Evaluation of recommender systems in streaming environments

João Vinagre, Alípio Jorge, João Gama

INESC TEC, University of Porto

October 2014
Recommender systems evaluation

- A/B tests
- Archival datasets
  - In lab controlled environments
  - Holdout
  - Convenient
  - Does not translate directly into good performance

We propose a **prequential evaluation** methodology

- Evaluates algorithms that learn from continuous flows of data
- Motivates the use of incremental algorithms
- Sees learning as a continuous and never finished process
- Applicable online and offline
Issues with batch evaluation

Classical approaches

- Dataset ordering
  - natural sequence of the data
  - test cases and hidden items picked randomly
- Time awareness
  - beware of shuffling
- Incremental updates
  - not convenient
- Session grouping
  - needs whole data set to sessionize
- Recommendation bias
  - recommendations influence observations
Evaluation methodologies
Classical approaches

- **Holdout** - used in **batch** evaluation
  1. **Split** the dataset in **training set** and **testing set**
  2. Obtain **recommendation model** using training set
  3. **Hide** a few ratings from each transaction in the testing set
  4. **Try** recommendation model on testing set and compare predictions with hidden values
Offline evaluation
Protocols and metrics

- Metrics w.r.t. type of prediction task
  - numeric ratings
    - measure the error on hidden ratings using MAE or RMSE (regression)
  - item prediction
    - Precision, Recall, F1 (classification)
    - MAP, Normalized Discounted Cumulative Gain (ranking)

- Protocols
  - separating observables from hidden
  - All-but-N, Given-N, All-but-One
Prequential evaluation
Evaluation in a non-stationary environment

- Resolves issues mentioned before
- Suitable for non-stationary environments
- **Test-then-learn** procedure for each new data point
- Can be used with incremental and batch algorithms
**Continuous monitoring** of the system’s performance over time;

Several metrics can be captured simultaneously;

If available, **user feedback can be included** in the loop;

**Real-time statistics can be integrated** in the algorithm – e.g. automatic parameter adjustment, drift/shift detection, triggering batch retraining;

- In ensembles, relative weights of individual algorithms can be adjusted;

Applicable to **item and rating prediction**;

**Applicable online and offline** – experiments are naturally reproducible if the same data sequence is available.
### Applying it

#### Algorithms and overall accuracy

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Algorithm</th>
<th>Recall@10</th>
<th>Update time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Music-listen</strong></td>
<td>BPRMF</td>
<td>0.028</td>
<td>0.846 ms</td>
</tr>
<tr>
<td></td>
<td>ISGD</td>
<td>0.061</td>
<td><strong>0.118 ms</strong></td>
</tr>
<tr>
<td></td>
<td>UserKNN</td>
<td><strong>0.139</strong></td>
<td>328.917 ms</td>
</tr>
<tr>
<td><strong>Lastfm-600k</strong></td>
<td>BPRMF</td>
<td>0.003</td>
<td>28.061 ms</td>
</tr>
<tr>
<td></td>
<td>ISGD</td>
<td><strong>0.034</strong></td>
<td>1.106 ms</td>
</tr>
<tr>
<td></td>
<td>UserKNN</td>
<td>0.006</td>
<td>290.133 ms</td>
</tr>
<tr>
<td><strong>Music-playlist</strong></td>
<td>BPRMF</td>
<td>0.020</td>
<td>1.889 ms</td>
</tr>
<tr>
<td></td>
<td>ISGD</td>
<td><strong>0.171</strong></td>
<td><strong>0.949 ms</strong></td>
</tr>
<tr>
<td></td>
<td>UserKNN</td>
<td>0.132</td>
<td>190.250 ms</td>
</tr>
<tr>
<td><strong>Movielens-1M</strong></td>
<td>BPRMF</td>
<td>0.080</td>
<td>0.173 ms</td>
</tr>
<tr>
<td></td>
<td>ISGD</td>
<td>0.050</td>
<td><strong>0.016 ms</strong></td>
</tr>
<tr>
<td></td>
<td>UserKNN</td>
<td><strong>0.110</strong></td>
<td>84.927 ms</td>
</tr>
</tbody>
</table>
Monitor the learning process as it evolves
Visualized with sliding window (2000 to 5000 points)
General results are confirmed but with more detail (e.g. ISGD vs UKNN on playlist)
Applying it
Dynamic statistical significance testing

- Significance tests using signed McNemar
- Applied over sliding windows (same size as above)
- Visualized in three colors
- ISGD vs the other two algorithms
Conclusions and future work

We proposed a **prequential evaluation framework**
- Monitors a variety of recommender systems’ metrics
- Monitors **statistical significance over time**
- We show an application with 3 incremental algorithms and 4 datasets

Future work
- Dynamic sliding window size
- Dynamic parameter setting