Performance Comparison of Neural and Non-Neural Approaches to Session-based Recommendation

Malte Ludewig  
TU Dortmund, Germany  
malte.ludewig@tu-dortmund.de

Noemi Mauro  
University of Torino, Italy  
noemi.mauro@unito.it

Sara Latifi  
University of Klagenfurt, Austria  
sara.latifi@aau.at

Dietmar Jannach  
University of Klagenfurt, Austria  
dietmar.jannach@aau.at

ABSTRACT
The benefits of neural approaches are undisputed in many application areas. However, today’s research practice in applied machine learning—where researchers often use a variety of baselines, datasets, and evaluation procedures—can make it difficult to understand how much progress is actually achieved through novel technical approaches. In this work, we focus on the fast-developing area of session-based recommendation and aim to contribute to a better understanding of what represents the state-of-the-art.

To that purpose, we have conducted an extensive set of experiments, using a variety of datasets, in which we benchmarked four neural approaches that were published in the last three years against each other and against a set of simpler baseline techniques, e.g., based on nearest neighbors. The evaluation of the algorithms under the exact same conditions revealed that the benefits of applying today’s neural approaches to session-based recommendations are still limited. In the majority of the cases, and in particular when precision and recall are used, it turned out that simple techniques in most cases outperform recent neural approaches. Our findings therefore point to certain major limitations of today’s research practice.

By sharing our evaluation framework publicly, we hope that some of these limitations can be overcome in the future.

CCS CONCEPTS
• Information systems → Recommender systems.

KEYWORDS
Session-based Recommendation; Evaluation; Reproducibility

1 INTRODUCTION
In recent years, we could observe an increased research interest in session-based recommendation problems. In such settings, the problem is not to make relevance predictions for items given the users’ long-term preferences, but to make recommendations given only a few user interactions in an ongoing session [19]. While such scenarios have been addressed in the literature previously, e.g., for web usage prediction [18], they have recently received more attention, e.g., due to the availability of public datasets.

From a technical perspective, almost all session-based algorithms proposed in recent years are based on deep learning (“neural”) architectures. A landmark work in this area is the gru4rec method, which is based on Recurrent Neural Networks (RNNs) [4, 5]. Today, gru4rec is often used as a baseline algorithm in experimental evaluations. However, recent research [7, 15] indicates that simpler methods based on nearest-neighbor techniques can outperform gru4rec in terms of certain accuracy measures. Therefore, when new neural algorithms are published and benchmarked against gru4rec alone, it is not clear whether or not these new methods are actually leading to progress beyond the more simple techniques.

This problem of unclear progress in applied machine learning is not entirely new. In the information retrieval (IR) field, for example, researchers already found in 2009 that the improvements reported over the years “don’t add up” [1]. Recent analyses [10, 16] furthermore indicate that some neural approaches that were recently published at top conferences do not outperform long-established baseline methods, when these are well tuned. The reasons for this non-progress lie in the choice of the baselines used in the experimental evaluations or the limited efforts by the authors to fine-tune the baselines. Sometimes, another problem is the lack of reproducibility of the results. Today, publishing the code of the algorithms is more and more encouraged. However, often the code used for data pre-processing, data splitting, hyper-parameter optimization, and evaluating is not provided. Given that many of these implementation details can affect accuracy, it is often very challenging to make reliable conclusions.

With this work, our goal is to shed light on the progress in the area of session-based recommendation algorithms. We report the results of an in-depth comparison of four recent neural algorithms and a set of mostly simpler baseline algorithms. All algorithms were benchmarked under identical settings within an evaluation framework that we built upon the code from [5]. Our results indicate that the progress that is achieved with neural approaches is sometimes...
very limited, and that well-tuned baselines often outperform even the latest complex models.

Generally, these observations call for improved research practices, as discussed previously in [12]. The availability of an evaluation environment for reproducible research can be one piece of this puzzle. We therefore publicly share our evaluation framework, which includes also code for data splitting, hyper-parameter optimization and a number of additional metrics.

2 BENCHMARKED ALGORITHMS

We have considered the four neural approaches shown in Table 1 in our comparison. We selected them by systematically scanning the proceedings of top-ranked conference series of the last three years. We only included works for which the source code was available and which did not use side information.

<table>
<thead>
<tr>
<th>Table 1: Neural Recommendation Strategies</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRU4REC (ICLR’16, CIKM’18)</td>
</tr>
<tr>
<td>NARM (CIKM’17)</td>
</tr>
<tr>
<td>STAMP (KDD’18)</td>
</tr>
<tr>
<td>NEXTITINET (WSDM’19)</td>
</tr>
</tbody>
</table>

As baselines we use the five techniques that were also used in [15], as well as a recent, more complex approach based on context trees (ct) [17]. All baselines methods shown in Table 2 have the advantage that they can take new interactions immediately into account without retraining, and they only have a small set of parameters to tune. Furthermore, scalability can be ensured for the neighborhood-based techniques through adequate sampling as discussed in [7]. We initially considered additional neural approaches such as [2, 9, 11, 14], but we did not include them in our evaluation for different reasons, e.g., because the source code was not available, or the algorithm also uses side information. We also did not consider sequential approaches like [3, 6, 20], because they are not really designed for session-based scenarios or require user IDs in the datasets.

<table>
<thead>
<tr>
<th>Table 2: Baseline Strategies</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
</tr>
<tr>
<td>SR</td>
</tr>
<tr>
<td>S-KNN</td>
</tr>
<tr>
<td>VS-KNN</td>
</tr>
<tr>
<td>CT</td>
</tr>
</tbody>
</table>

3 DATASETS AND EVALUATION APPROACH

3.1 Datasets

We conducted experiments with seven datasets, four from the e-commerce domain and three from the music domain, see Table 3. Six of these datasets are publicly available. These datasets were also used for the comparison of algorithms in [8, 15] and [13].

<table>
<thead>
<tr>
<th>Table 3: Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
</tr>
<tr>
<td>RSC15</td>
</tr>
<tr>
<td>RETAIL</td>
</tr>
<tr>
<td>DIGI</td>
</tr>
<tr>
<td>ZALANDO</td>
</tr>
<tr>
<td>30MU</td>
</tr>
<tr>
<td>NOWP</td>
</tr>
<tr>
<td>AGTM</td>
</tr>
</tbody>
</table>

Some previous works on session-based recommendation use a single training-test split in their evaluation or very small subsets of the original datasets (e.g., only 1/4 of the RSC15 dataset) [4, 5, 8, 13]. In our work, we followed the approach of [15] and created, for each dataset, five subsets contiguous in time to be able to make multiple measurements in order to minimize the risk of random effects. Table 4 shows the average characteristics of these multiple subsets. Pointers to the resulting datasets and the train-test splits used in the experiments can be found online1, together with the code of our evaluation framework. For all datasets, we removed sessions that contained only one interaction.

<table>
<thead>
<tr>
<th>Table 4: Characteristics of the datasets. The values are averaged over all five splits.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>RSC15</td>
</tr>
<tr>
<td>RETAIL</td>
</tr>
<tr>
<td>DIGI</td>
</tr>
<tr>
<td>ZALANDO</td>
</tr>
<tr>
<td>30MU</td>
</tr>
<tr>
<td>NOWP</td>
</tr>
<tr>
<td>AGTM</td>
</tr>
</tbody>
</table>

3.2 Experimental Procedure

Hyper-Parameter Optimization. We tuned the hyper-parameters for all methods for each dataset systematically, using a subset of the training data—covering the same amount of days as the test set—for validation. As the training process can be time-consuming and the parameter space is large, we applied a random optimization approach with 100 iterations as in [4, 8, 13] (50 iterations for NARM) to find a suitable set of parameters. All models were optimized for the Mean Reciprocal Rank (MRR@20). The ranges and the final values of the hyper-parameters for each dataset can be found online.

Protocol and Metrics. Similar to [4, 5] and other works, we used the last n days of each dataset as test data and the rest for training. For each session in the test data, we incrementally “revealed” one interaction after the other. After each revealed interaction, we

1https://rn5l.github.io/session-rec/index.html
computed recommendation lists and then compared the recommen-
dations with the still hidden elements in the session.

In [5], where gru4rec was proposed, and in subsequent works,
the evaluation procedure is based on measuring to what extent an
algorithm is able to predict the immediate next item in a session.
Their corresponding measurement of the Hit Rate (HR@20) and
the MRR@20 is therefore based on the existence of this next item
in a given top-n recommendation list. In reality, however, usually
more than one item is shown and being able to identify more than
one relevant item for a given session is typically favorable over
just predicting the immediate next one correctly. In this work, we
therefore focus on traditional precision, recall, and mean average
precision (MAP) measures, which consider all items that appear in
the currently hidden part of the session as relevant. As the neural
approaches are not explicitly designed to predict multiple items
and for the sake of completeness, we report both types of measure-
ments.

4 RESULTS

E-Commerce Domain. Table 5 shows the results for the domain
of e-commerce.² On the RETAIL and the DIGI dataset, the nearest
neighbor methods led to the highest accuracy results—averaged
across folds—on all measures. For the ZALANDO dataset, neighbor-
hood methods were again best, except for the MRR. The differences
to the best complex model are in many cases significant.

Only for the RSC15 dataset we can observe that a neural method
(narm) is able to consistently outperform our best baseline vs-knn
on all measures. Interestingly, however, it is one of the earlier neural
methods in this comparison. The results for the RSC15 dataset are
generally different from the other results. The ct method, for exam-
ple, was very competitive on the MRR for this dataset. Stamp, while
being a very recent method, was not among the top performers
except for this dataset. Given these observations, it seems that the
RSC15 dataset has some unique characteristics that are different
from the other e-commerce datasets.

For the larger ZALANDO and RSC15 datasets, we do not include
measurements for the most recent nextitnet method. We found
that the method does not scale well and we could not complete the
hyper-parameter tuning process within weeks on our machines
(also for two music datasets).

Music Domain. Table 6 shows the results for the music domain.
The results are mostly aligned with the e-commerce results. On
all datasets, the nearest-neighbor methods outperform other all
techniques on precision, recall, MAP, and the hit rate. In terms of
the MRR measure, the non-neural CT method consistently leads to
the highest values. The simple SR method is again competitive in
terms of the MRR, and gru4rec as well as narm are again among
the top-performing neural approaches. The neighborhood methods
in all cases are not in the leading positions in terms of the MRR
and even lead to the lowest MRR performance on the AOTM dataset.
The Stamp method can consistently be found at the lower ranks in this
comparison.

²The highest value across all techniques is printed in bold; the highest value obtained
by the other family of algorithms—baseline or complex model—is underlined. Stars in-
dicate significant differences according to a Student’s t-test with Bonferroni correction
between the best-performing techniques from each category. *: p<0.05, **: p<0.01.
Table 6: Results for the music domain datasets

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MAP@20</th>
<th>P@20</th>
<th>R@20</th>
<th>HR@20</th>
<th>MRR@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOWP</td>
<td>0.0139</td>
<td>0.0209</td>
<td>0.0309</td>
<td>0.0417</td>
<td>0.0545</td>
</tr>
<tr>
<td>VS-KNN</td>
<td>0.0673</td>
<td>0.0845</td>
<td>0.0935</td>
<td>0.1034</td>
<td>0.1206</td>
</tr>
<tr>
<td>S-KNN</td>
<td>0.0986</td>
<td>0.1086</td>
<td>0.1186</td>
<td>0.1286</td>
<td>0.1386</td>
</tr>
<tr>
<td>AR</td>
<td>0.1176</td>
<td>0.1276</td>
<td>0.1376</td>
<td>0.1476</td>
<td>0.1576</td>
</tr>
<tr>
<td>SR</td>
<td>0.1266</td>
<td>0.1366</td>
<td>0.1466</td>
<td>0.1566</td>
<td>0.1666</td>
</tr>
<tr>
<td>NARM</td>
<td>0.1356</td>
<td>0.1456</td>
<td>0.1556</td>
<td>0.1656</td>
<td>0.1756</td>
</tr>
<tr>
<td>GRU4REC</td>
<td>0.1446</td>
<td>0.1546</td>
<td>0.1646</td>
<td>0.1746</td>
<td>0.1846</td>
</tr>
<tr>
<td>STAMP</td>
<td>0.1536</td>
<td>0.1636</td>
<td>0.1736</td>
<td>0.1836</td>
<td>0.1936</td>
</tr>
<tr>
<td>CT</td>
<td>0.1626</td>
<td>0.1726</td>
<td>0.1826</td>
<td>0.1926</td>
<td>0.2026</td>
</tr>
</tbody>
</table>

Table 7: Running times

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Training RSC15</th>
<th>Predicting (ms) ZALANDO</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRU4REC (on GPU)</td>
<td>0.05h</td>
<td>1.51h</td>
</tr>
<tr>
<td>STAMP (on GPU)</td>
<td>1.25h</td>
<td>7.61h</td>
</tr>
<tr>
<td>NARM (on GPU)</td>
<td>4.36h</td>
<td>12.99h</td>
</tr>
<tr>
<td>NEXTITNET (on GPU)</td>
<td>26.96h</td>
<td>7.16h</td>
</tr>
</tbody>
</table>

3Times were measured on a workstation computer with an Intel Core i7-4790k processor and a Nvidia GeForce GTX 1080 Ti graphics card (Cuda 10.1/CuDNN 7.5).

it seems that progress in neural session-based recommendation is still limited, and the various reported improvements over the landmark GRU4REC method are seemingly not enough to consistently outperform much simpler techniques.

4.1 Additional Observations

Scalability. Scalability can be an issue for some of the complex models, with GRU4REC being among the faster approaches. The authors of STAMP and NARM, for example, use only 1/4 or 1/4 of the RSC15 dataset in their own experiments. Similarly, the largest dataset used for the evaluation of NEXTITNET has about 2 million sessions, which is a fraction of the original RSC15 dataset.

We measured the runtimes of training and prediction for all methods in all experiments. As an example, we report the results for RSC15 and ZALANDO in terms of the training time for one split and the average time needed to generate a recommendation list.

Methods like SR or VS-KNN do not learn complex models. They only need some time to count co-occurrences or prepare data structures. Also, the CT technique can be efficiently initialized. Training GRU4REC on one data split on our hardware took less than an hour. STAMP needed only slightly more time than GRU4REC, but NARM was four times slower. Finally, the most recent convolutional NEXTITNET method seems to be limited in terms of practical applicability as it needs more than one day for training on a GPU even for datasets of modest size. When datasets are used that comprise a larger set of items, e.g., the one from Zalando, the performance differences are even more pronounced.

coverage and Popularity Bias. Previous work has indicated that some methods, in particular the simpler ones, can have a tendency to recommend more popular items [15]. At the same time, some algorithms can focus their recommendations on a small set of items that are recommended to everyone, which can be undesired in certain domains and lead to limited personalization.

To identify such potential differences, we measured the popularity bias of each algorithm by averaging the min-max normalized popularity values of the recommended items in the top-20 recommendations. Furthermore, we determined the fraction of items that ever appeared in the generated top-20 recommendations (coverage).

The general tendencies across datasets are as follows. In terms of the popularity bias, CT is usually very different from the other methods, and it focuses much more on popular items. For the other methods, no clear ranking was found across datasets. In many cases, however, GRU4REC is among the methods that recommend the least popular (or: most novel) items. GRU4REC also often has the highest and STAMP the lowest coverage. VS-KNN is similar to the other neural approaches in terms of coverage.

5 CONCLUSIONS

Our work indicates that even though a number of papers on session-based recommendations were published at very competitive conferences in the last years, progress seems to be still limited (or only phantom progress) despite the increasing computational complexity of the models. Similar to the IR domain, one main problem seems to lie in the choice of the baselines, and our work points to a potentially major limitation of today’s research practice.

A general phenomenon in that context is that previous non-neural approaches—as well as simpler methods—are often disregarded in empirical evaluations, and only neural methods are used as baselines despite their possibly unclear competitiveness.

In some papers, little is also said about hyper-parameter optimization for the baselines. In addition, the code which is used in the optimization and evaluation procedures is not always shared, making reproducibility an issue. With our work, we provide a framework based on the work from [5, 15], where various algorithms can be benchmarked under the exact same conditions, using different evaluation schemes. Overall, we hope that this environment is helpful for other researchers to achieve higher levels of reproducibility and faster progress in this area.
REFERENCES


