Temporal-Contextual Recommendation in Real-Time

Yifei Ma*  Balakrishnan (Murali) Narayanaswamy*
Haibin Lin  Hao Ding

ytheim@amazon.com
*Equal contribution authors
Objective

• Real-time recommendation without ML/DL experience
• One network architecture to fit most application scenarios
Background

• Most rec sys are built from user/item latent representations

• Sequence models with ordered user histories

• However, there are still gaps in practice
  • Metadata
  • Temporal drifts
Our Contributions

- **Temporal**
  - Time-delta changes
  - Explicit contexts

- **Debiased**

- **Contextual**
  - Recommend by item meta-data

- Continuous item cold-start
- Negative sampling
Sequence Model and Context Changes

• Exponential Moving Average (EMA) – smooth changes in context

• Gating Recurrent Units (GRUs) – implicit context changes

Marry has a little lamb

Subject Verb Object

• Back Propagation Through Time (BPTT) – active memory of ~15 items
Explicit Context Changes

👍Skip connection achieves long-term memory effects [Quadrana et al., 2017]
👎Slower in minibatch training due to irregular computation flows

Remap time-delta as RNN input?

Efficient implementation inspired from https://gluon-nlp.mxnet.io/
Recommendation Changes with Time-Delta

• Δt between last visit and next recommendation
• As Δt increases, items become more general

<table>
<thead>
<tr>
<th>Δt</th>
<th>title</th>
<th>genres</th>
<th>popularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Purple Rose of Cairo</td>
<td>Comedy</td>
<td>Drama</td>
</tr>
<tr>
<td>60</td>
<td>Unbearable Lightness</td>
<td>Drama</td>
<td>0.000209</td>
</tr>
<tr>
<td>3600</td>
<td>Local Hero</td>
<td>Comedy</td>
<td>0.000195</td>
</tr>
<tr>
<td>86400</td>
<td>Big</td>
<td>Comedy</td>
<td>Drama</td>
</tr>
</tbody>
</table>

Synthetic Memory Benchmark

Reoccurring purchase after a noisy session; memory capacity = $t$

RNN: not sensitive to explicit context changes

HGRU [Quadrana et al.]: $50 < t < 60$

Irregular computation flows, 10x slower

HRNN [ours]: $10 < t < 20$

(Meet most use-cases)
Speed and Performance Trade-Offs

• Subreddit-interactions prediction (18271 users, 27453 subreddits)
• HRNN: GRU + time-delta (ours) +0.16
• HGRU: GRU + time-delta + hierarchy +0.05 10x slower
• HRNN (ours) meets most use cases

<table>
<thead>
<tr>
<th>Hit@5 on reddit recommendation</th>
<th>GRU</th>
<th>HRNN</th>
<th>HGRU</th>
<th>Pop</th>
<th>BPR^5</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 hidden</td>
<td>0.26</td>
<td>0.42</td>
<td>0.47</td>
<td>0.11</td>
<td>0.39</td>
</tr>
</tbody>
</table>

https://www.kaggle.com/colemaclean/subreddit-interactions
Speed and Performance Trade-Offs

- Subreddit-interactions prediction (18271 users, 27453 subreddits)
- HRNN: GRU + time-delta (ours) +0.16
- HGRU: GRU + time-delta + hierarchy +0.05 10x slower
- HRNN (ours) meets most use cases

### Hit@5 on reddit recommendation

<table>
<thead>
<tr>
<th></th>
<th>GRU</th>
<th>HRNN</th>
<th>HGRU</th>
<th>Pop</th>
<th>BPR^5</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 hidden</td>
<td>0.26</td>
<td>0.42</td>
<td>0.47</td>
<td>0.11</td>
<td>0.39</td>
</tr>
<tr>
<td>200 hidden</td>
<td>0.55</td>
<td>0.55</td>
<td>0.55</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Caveat:
  - Largest benefit comes from model tuning
  - Public datasets have limited complexity
Contributions

- Temporal
- Contextual
- Debiased

Time-delta changes
Explicit contexts

Continuous item cold-start
Negative sampling

Recommend by item meta-data
Recommend by Item Meta-Data

Factorization machines [Rendle 2011]: a game of point collection

<table>
<thead>
<tr>
<th>Item</th>
<th>Rain</th>
<th>No rain</th>
<th>Wind</th>
<th>No wind</th>
<th>Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Umbrella ($w_1$)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Jacket ($w_2$)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Nothing ($w_3$)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Context</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Meta score = [context meta-data] * [item meta-data]
HRNN score = [user embedding] * [item embedding]
A hybrid model adapts to both types of data sources!
Recommend New Movies in a Hold-Out Set

- Remove a random half of unique movies from training
- Expect similar genres top rank top within the hold-out set

**Genres in user history**  
<table>
<thead>
<tr>
<th>Movie</th>
<th>Genres</th>
</tr>
</thead>
<tbody>
<tr>
<td>Braveheart (1995)</td>
<td>Action</td>
</tr>
<tr>
<td>Farinelli: il castrato (1994)</td>
<td>Drama</td>
</tr>
<tr>
<td>Edward Scissorhands (1990)</td>
<td>Drama</td>
</tr>
<tr>
<td>Back to the Future (1985)</td>
<td>Action</td>
</tr>
<tr>
<td>Road Warrior, The (Mad Max 2) (1981)</td>
<td>Action</td>
</tr>
<tr>
<td>Grumpy Old Men (1993)</td>
<td>Comedy</td>
</tr>
<tr>
<td>Halloween: The Curse of Michael Myers (Halloween II)</td>
<td>Horror</td>
</tr>
</tbody>
</table>

**Genres in top-ranked hold-out movies**  
<table>
<thead>
<tr>
<th>Movie</th>
<th>Genres</th>
</tr>
</thead>
<tbody>
<tr>
<td>TerrorVision (1986)</td>
<td>Comedy</td>
</tr>
<tr>
<td>Tremors II: Aftershocks (1996)</td>
<td>Comedy</td>
</tr>
<tr>
<td>Evil Aliens (2005)</td>
<td>Comedy</td>
</tr>
<tr>
<td>Thing with Two Heads, The (1972)</td>
<td>Comedy</td>
</tr>
</tbody>
</table>

**Limitations?**
- When warm items present, underscore cold items due to low confidence
Contributions

- Temporal
- Contextual
- Debiased

Time-delta changes
Explicit contexts

Continuous item cold-start
Negative sampling

Recommend by item meta-data
Temporal Drifts in Item Popularity

• In news or media domains, new items easily drive >50% of traffic;
• Count global item popularity in Time Intervals (T)
  • $T = 12$ hours, 30 minutes, etc.
  • Test with previous $T$
  • Null = bootstrap from same $T$

• How to use the stale information?

Stale information after $T = 12$ hours

No significant drifts in $T = 30$ minutes
Item Trend Debiasing

We learn to attribute between:

\[ P(\text{exposure} \mid \text{freshness, popularity}) \]
\[ P(\text{consumption} \mid \text{exposure, user preference}) \]

Related to Inverse Propensity Scoring
More theoretical insights in [Ma et al., 2019]
Model inspired by [Wu et al., 2017]
Negative Sampling

Vanilla RNN softmax on every item sample from popular items + debiasing

<table>
<thead>
<tr>
<th>Output size (m)</th>
<th>Output</th>
<th>ml-1m</th>
<th>ml-10m</th>
<th>ml-20m</th>
<th>Taobao</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IS</td>
<td>62</td>
<td>255</td>
<td>362</td>
<td>1087</td>
</tr>
<tr>
<td></td>
<td>Dense</td>
<td>1683</td>
<td>65134</td>
<td>131263</td>
<td>1183451</td>
</tr>
<tr>
<td>Throughput (#items/sec)</td>
<td>IS</td>
<td>23k</td>
<td>20k</td>
<td>20k</td>
<td>7.8k</td>
</tr>
<tr>
<td></td>
<td>Dense</td>
<td>23k</td>
<td>23k</td>
<td>17k</td>
<td>631</td>
</tr>
<tr>
<td>PPL</td>
<td>IS</td>
<td>377</td>
<td>405</td>
<td>455</td>
<td>17.6k</td>
</tr>
<tr>
<td></td>
<td>Dense</td>
<td>409</td>
<td>439</td>
<td>494</td>
<td>119k</td>
</tr>
<tr>
<td>NDCG</td>
<td>IS</td>
<td>0.128</td>
<td>0.123</td>
<td>0.12</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>Dense</td>
<td>0.123</td>
<td>0.119</td>
<td>0.115</td>
<td>0.08</td>
</tr>
</tbody>
</table>

10x speed-up; 2x better performance in unit time
Bring Everything Together

Temporal

- Time-delta changes
- Explicit contexts

Debiased

Continuous item cold-start
Negative sampling

Contextual

Recommend by item meta-data
Case Study: Outbrain News Recommendation

14 days of news articles; Online VS-KNN > Offline GRU

Reproduce on larger set of items
Sample 1% of users
Filter user_activities >= 10
Filter item_views >= 100

Retain 4M views from
377k users & 31k items

https://www.kaggle.com/c/outbrain-click-prediction
Case Study: Outbrain News Recommendation

Item trend debiasing
- Item freshness since release
- Total views in last hour

Improves validation & test with long histories & futures!

Recent POP

[Ma et al., 2018]
Case Study: Outbrain News Recommendation

Temporal

Debiased

Contextual

- [Ma et al., 2018]
- Ours-Everything

Better fit

More stable

hit at 25

Ours-Everything

[Ma et al., 2018]

validation

test

2016

Jun 23

24

25

26
Case Study: Outbrain News Recommendation

VS-KNN depends on good cold-start item recs in the first place
Case Study: Outbrain News Recommendation

Retrained Ours-Everything seems to bridge the gaps.

Retrain every 2 hours;
Delay by 1 hour;
Serve for 2 hours.
Conclusion

• Fill the practicality gap of session-aware RNN models
• Use temporal, contextual, side information
• Address time-varying confounding variables
• More theoretical details in paper
• Adopted by Amazon Personalize

https://github.com/aws-samples/amazon-personalize-samples/