Deep Learning for Recommender Systems

Marcel Kurovski

Karlsruhe, October 25th 2017
About Me

- Industrial Engineer (M.Sc.)
- Data Scientist at inovex
- Machine Learning – focus on Deep Learning
- Masterthesis:
  Deep Learning for Recommender Systems:
  Joint Learning of Preference and Similarity

Marcel Kurovski
Agenda

1. Motivation
2. State-of-the-Art
3. Vehicle Recommendations with Deep Learning
4. ACM RecSys Conference 2017
5. Discussion
Annual Data Sphere increases exponentially

[Graph showing exponential increase of data from 2010 to 2025, with labels for "Information to Humans" and "Processing Capacity".

International Data Corporation: Data Age 2025 study, April 2017]
Information Overload

https://www.linkedin.com/pulse/its-information-overload-filter-failure-productivity-industry-zavats/
“It’s not information overload. It’s filter failure.”

- Clay Shirky
Collaborative Filtering

Diagram showing a matrix representation of users (m) and items (n) with some ratings indicated (1s and ?s). The matrix is used to illustrate how collaborative filtering works.
Collaborative Filtering

Hey, I like tracks P, Q, R, S!

Well, I like tracks Q, R, S, T!

Then you should check out track P!

Nice! Btw try track T!
Matrix Factorization

\[
\min_{x^*, y^*} \sum_{u, i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left( \sum_u ||x_u||^2 + \sum_i ||y_i||^2 \right)
\]

Cold Start
SPARSITY
## Sparsity Comparison

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Users</th>
<th>Items</th>
<th>Ratings</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movielens 1M</td>
<td>6,040</td>
<td>3,883</td>
<td>1,000,209</td>
<td>4.2600%</td>
</tr>
<tr>
<td>Movielens 10M</td>
<td>69,878</td>
<td>10,681</td>
<td>10,000,054</td>
<td>1.3400%</td>
</tr>
<tr>
<td>Movielens 20M</td>
<td>138,493</td>
<td>27,278</td>
<td>20,000,263</td>
<td>0.5300%</td>
</tr>
<tr>
<td>Jester</td>
<td>124,113</td>
<td>150</td>
<td>5,865,235</td>
<td>31.5000%</td>
</tr>
<tr>
<td>Book-Crossing</td>
<td>92,107</td>
<td>271,379</td>
<td>1,031,175</td>
<td>0.0041%</td>
</tr>
<tr>
<td>Last.fm</td>
<td>1,892</td>
<td>17,632</td>
<td>92,834</td>
<td>0.2800%</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>5,583,724</td>
<td>4,936,761</td>
<td>417,996,366</td>
<td>0.0015%</td>
</tr>
<tr>
<td>Vehicles All</td>
<td>100,000</td>
<td>1,711,190</td>
<td>7,849,006</td>
<td>0.0046%</td>
</tr>
<tr>
<td>Vehicles Train</td>
<td>100,000</td>
<td>1,711,190</td>
<td>6,474,853</td>
<td>0.0038%</td>
</tr>
<tr>
<td>Vehicles Test</td>
<td>100,000</td>
<td>1,711,190</td>
<td>1,374,153</td>
<td>0.0008%</td>
</tr>
</tbody>
</table>

adapted from http://www.kdnuggets.com/2016/02/nine-datasets-investigating-recommender-systems.html
Sparsity Comparison

MovieLens 1M: 4.26%

MovieLens 20M: 0.53%

Last.fm: 0.28%

Vehicles All: 0.0046%
Content-based Filtering

Diagram showing connections between users and items with features like age, gender, history, model, color, and mileage. A matrix is also shown with rows for users and columns for items, indicating preferences or dislikes.

$n$ Items

$m$ Users

1 1 1

? 1 ? ?

1 1 1

1 1
“Deep Learning becomes a general-purpose solution for nearly all learning problems.”

- Covington et al.
Motivation: Deep Learning for RecSys

- Information Overload
- Deep Learning
- Information Filtering
- Learning Problem
- Recommender Systems
Agenda

1. Motivation
2. State-of-the-Art
3. Vehicle
   Recommendations with Deep Learning
4. ACM RecSys Conference 2017
5. Discussion
Recommendations are everywhere
„The company reported a 29% sales increase to $12.83 billion [...] Amazon has integrated recommendations into nearly every part of the purchasing process from product discovery to checkout.“

http://fortune.com/2012/07/30/amazons-recommendation-secret/
“Our recommender system is used on most screens of the Netflix product beyond the homepage, and in total influences choice for about 80% of hours streamed at Netflix. The remaining 20% comes from search [...]“
## Domains and Types for DLRS

<table>
<thead>
<tr>
<th>Domain</th>
<th>Example</th>
<th>Year</th>
<th>Example</th>
<th>Year</th>
<th>Example</th>
<th>Year</th>
<th>Example</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNNs</td>
<td>YouTube</td>
<td>2015</td>
<td>Yahoo!</td>
<td>2015</td>
<td>Microsoft</td>
<td>2015</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNNs</td>
<td>Spotify</td>
<td>2013</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RNNs</td>
<td>Telefonica</td>
<td>2017</td>
<td>XING</td>
<td>2017</td>
<td>citeulike</td>
<td>2016</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AEs</td>
<td>Netflix</td>
<td>2015</td>
<td></td>
<td></td>
<td>citeulike</td>
<td>2015</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DLRS: Deep Learning based Recommender Systems**
DNNs for Video-Recommendations (1)
DNNs for Video-Recommendations

\[ P(w_t = i|U, C) = \frac{e^{v_{i,u}}}{\sum_{j \in V} e^{v_{j,u}}} \]
DNNs for Video-Recommendations (3)

Deep Candidate Generation

Deep Ranking

Wide and Deep Learning for App-Recos (1)

https://research.googleblog.com/2016/06/wide-deep-learning-better-together-with.html
Wide and Deep Learning for App-Recos (2)

Deep Component

Embeddings

Wide Component

https://research.googleblog.com/2016/06/wide-deep-learning-better-together-with.html
Agenda

1. Motivation
2. State-of-the-Art
3. Vehicle Recommendations with Deep Learning
4. ACM RecSys Conference 2017
5. Discussion
Vehicle Recommendations: End-to-End Approach

Data → Preprocessing → Classifier Training → Candidate Generation → Serving → Ranking
Vehicle Recommendations: Technologies

Hardware
- GPU-Server
  - NVIDIA Tesla K80
  - 4x Intel Xeon 3.5 GHz
  - 64GB RAM, 850GB Disk
- AWS Instances
Vehicle Recommendations: Data

Users & Interactions

- **Registered Users**
- Sample Size: **100,000 Users**
- **Events**: View, Bookmark, Contact

<table>
<thead>
<tr>
<th>$F_{cont}$</th>
<th>$F_{cat}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>consumption</td>
<td>airbag</td>
</tr>
<tr>
<td>first registration</td>
<td>category</td>
</tr>
<tr>
<td>latitude</td>
<td>climatisation</td>
</tr>
<tr>
<td>longitude</td>
<td>color</td>
</tr>
<tr>
<td>mileage</td>
<td>condition</td>
</tr>
<tr>
<td>price</td>
<td>country</td>
</tr>
<tr>
<td></td>
<td>doors</td>
</tr>
<tr>
<td></td>
<td>make ID</td>
</tr>
<tr>
<td></td>
<td>model ID</td>
</tr>
<tr>
<td></td>
<td>previous owners</td>
</tr>
<tr>
<td></td>
<td>seats</td>
</tr>
<tr>
<td></td>
<td>fuel</td>
</tr>
<tr>
<td></td>
<td>subcategory</td>
</tr>
<tr>
<td></td>
<td>transmission</td>
</tr>
</tbody>
</table>

Time-based Train-Test-Split

- April 2017
- May

**CW 14** | **CW 15** | **CW 16** | **CW 17** | **CW 18**

Training

Test

85 : 15
What does ‘Embedding‘ actually mean?

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>blue</td>
<td>0</td>
</tr>
<tr>
<td>1.</td>
<td>green</td>
<td>0</td>
</tr>
<tr>
<td>2.</td>
<td>red</td>
<td>0</td>
</tr>
<tr>
<td>3.</td>
<td>yellow</td>
<td>0</td>
</tr>
<tr>
<td>4.</td>
<td>orange</td>
<td>0</td>
</tr>
<tr>
<td>5.</td>
<td>black</td>
<td>1</td>
</tr>
<tr>
<td>6.</td>
<td>white</td>
<td>0</td>
</tr>
<tr>
<td>7.</td>
<td>brown</td>
<td>0</td>
</tr>
</tbody>
</table>

**One-Hot-Encoding**

**binary Embedding**
Deep Component

- ELU (64)
- ELU (128)
- ELU (256)

Wide Component

- Cross user-item transformations

Embedding $\mathbf{u}_{\text{cat}}$

- $\mathbf{u}_{\text{cont}}$
- $\mathbf{e}_{\text{climatisation}}$
- $\mathbf{e}_{\text{color}}$
- $\ldots$
- $\mathbf{e}_{\text{transmission}}$

Embedding $\mathbf{i}_{\text{cat}}$

- $\mathbf{i}_{\text{cont}}$
- $\mathbf{c}_{\text{consumption}}$
- $\mathbf{f}_{\text{first_reg}}$
- $\ldots$
- $\mathbf{p}_{\text{price}}$

Feature transformation

- One-many-encoding
- One-hot-encoding

Output

$\text{Probability that user } u \text{ likes vehicle } i$
Vehicle Recommendations: End-to-End Approach
Vehicle Recommendations: Ranking

**Target:** Rank Candidates descendantly by interaction probability

\[ k \leq T \]
Vehicle Recommendations: End-to-End Approach

- Data
- Preprocessing
- Classifier Training
- Ranking
- Serving
- Candidate Generation

36
Results: DLRS Recommendation Relevance

- DLRS Recommendation Relevance
  - MAP@k for different models and parameters:
    - CF ($\alpha=0.03$, d=100)
    - Hybrid CF-CBF ($\alpha=0.03$, d=100)
    - Hybrid CF-CBF ($\alpha=0.03$, d=700)
    - DL (multi-cos)

Graph indicating improvements in MAP@k for different values of $k$ (1, 5, 10) with $+20\%$ and $+65\%$ increases in performance.
Agenda

1. Motivation
2. State-of-the-Art
3. Vehicle Recommendations with Deep Learning
4. ACM RecSys Conference 2017
5. Discussion
ACM RecSys Conference 2017

627 Participants
43 Countries

„Accuracy doesn’t matter – impact does!“

„Try to not use MovieLens“

„People are most curious about themselves“
RNNs for Video and Job Recommendations
"We can only see a short distance ahead, but we can see plenty there that needs to be done."

- Alan Turing
References


