"Informed dataset selection with APS" - Reproduced

Abdelrahman Al-Taslaq
abdelrahman.altaslaq@student.uni-siegen.de
University of Siegen
Siegen, Germany

ABSTRACT

This paper presents a reproduction and validation of the experiment described in "Informed Dataset Selection with 'Algorithm Performance Spaces'" by Joeran Beel et al. (2024)[2]. The study evaluates 15 diverse recommendation algorithms—including LR, NFM, AFM, DCN, DeepFM, DSSM, FiGNN, AutoInt, EulerNet, FM, FNN, FwFM, PNN, Wide Deep, and xDeepFM—across 17 datasets sourced from Kaggle [4]. These datasets span a wide range of domains, including product reviews and ratings for items such as iPhones, airlines, movies, books, clothing, and hotels among others. The experiments demonstrate that random dataset selection does not provide meaningful insight into algorithm performance. In contrast, leveraging the Algorithm Performance Space (APS) allows for a more informed and structured evaluation process.

KEYWORDS

APS, Diverse, Performance

1 INTRODUCTION

1.1 Background

In recommender systems research, the choice of datasets used for experimentation is a foundational aspect of model development and evaluation. Recommender algorithms are designed to predict user preferences based on patterns in historical data, and the nature of the dataset directly influences how those patterns are learned, how well they generalize, and how model performance is interpreted. A model that performs well on one dataset may fail on another, due to differences in user behavior, item diversity, sparsity levels, domain specificity, and other underlying data characteristics. This makes the selection of datasets not just a technical detail, but a critical factor that can significantly shape the outcomes of experimental research. As recommender systems continue to expand into a wider range of domains and are used by increasingly diverse user groups with varying preferences and behaviors, the importance of selecting datasets thoughtfully has been greater. There is a growing need for principled, transparent, and performance-aware methods of dataset selection that go beyond simply picking what is convenient or traditionally used. Such methods should be designed to capture the complexity and variability of real-world environments where these systems operate. In practice, however, the process of choosing datasets is often informal or based on convention. Researchers frequently use datasets that are publicly available, well-known, or commonly referenced in prior work. Datasets like MovieLens and Amazon product reviews are widely used benchmarks in the recommender systems community. These are often selected because they

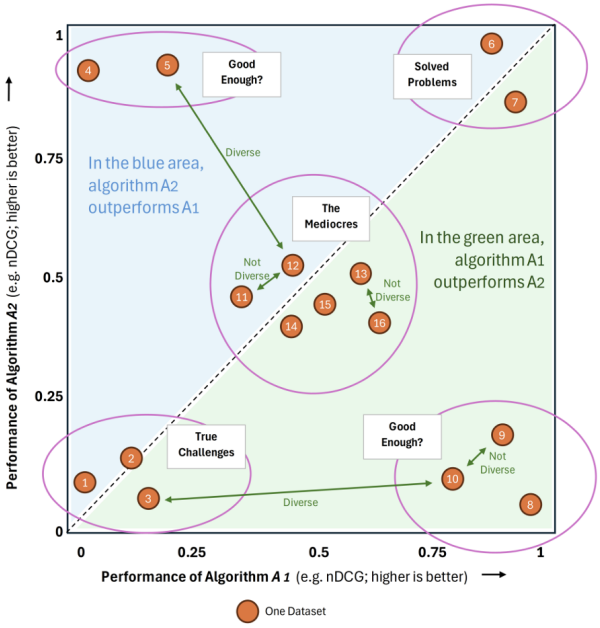

Figure 1: Illustration of an APS diagram [2]

are typically marked as "standard" or "public," and are selected without a deeper examination of whether they offer sufficient variety or challenge to properly test and differentiate algorithms[2].

1.2 Research Problem

Choosing datasets mainly because they are popular, easy to access, or widely cited creates several challenges in recommender system research. When researchers use well-known benchmarks like MovieLens or Amazon reviews simply out of popularity or convenience, evaluations often fail to capture the full diversity of real-world recommendation problems. These datasets might share similar properties—such as specific domains, user demographics, or data sparsity—that may favor some algorithms while disadvantaging others. Relying on popular datasets can lead to results that don't generalize well beyond these familiar settings; an algorithm tuned on movie data may not perform as well in retail or social media contexts. Moreover, many popular datasets don't provide enough variety or difficulty to clearly differentiate between algorithms, making it harder to identify true improvements and slowing research progress.

To illustrate, if all algorithms perform similarly on a given dataset, it becomes hard to tell their individual strengths and weaknesses

or to identify which methods work better for specific problems. Additionally, datasets that are extremely sparse or contain a lot of noise can disproportionately affect some algorithms, causing results to be skewed and not accurately reflecting how well the algorithms would perform in real-world settings. This can mask important differences in algorithm effectiveness and limit the usefulness of the evaluation. Therefore, Careful consideration of dataset selection is essential when evaluating algorithms.

1.3 Original Work

What the authors did:

To address the problem of choosing datasets for recommender systems researches, the authors proposed the method Algorithm Performance Space (APS)—a framework to support more thoughtful, performance-based dataset selection.[2]

APS visualizes datasets as points in an n-dimensional space, with each dimension measuring the performance of a specific recommendation algorithm based on the nDCG metric. The distance between datasets reflects how differently algorithms perform on them. This idea redefines dataset diversity not by metadata (e.g., domain, sparsity) but by algorithmic behavior. In addition to that, APS classifies the datasets according their solvability.[2]

To operationalize APS, the authors ran 29 algorithms across 95 datasets using RecBole. This generated a high-dimensional space, which they further explored through 812 mini-APS, each being a 2D subspace based on a pair of algorithms. They also applied Principal Component Analysis (PCA) to create a 2D projection summarizing the overall space. Through this, they produced visual tools that illustrate how datasets relate based on algorithmic performance similarities or differences.[2]

Main findings:

They confirmed that dataset choice in recommender systems is frequently based on convenience or popularity rather than meaningful metrics. One striking observation was that Amazon datasets, despite being widely used and appearing diverse based on metadata in prior studies, tend to be highly similar in APS—suggesting algorithms perform similarly on them. This calls into question their utility for evaluating generalization or distinguishing algorithmic strengths.

In contrast, MovieLens datasets, often seen as “standard” or even limited, showed more diversity in algorithmic performance, thus making them more valuable than previously thought. The authors also noted clusters in APS: datasets in the top-right performed well across all algorithms—labeled as “solved problems”—while those in the bottom-left were uniformly difficult, offering opportunities for further research.

Interestingly, many datasets clustered closely in APS, meaning algorithms tended to succeed or fail consistently across them. This clustering emphasizes the challenge of finding truly diverse datasets. The PCA-reduced APS revealed that many less popular datasets (e.g., FilmTrust, Docear, KGRec-Music) appear far apart, highlighting them as promising candidates for novel research.

Conclusion:

APS provides a transparent, performance-oriented framework for selecting datasets, enabling researchers to justify their choices based on actual algorithm behavior rather than availability or tradition. It encourages the identification of datasets that offer real algorithmic challenges and promotes generalizable evaluation. Although the APS framework is still evolving, it represents a significant shift from arbitrary dataset use to evidence-driven experimentation.

The authors envision APS becoming a community resource potentially expandable and customizable—where researchers could map new datasets or algorithms and visualize their positions. They emphasize that while APS should not rigidly dictate dataset choice, it enables a new level of rigor and clarity in experimental design. Open questions remain, such as how best to calculate distances in high dimensions, whether to standardize APS globally, and how to balance comprehensiveness with usability.

Ultimately, APS is a powerful conceptual and practical tool that may help the recommender systems community move toward better reproducibility, fairness, and scientific progress.

1.4 Research Goal

To prove the assumptions made by the authors in the original paper [2], their work should be reproduced using different parameter,algorithms,datasets and/or metrics. This paper shows the reproduction results of using different algorithms and datasets than which were used in the original paper. The assumption should hold if it was clear from the results that choosing diverse datasets leads to a better ranking and evaluation of the algorithms than just choosing randomly or choosing datasets that are close to each other in the APS diagrams.

2 METHODOLOGY

My experiment aims to reproduce and validate the findings of the paper "Informed Dataset Selection with Algorithm Performance Spaces" by implementing a similar framework on a new set of real-world datasets. The work was carried out using Python 3.11 on a personal ASUS laptop equipped with an AMD Ryzen 5000 series CPU, and the entire coding environment was managed through Visual Studio Code. The source code, configurations, and dataset loader files for this project are available in a GitHub repository, referenced as [3].

To remain as faithful as possible to the original methodology, the same pipeline structure was followed. However, a significant change was made to the dataset filtering step. The original paper used a 5-core filter, which retains only users and items with at least five interactions. In this experiment, a 1-core filter was used instead. This decision was made based on the nature of the datasets being processed—many were relatively small or lacked dense interaction histories. Applying a 5-core filter would have excluded large portions of the data, so the 1-core threshold was chosen to retain more information while still meeting a minimum level of interaction integrity.

This reproduction did not include any of the datasets from the original work; instead, all datasets were newly sourced from Kaggle[4], covering a broad range of domains, user behaviors, and

product categories. Initially, 32 datasets were selected. These included product reviews and ratings for different items: a comparison of Adidas vs. Nike reviews, wine reviews, two separate airline review datasets, two datasets carrying different reviews about various products, one hotel reviews dataset, an anime rating dataset, and reviews covering books, clothing, Disneyland, Starbucks, Ryanair, laptops, video games,iPhone and rottentomato movies rating dataset.

Each dataset was manually inspected, and the column names were modified to match the formatting requirements for the recommender system pipeline, the underlying data values remained unchanged. These requirements included the presence of a user ID column, an item ID column, a rating column, and if available, a timestamp column and metadata features. Following this review, 15 datasets were excluded from the project due to issues such as missing critical columns, invalid data types, or extensive gaps in the data that made them unusable. As a result, 17 datasets were used in the final experiment. For each dataset, a dedicated data loader was written, and the dataset was stored in its own directory, containing the raw .csv file and all intermediate and final outputs.

To perform the algorithmic experiments, the project utilized the RecBole library—a comprehensive Python framework for developing and benchmarking recommender systems. The algorithms used in this reproduction did not include any of the algorithms from the original work. A total of 15 algorithms were evaluated: Logistic Regression (LR), Neural Factorization Machine (NFM), Attentional Factorization Machine (AFM), Deep Cross Network (DCN), DeepFM, Deep Structured Semantic Model (DSSM),Feature-specific Interpretable Graph Neural Network (FiGNN), Automatic Feature Interaction Learning (AutoInt), EulerNet, Factorization Machine (FM), Feedforward Neural Network (FNN), Field-weighted Factorization Machine (FwFM), Product-based Neural Network (PNN), Wide Deep, and xDeepFM.

The pipeline for processing and training followed several sequential steps for each dataset. The first stage was fitting, in which the raw dataset was preprocessed and transformed into a structured format required by RecBole. The processed data was saved back into the dataset's directory. Following this, the data went through an atomic transformation, which further refines the data structure into the specific input format used by RecBole for training and evaluation.

Once preprocessing was completed, the next phase involved training and evaluation. Each dataset was split into five folds to perform cross-validation. For each fold, every algorithm was trained for 50 epochs. After training, the models were used to generate predictions, and these predictions were evaluated using NDCG@10 (Normalized Discounted Cumulative Gain at rank 10) among others: NDCG@1, HR@1, Recall@1, NDCG@3, HR@3, Recall@3, NDCG@5, HR@5, Recall@5, HR@10, Recall@10, NDCG@20, HR@20, Recall@20.

At the end of the process, each dataset folder contained multiple JSON result files summarizing the different stages of the pipeline. These included a fit results json file for the fitting process, a prediction results json file for the model's prediction outputs, and an evaluation results json file containing the evaluation metrics for each fold and each algorithm.

Once all evaluations were complete, the results were compiled into a consolidated file named merged.csv, which listed performance metrics across all datasets and algorithms. This merged dataset was then used to generate Algorithm Performance Space (APS) diagrams based on the metric NDCG@10. APS plots are two-dimensional visualizations in which each point represents a dataset, and its position reflects the performance of two algorithms. Since there were 15 algorithms, 210 unique APS plots were generated, each comparing a pair of algorithms. These diagrams helped visualize how similarly or differently datasets behaved when evaluated by different models.

In each APS plot, datasets were represented as points and were randomly colored to distinguish them visually. Datasets that were not evaluated (for example, due to an error during training or evaluation, or having zero NDCG) were excluded from the corresponding plots. These APS diagrams make it easy to spot patterns—for instance, datasets that cluster closely suggest similar algorithmic behavior and therefore less diversity, while datasets that are widely spread indicate more varied behavior and higher diversity.

In addition to the APS plots, a Principal Component Analysis (PCA) was applied to the full algorithm-dataset performance matrix to reduce it to two dimensions. This provided a single, more interpretable plot showing how all datasets are distributed based on the variance in algorithm performance. As in the original paper, it was emphasized that the PCA axes do not directly reflect performance metrics but are instead linear combinations that capture the most variance. The resulting PCA plot also featured randomly colored datasets for easier interpretation.

The experiment successfully reproduced the key ideas of the APS paper, adapting them to available datasets and computing resources. Results show that APS is a useful tool for analyzing dataset diversity and guiding dataset selection in recommender system research. Some datasets yield uniform algorithm performance, offering limited value, while others show diverse results, making them more useful for benchmarking and development.

3 RESULTS & DISCUSSION

Results The result was 210 2D-APS diagrams, in addition to a PCA diagram. The goal of this experiment was to prove the assumptions made by the original authors, in the following we will prove the main assumption, according to some diagrams form the results.

APS

Main assumption in the original paper:

> *"The rationale behind the APS is as follows. If some datasets D1...Dm are close to each other in the APS, this indicates that all al gorithms A1...An in the APS have performed similarly on them.To clarify, this does not imply that all n algorithms achieved the same performance on the m datasets. It could be, for instance, that algo rithms A1, A2 and A3 performed well on the m datasets; algorithms A4 and A5 performed poorly on the m datasets and algorithms A6...An exhibited mediocre performance on the m datasets. In other words, algorithm A1 performed consistently across the m datasets, A2 performed consistently across the m datasets (but not neces sarily similarly to A1) and all other n algorithms also*

performed consistently across the m datasets. Therefore, it seems likely to us that a novel algorithm An+1– that is not part of the APS– will also perform consistently across the m datasets (whether performance will be high, low or mediocre cannot be predicted). If, for instance, algorithm An+1 performs poorly on one or two of the n datasets, we consider it extremely likely that algorithm An+1 will perform similarly on the remaining n datasets because this behavior was true for all n algorithms. Consequently, evaluating the novel algorithm on one or two of the n datasets would be sufficient" ... "Based on the above rationale, we argue that researchers typically should choose datasets with high diversity, i.e., datasets that are highly distant from each other in the APS. This approach allows researchers to determine whethertheiralgorithmisan"all-rounder" that performs well across various scenarios or excels only in specific areas of the APS"[2]

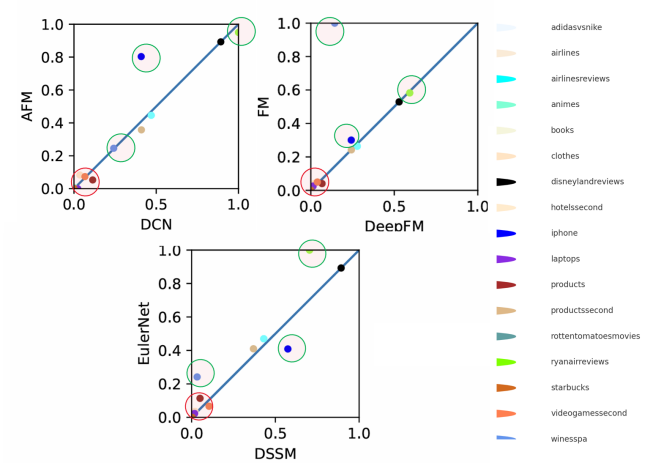

Figure 2

To validate the assumption, one can observe how a group of algorithms behaves across a set of datasets by examining their APS diagrams. If multiple algorithms consistently perform similarly—indicated by tight clustering of datasets—then introducing a new algorithm and seeing the same clustering pattern suggests it behaves similarly too. This would prove the assumption and supports the idea that evaluating one or two algorithms on just a few representative clustered datasets may be sufficient to infer the performance on the other algorithms.

So by looking at the datasets marked within a red circle in the diagrams in figure 2, we notice that the assumption holds. These datasets are close to each other, hence not diverse. If we look at how the 6 algorithms AFM, DCN, FM, DeepFM, EulerNet, DSSM we notice that all these algorithms performed consistently and similar on the n datasets, marked within the red circle. According to the assumption, a new algorithm should now also perform consistently and similar, which we notice by looking at new 2 algorithms FwFM and xDeepFM in figure 3. They performed similarly to the other 6 algorithms. That means just by looking at one or two algorithms and their performance to n close datasets (not diverse), we would be able to expect the evaluation of the rest of all the other algorithms, if the first two algorithms performed similarly.

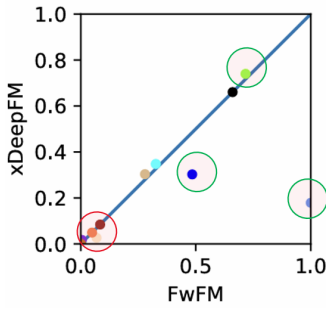

Figure 3

We now see that we can not evaluate the algorithms based on random selection of the datasets, and for sure we should not choose datasets that are close to each other in the APS diagrams, as this will not help evaluate the algorithms fairly, or rank them or understand them, since they will all perform similarly. Instead, we should pick diverse datasets, datasets that are plotted far from each other in the APS diagrams, where the same algorithm performs differently on them, well on some, and poorly on the others. We focus now on the datasets that are marked within green circles in figure 2 and 3, we notice that these three datasets are diverse, and algorithms perform differently on them. Studying the evaluation on these algorithms will deliver us some clear ranking of which algorithm is better than which, for example, algorithm AFM showed the best performance overall, since it showed very good performance values on two of the three datasets, then comes algorithm FwFM, with almost as high performance as in AFM, on two of the three datasets. In this case, choosing diverse datasets helped us pick a better algorithm than another. Aps can also help us have a better understanding of the algorithms, where we can see on which datasets the algorithm performed well, and have an idea where this algorithm might be suitable the best, and on which kind of data.

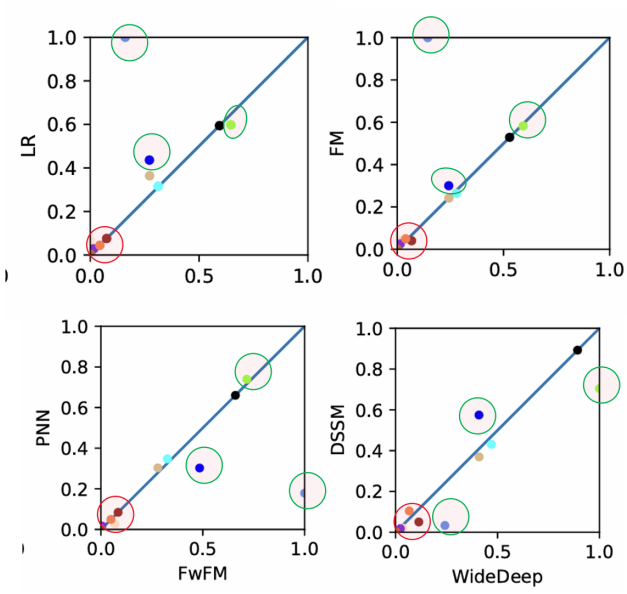

Figure 4: Different APS diagrams from the results

PCA

Figure 5 plot shows a 2D projection of the Algorithm Performance Space (APS) using Principal Component Analysis (PCA). PCA is a dimensionality reduction technique that transforms high-dimensional data (in this case, algorithm performances across 210 APS diagrams) into a lower-dimensional space while preserving as much variance (information) as possible. Each point in the plot represents a dataset, positioned based on how algorithms perform on it.The horizontal axis (Component 1) explains 86.75% of the variance in the data, and the vertical axis (Component 2) explains another 12.31%. Together, they give a good overview of how datasets relate to each other in terms of algorithm performance. Clusters on the left side (with no labels) indicate datasets on which algorithms behave similarly, suggesting low diversity in performance. This includes many tightly grouped colored dots. The most diverse datasets were highlighted —those that are far from the main cluster—by labeling them. These include: disneylandreviews, ryanairreviews, iphone and airlinesreviews. These datasets are farther from the cluster, which means that algorithms behave differently on them compared to the majority. This diversity is what makes them interesting and valuable for further analysis. The axes don't represent actual performance (e.g., accuracy or F1 score), but rather patterns of performance variation across datasets. So being in the top-right corner doesn't mean a dataset leads to high performance — just that it's performance behavior is distinct. This PCA supports the idea that testing on a few diverse datasets may be enough to infer an algorithm's general behavior, as it shows that while many datasets cluster together (suggesting redundancy), a few datasets behave quite differently with respect to algorithm performance. By focusing on these diverse datasets, we can maximize the information gain when evaluating new algorithms, aligning with the core assumption of the original study.

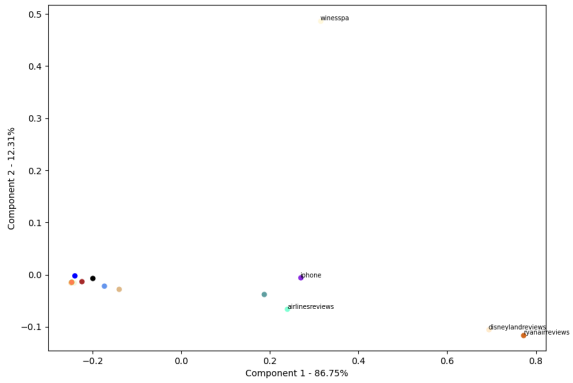

Figure 5: Principle Component Analysis(PCA)

Discussion

The use of Algorithm Performance Space (APS) analysis provided a principled framework for identifying which datasets should be prioritized when evaluating algorithm performance. Specifically, it enabled us to distinguish datasets that exhibit high diversity in algorithm behavior—those which are likely to be more informative and discriminative during evaluation. Rather than selecting datasets arbitrarily or relying solely on popularity or historical usage (e.g., frequently cited or widely adopted benchmarks), APS offers a data-driven rationale for selection. Popularity alone does not necessarily imply that a dataset is representative or suitable for robust algorithm comparison. In fact, selecting datasets without justifiable reasoning can lead to biased or incomplete evaluations.

This is not to suggest that random selection of datasets will always invariably produce flawed evaluations. In some cases, randomly chosen datasets may still yield useful insights, particularly if the algorithm is highly consistent in its performance. However, APS presents a more reliable and systematic alternative by allowing researchers to visualize and quantify performance patterns across datasets. This increases confidence that the selected datasets will meaningfully differentiate algorithm capabilities.

Nevertheless, certain experimental variables remain that could potentially affect the generalizability of our results. For instance, one may ask whether similar patterns in APS would emerge if a different evaluation metric were used—such as switching from NDCG to MAP or RMSE. Likewise, it is worth questioning whether variations in other experimental parameters, such as model architecture, hyperparameter settings, or training regimes (e.g., number of epochs), would lead to significant changes in the resulting APS structure. These open questions suggest avenues for further investigation, particularly to test the robustness and reproducibility of insights derived from APS-guided evaluation strategies.

4 LIMITATIONS

All parameters and metrics used in the original paper match those used in this reproduction, except for the parameter core, in this experiment, a 1-core filter was used instead of 5-core. Algorithms

and datasets are also different than the ones used in the original paper.

5 ACKNOWLEDGMENTS

This work was conducted as part of the Machine Learning Praktikum at the University of Siegen [1].

REFERENCES

[1] Joeran Beel and Lukas Wegmeth. 2025. Machine Learning Praktikum. *Universisät Siegen* (2025).
[2] Joeran Beel, Lukas Wegmeth, Lien Michiels, and Steffen Schulz. 2024. Informed Dataset Selection with 'Algorithm Performance Spaces'. *18th ACM Conference on Recommender Systems (RecSys '24), October 14–18, 2024, Bari, Italy* (2024).
[3] github :. https://github.com/taslaq2001/Informed-Dataset-Selection-with-APS/tree/main/.
[4] Kaggle :. https://www.kaggle.com/.