: Realistic Example




Exploring the Role of Small Differences in Predictive Accuracy using Simulated Data

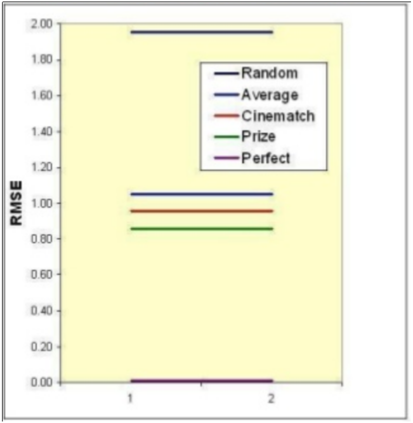
# Simulated Experiments: Example



# Interpretation of Results

- what do the numbers mean?
- what do (small) differences mean?
- are they significant?
    - statistically?
    - practically?

# Interpretation of Results




Introduction to Recommender Systems, Xavier Amatriain

# Magic Barrier

- noise in user ratings / behaviour
- magic barrier – unknown level of prediction accuracy a recommender system can attain
- are we close?
- is further improvement important?

# Summary

Proper evaluation is difficult...

- not clear what to measure, how
- things we care about are hard to measure
- many choices that can influence results
  - metrics (RMSE, AUC, ranking...) and their details (thresholds, normalization, averaging...)
  - experimental settings
- it is easy to cheat (unintentionally), overfit

specific examples (case studies) in next lectures

# Evaluation and Projects

What kind of evaluation is relevant?

- offline experiments, historical data
- online experiments (AB testing)
- simulated data

How will you perform the evaluation?