Evaluation of Recommender Systems

Radek Pelánek
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:

![Graph showing comparison between performance with and without recommendations.]

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 of 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

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 ⇒ better performance of the recommender system?
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

<table>
<thead>
<tr>
<th>predicted</th>
<th>actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3</td>
<td>2</td>
</tr>
<tr>
<td>4.2</td>
<td>3</td>
</tr>
<tr>
<td>4.8</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>4</td>
</tr>
<tr>
<td>3.5</td>
<td>1</td>
</tr>
<tr>
<td>3.8</td>
<td>4</td>
</tr>
</tbody>
</table>
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

- High bias (underfit)
- "Just right"
- High variance (overfit)

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)
Cross-validation

train/test set division:
- typical ratio: 80 % train, 20 % test
- $N$-fold cross validation: $N$ folds, in each turn one fold is the testing set
- how to divide the data: time, user-stratified, ...
Train/Test Set Division

Bayesian Knowledge Tracing, Logistic Models, and Beyond: An Overview of Learner Modeling Techniques
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
<table>
<thead>
<tr>
<th>predicted</th>
<th>actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3</td>
<td>2</td>
</tr>
<tr>
<td>4.2</td>
<td>3</td>
</tr>
<tr>
<td>4.8</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>4</td>
</tr>
<tr>
<td>3.5</td>
<td>1</td>
</tr>
<tr>
<td>3.8</td>
<td>4</td>
</tr>
</tbody>
</table>
### Metrics

<table>
<thead>
<tr>
<th>predicted</th>
<th>actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3</td>
<td>2</td>
</tr>
<tr>
<td>4.2</td>
<td>3</td>
</tr>
<tr>
<td>4.8</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>4</td>
</tr>
<tr>
<td>3.5</td>
<td>1</td>
</tr>
<tr>
<td>3.8</td>
<td>4</td>
</tr>
</tbody>
</table>

- **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
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” ~ Likert scale (psychometrics)
what kind of data?

1 to 5 “stars” \sim \text{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

<table>
<thead>
<tr>
<th>Prediction</th>
<th>Reality</th>
<th>All good items</th>
<th>All recommended items</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rated Good</td>
<td>Actually Good</td>
<td>True Positive (tp)</td>
<td>Good items recommended / all recommendations</td>
</tr>
<tr>
<td>Rated Bad</td>
<td>Actually Bad</td>
<td>False Negative (fn)</td>
<td>Good items recommended / all good items</td>
</tr>
<tr>
<td></td>
<td></td>
<td>False Positive (fp)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>True Negative (tn)</td>
<td></td>
</tr>
</tbody>
</table>

- accuracy
- precision = \(\frac{TP}{TP + FP}\)
good items recommended / all recommendations
- recall = \(\frac{TP}{TP + FN}\)
good items recommended / all good items
- \(F_1 = \frac{2TP}{2TP + FP + FN}\) 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

Metrics for Evaluation of Student Models
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

Measuring Predictive Performance of User Models: The Details Matter
**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?

It depends...

My advice: use RMSE as the basic metric.
which metric should we use in evaluation?
does it matter?

it depends...
my advice: use RMSE as the basic metric
Evaluating collaborative filtering recommender systems, Herlocker et al., 2004
Beyond Accuracy of Predictions

harder to measure (user studies may be required) ⇒ 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

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 → 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 1. Common error bars

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

- 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 Recommendation 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
Simulated Experiments: Example

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

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

Predictions for movies dataset:

- focus on evaluation
- offline experiments
- proper comparison of different models
- attention to evaluation issues: choice of metric, overfitting, cross-validation
New System for Simple Domain

- projects: fun facts, cocktails, ...
- online experiments, AB testing, collecting data (> 50 users)
- comparing different versions of recommendations (random, simple popularity based, content based, collaborative filtering)
- report on results, preferably including statistical issues (significance of results)
Prototypes with More Complex Data

- projects: boardgames, points of interest, ...
- descriptive statistics of available data (distribution of ratings, items into categories, ...), design/selection of “features”
- when applicable: basic evaluation of predictions / recommendations on historical data
- implementation of several recommendations
- simple AB testing – at least “qualitative” evaluation “on ourselves” (can we recognize random recommendation from more sophisticated one?)
- proposal for more complex evaluation (during presentation)