Exploratory Methods for Evaluating Recommender Systems

Joey De Pauw
joey.depauw@uantwerpen.be
University of Antwerp

ABSTRACT

A common and recently widely accepted problem in the field of machine learning is the black box nature of many algorithms. In practice, many machine learning algorithms can only be viewed and evaluated in terms of their inputs and outputs, without taking their internal workings into account. Perhaps the most notorious examples in this context are artificial neural networks and deep learning techniques, but they are certainly not the only techniques that suffer from this problem. Matrix factorisation models for recommendation systems, for example, suffer from the same lack of interpretability. Our research focuses on applying and adapting pattern mining techniques to gain meaningful insights in recommendation algorithms by analysing them in terms of both their input and output, also allowing us to compare different algorithms and discover the hidden biases that lead to those differences.

CCS CONCEPTS

• **Information systems → Evaluation of retrieval results**; **Recommender systems**; *Data mining*.

KEYWORDS

Interpretability; Model Explanation; Pattern Mining

ACM Reference Format:
Joey De Pauw. 2020. Exploratory Methods for Evaluating Recommender Systems. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20), September 22–26, 2020, Virtual Event, Brazil.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3383313.3411456

1 INTRODUCTION

Explaining the results of machine learning algorithms has become an important topic in the area of Artificial Intelligence. Most of this research, however, focuses on the explanation of individual actions, such as classifications, or recommendations, answering questions such as: "Why was I recommended this item?", or "Why was that person classified in that class?". Although these are very important questions indeed, we argue that the ability to explain the working of the entire model is of crucial importance to its users as well, answering questions such as: "For what subsets of my data does this method work best?".

Today, in most scientific literature, different machine learning models are being compared to each other by evaluating their performance on a selected collection of historic datasets, for a selected set of metrics. The results of such experiments are typically reported in a table such as the illustrative example in Figure 1, where the best scores are emphasised in bold and the winning model is the one that outperforms the other models for most of the datasets and metrics.

model-variants	(a) *ML-20M* Recall @20	Recall @50	NDCG @100	(b) *Netflix* Recall @20	Recall @50	NDCG @100	(c) *MSD* Recall @20	Recall @50	NDCG @100
ADMM ≥ 0	0.376	0.502	0.407	0.352	0.435	0.384	0.330	0.425	0.388
ADMM L1	0.391	0.521	0.420	0.362	0.445	0.394	**0.334**	0.428	0.390
ADMM ≥ 0 & L1	0.376	0.502	0.407	0.352	0.435	0.384	0.330	0.425	0.388
Dense	0.391	0.521	0.420	0.362	0.445	0.393	**0.333**	0.428	0.389
Dense ≥ 0	0.373	0.499	0.402	0.345	0.429	0.377	0.324	0.418	0.379
Sparse Approx.	0.391	0.521	0.420	0.361	0.445	0.393	**0.333**	0.427	0.389
Sparse Approx. ≥ 0	0.373	0.499	0.402	0.344	0.428	0.376	0.324	0.417	0.377
Centered	0.391	0.521	0.421	0.363	0.446	0.394	**0.333**	0.428	0.389
Centered & item-L2	0.391	0.522	0.422	**0.365**	**0.448**	**0.397**	**0.334**	**0.429**	**0.391**
Centered & item-L2 & L1	0.391	0.522	0.422	**0.365**	**0.449**	**0.398**	**0.334**	**0.430**	**0.392**
	Results reproduced from [14]:								
SLIM	0.370	0.495	0.401	0.347	0.428	0.379	– did not finish in [14] –		
WMF	0.360	0.498	0.386	0.316	0.404	0.351	0.211	0.312	0.257
CDAE	0.391	0.523	0.418	0.343	0.428	0.376	0.188	0.283	0.237
MULT-VAE^{PR}	**0.395**	**0.537**	**0.426**	0.351	0.444	0.386	0.266	0.364	0.316
MULT-DAE	0.387	0.524	0.419	0.344	0.438	0.380	0.266	0.363	0.313

Figure 1: Example of an experimental comparison between methods from a study in 2020 [23]©. Values have been blurred because they are not important to illustrate the common practice of reporting results, where performance is evaluated on a selected collection of datasets, for a selected set of metrics. Although general arguments are useful to explain the overall trend, often the relation between individual results is not known.

Although general arguments try to motivate the superiority of one algorithm over another, very little is typically known about the underlying reasons. More importantly, when multiple models are deployed in practice, it is rarely the case that one model is consistently the best. Different authors often report high levels of accuracy that are only possible under strict assumptions; while in real-world applications, where cold-start, noise and concept drift [28] are common, the accuracy is far lower and less stable [13, 20, 22]. Therefore, we address the need for exploratory methods to evaluate and analyse different machine learning models and answer questions such as:

- why does model A outperform model B for metric M;
- why did model A outperform model B on the data of day D_1 and not on the data of day D_2;
- for what subset of the data does model B outperform model A?

Possible answers to these questions could be: "for all data that satisfies formula f, model A consistently outperforms model B", or "on day D_1, most instances were of type t_1, and on day D_2, most instances were of type t_2, suggesting that model A is better suited for data of type t_1".

We aim to study and develop techniques that can find the most suitable formulas and data type descriptions in order to produce such answers. Our focus will primarily go to pattern mining and

RecSys '20, September 22–26, 2020, Virtual Event, Brazil

ACM ISBN 978-1-4503-7583-2/20/09.
https://doi.org/10.1145/3383313.3411456

subgroup discovery techniques, where patterns can be association rules over itemsets, episodes or other more expressive types of patterns [1]. We will also study techniques for analysing concept drift [28], and their relation to the performance of different models.

2 RELATED WORK

This research can be situated in several active areas of research: *pattern mining*, *recommender systems*, *interpretability of machine learning models*, and *meta-learning*. The following sections describe the state of the art and current challenges in each of these fields with respect to our topic of research.

2.1 Pattern Mining

The goal of this research is not to advance the state of the art in pattern mining, but rather to investigate domain specific synergies with respect to recommender systems data. A known issue with pattern mining is the *pattern explosion problem* and the difficulty to reduce the potentially huge set of reported patterns to the most interesting subset. The idea is that finding a smaller set of interesting patterns becomes easier and more accurate given a specific domain, use case or target end-user. Consider an example where we concatenate for each user their history and the set of recommended items. The fact that every item can occur as a recommendation as well as in the history, cannot trivially be represented in the traditional pattern mining setting, whereas domain specific solutions are able to exploit this information.

Finding generic, objective interestingness measures to filter or sort patterns has already been studied extensively. These measures mostly focus on redundancy [3, 26] or unexpectedness [27], which already helps to reduce the set of reported patterns [9]. Finding truly interesting patterns however, remains inherently subjective and dependent on the domain. To the best of our knowledge, no substantial amount of work has been done in the context of applying pattern mining techniques on recommender system data for the purpose of interpretability or for model selection.

2.2 Recommender Systems

The evaluation of recommender systems is an active area of research, with recent studies [8, 13, 19] showing that offline metrics give poor estimates of the online performance. Furthermore, it is widely accepted that performance should not only be assessed in terms of accuracy, but by a wider variety of metrics, like diversity, novelty, coverage and serendipity [13, 14].

In a positioning paper from 2017 [2], Beel argues that, in the context of recommender systems, single numbers express only average effectiveness over usually a rather long period, which only provides a vague and static view of the data. He then proposes that recommender system researchers should instead calculate metrics for time-series such as weeks or months. This way, results show how an algorithm's effectiveness develops over time [2].

This idea is illustrated in Figure 2, where we visualize how, in real life, it is rarely the case that one recommender system is consistently the best. Typically, recommenders are evaluated on a static snapshot of the data, where often, the time when the snapshot was taken, can change the outcome and conclusions of the experiments. To make matters worse, the ordering of interactions is often neglected

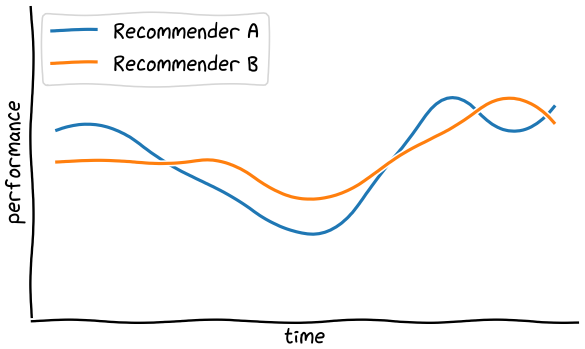

Figure 2: Illustration of comparing two recommenders over time in an A/B test. The conclusion of which algorithm performs best depends heavily on at which point in time a snapshot is taken.

in offline evaluation, effectively enabling algorithms to predict the past based on future events [10]. We can even extend this problem setting beyond the notion of time, by allowing a partitioning in subsets based on arbitrary criteria, in the hope to dig deeper and explain what actually causes the effectiveness of algorithms to differ.

Previous work with respect to the interpretability of recommender systems has mostly focused on prediction explanation, like for example counterfactual explanation [17]. However, very few studies talk about explanation or interpretation of the recommender system itself or its results. Questions like "Under which conditions is item A consumed most often?" or "What do users who are recommended item A have in common?" could provide meaningful insights for the development and tuning of algorithms, but the automated answering of these questions has not received much attention from researchers.

A preliminary study by Moens et al. [15] has demonstrated that pattern mining may be a feasible approach for answering these types of questions. Furthermore, due to the explorative nature of pattern mining, the technique may even help uncover answers to important questions that the developer didn't know to ask (yet).

2.3 Interpretability of Machine Learning Models

There is an active research trend towards post-hoc, model-agnostic interpretability tools [16], with recent developments such as permutation feature importance [7], SHAP [12] and LIME [18] for explaining individual actions (local interpretability). Our approach on the other hand can be classified as a post-hoc method for *global* interpretability. Compared to the various advancements in local interpretability, methods for global model interpretability remain limited, making it difficult to achieve in practice. Options for global interpretability include training an explainable global surrogate model (if one exists at the given scale and complexity [11]) or resorting to a modular level: explaining parts of the model in isolation [16].

2.4 Meta-learning

Meta-learning studies how learning systems can increase in efficiency through experience [25]. One type of meta-knowledge on

a learning algorithm is its *bias* on the set of possible hypotheses explaining a concept [24]. This is a natural consequence of the no free lunch theorem, which states that, if an algorithm performs well on a certain class of problems then it necessarily pays for that with degraded performance on the set of all remaining problems [29].

One of the goals in meta-learning is to learn the bias of a learning algorithm L and investigate what causes a learning algorithm to dominate in a specific region (a set of similar tasks). The problem can be decomposed in two parts: 1) determine the properties of the task in the region that make L suitable for such region; and 2) determine the properties of L (i.e., what are the components contained by algorithm L and how they interact with each other) that contribute to the domination in the region. [25]

Our research topic has significant overlap with this specific goal in the broader field of meta-learning. Also closely related are: 1) the field of concept drift, where we study unexpected changes of data distributions over time and the impact this has on learning algorithms [28, 30]; and 2) the area of ensemble models, where multiple models are being combined in order to use the best possible model for every instance. The main goal of these fields is, however, to build a superior model, and not so much to explain why and when which underlying model works best.

3 SCIENTIFIC RESEARCH OBJECTIVES

One of the main reasons why recommender systems (and per extension machine learning algorithms) lack native interpretability, is the intractable scale at which they operate. Often, vast amounts of data are required for these algorithms to learn from, up to the point where a human can no longer manually inspect every input/output pair. Though this may seem like a problem at first, we believe the readily availability of large amounts of data can instead be exploited to help circumvent the black box problem. More concretely we will investigate these research questions:

RQ1 How and why do the results of two machine learning algorithms differ for a given dataset?
  (1) Which ways to describe differences are most suited given the context?
  (2) Is it possible to derive what the most likely causes for these differences are?
  (3) Can knowledge of the differences be used to dynamically combine algorithms by selecting/weighting them?

RQ2 Can we automatically find subsets of a dataset where the performance of an algorithm differs? For example through a partitioning over time, user gender or item popularity.

We give an overview of four concrete research objectives (RO) that relate to our more general research questions. Each objective is first described and then the contribution of our research w.r.t. the objective is discussed.

RO1 More Informed Evaluation. As mentioned, accurately evaluating the performance of recommender systems in an offline setting, remains an open problem. The most notable implication of this problem, is that it remains difficult to experimentally validate current and future innovations in recommendation algorithms. Furthermore, with biased metrics that, additionally, only reflect a small amount of the desired properties, we may even be optimizing for the wrong goals and discarding interesting algorithms of which the value cannot accurately be formalised yet [14].

⇒ The main goal of this research is to extend and improve evaluation methods to provide more insight, which contributes to better model selection.

RO2 Improve Performance. Consider the phenomenon of cannibalisation [15], where the presence of some items in a top-K recommendation list causes other items to be consumed less often. A recommendation algorithm that can account for this would, without doubt, be of great value in certain contexts (e.g. to optimise top-K recommendations). However there exists, to the best of our knowledge, no formal way to measure the degree of cannibalisation, nor to take it into account during the generation of candidates.

⇒ Discovering and quantifying cannibalisation as a pattern can potentially improve the performance of recommendation algorithms. Similarly, other interesting phenomena, such as item fatigue and CTR variation [21], may be described as actionable patterns.

RO3 Gain Insight in Behaviour of Model. Data scientists should always be wary of the risk that an incidental relation was learnt between the (often complex) features and the labels. The husky or wolf classification is a good example of this, where the classifier learnt to recognise the snow on all wolf images, instead of the actual wolves [18]. Detecting this problem is a first and non-trivial step towards avoiding or correcting it.

⇒ Our research includes finding interesting subsets of the data where model performance varies. For example, if the "snow" feature can be formalised, our techniques should pick up that all misclassified husky images also contain snow and that misclassified wolves tend to be missing this feature.

RO4 A Step Towards Fairness, Accountability and Transparency. Insight into the behaviour of the model can also contribute to the concepts of fairness, accountability and transparency, in the sense that bringing to light what the system is doing, can help raise the practitioner's attention to hidden issues.

⇒ Some of the issues that can be detected are undesired discrimination against subgroups of users or items, strong filter bubbles and polarisation [4–6].

4 RESEARCH METHODOLOGY

We primarily focus on *recommender systems*, and additionally consider generalisations to other machine learning tasks.

4.1 Recommender Systems - Offline

To achieve our research objectives, we propose the method described in Figure 3, which illustrates how a recommender system is typically evaluated in an offline setting. However, in addition to summarizing the system with various metrics, a more informed and extensive *evaluation* step is included. In this evaluation step the goal is to learn insightful patterns from the inputs and the outputs of the system(s) under study.

Though the idea behind the method is not new, automating the process is still considered non-trivial due to the need for domain knowledge to define interestingness and the pattern explosion problem. One can imagine it is trivial to look for a specific pattern (e.g. "How many times are the items X and Y recommended together?"),

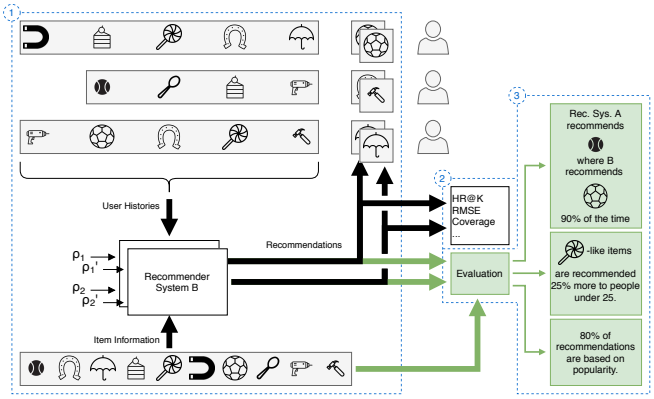

Figure 3: Global overview of our approach. 1) User history and item information are used by recommender systems to predict candidate items a user might like. 2) In offline evaluation, algorithms are typically evaluated by leaving items out of the training history and inspecting whether they appear in the top-K recommendations. 3) Pattern mining with the right interestingness measures could provide more detailed insights into the global behaviour of the system. Two systems (or the influence of a parameter (ρ_i) on one system) can even be compared by inspecting the differences between the two result sets.

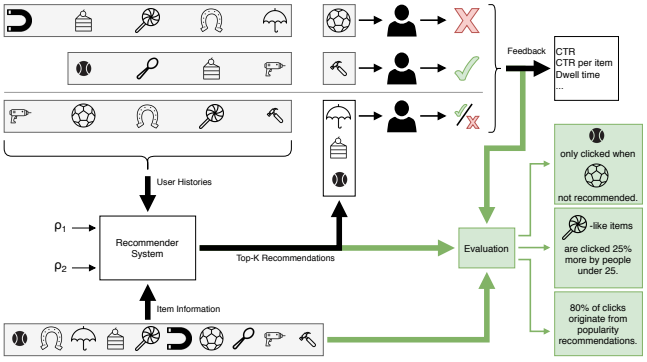

Figure 4: In online evaluation (A/B tests), user feedback is captured and aggregated in metrics like clickthrough rate (CTR) and dwell time, which in turn are used to assess the performance of a recommender system. Coupling the user responses back to the recommendations and item information however can potentially lead to interesting insights into *why* the recommender systems performs good or bad.

whereas generating all *interesting* patterns that make sense from the perspective of the user, is non-trivial.

Another challenge is to combine the output of two or more recommendation algorithms (or instances with different parameters) and define in which respect they differ. Typically one simply compares them by evaluating their performance on a selected collection of datasets, for a selected set of metrics and concludes that A is better than B. Our research will be directed at finding a reason as to *why* A performs better than B. If this problem is solved, we can equivalently also find the influence a parameter has on the results of an algorithm by comparing those two instances of the algorithm.

For example, given our approach, it would be possible to discover that one system performs better on a specific subset of the users (or items) [5]. It can also help guide the optimisation process by clarifying the effects of certain parameters, and in the limit even improve recommendations by selecting different algorithms for different users. The feasibility of this idea has already been demonstrated in the context of news recommendations, where it was found that shifting the weight of popularity-based recommendations based on the time of day, improved performance [13].

4.2 Recommender Systems - Online

Complementary to offline evaluation, online evaluation (in the form of A/B testing) can provide richer metrics that are closer to the dependent variable. Naturally, we should be able to extend the idea described in Figure 3 to the online scenario. The envisioned method and possible results are described in Figure 4, where we can see the addition of user feedback as input. Finding cannibalisation for example, can now be described as a pattern with a certain support (see first result pattern in Figure 4). What's notable here is that this kind of pattern requires the absence (or complement) of some item(s) to be taken into account. Finding interesting patterns of this type makes the problem even more complex, because items are absent much more frequently and not at random (MNAR), which can complicate pruning. In addition, often the absence of an item is not meaningful, resulting in an exponential increase in hard to filter, meaningless patterns. Little work has been done towards finding interesting patterns when complements are considered.

Furthermore, the comparison of two algorithms can also trivially be extended to the online case.

4.3 Generalisation to Other Supervised Tasks

Recommenders fit within the class of supervised learning algorithms. Subsequently, it would be interesting to investigate whether the results and techniques of this research can be generalized to other supervised learning tasks.

5 CONCLUSION

Recommender systems are inherently difficult to evaluate offline, and even in online evaluation, clear explanations of their performance are still required to draw accurate conclusions. In addition, recommender systems often operate at a large scale and many popular algorithms have a black box nature. We address the need for exploratory evaluation methods that, in addition to summarising the performance of recommendation algorithms, also discern subsets of interest or general reasons that explain the achieved results.

ACKNOWLEDGMENTS

This work is conducted under the supervision of Prof. Bart Goethals. Special thanks to Jan Van Balen and Olivier Jeunen for their valuable feedback. This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

REFERENCES

[1] Charu C. Aggarwal and Jiawei Han. 2014. *Frequent Pattern Mining.* Springer Publishing Company, Incorporated.

[2] Joeran Beel. 2017. It's Time to Consider "Time" when Evaluating Recommender-System Algorithms [Proposal]. arXiv:1708.08447 [cs.IR]

[3] Toon Calders and Bart Goethals. 2002. Mining All Non-Derivable Frequent Itemsets. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD '02).* Springer-Verlag, Berlin, Heidelberg, 74–85.

[4] Michael D Ekstrand, Robin Burke, and Fernando Diaz. 2019. Fairness and Discrimination in Recommendation and Retrieval. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19).* Association for Computing Machinery, New York, NY, USA, 576–577. https://doi.org/10.1145/3298689.3346964

[5] Michael D Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera. 2018. All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. In *Conference on Fairness, Accountability and Transparency.* 172–186.

[6] Michael D. Ekstrand, Mucun Tian, Mohammed R. Imran Kazi, Hoda Mehrpouyan, and Daniel Kluver. 2018. Exploring Author Gender in Book Rating and Recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18).* Association for Computing Machinery, New York, NY, USA, 242–250. https://doi.org/10.1145/3240323.3240373

[7] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. 2019. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research* 20, 177 (2019), 1–81.

[8] Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Bruttin, and Amr Huber. 2014. Offline and Online Evaluation of News Recommender Systems at Swissinfo.Ch. In *Proceedings of the 8th ACM Conference on Recommender Systems* (Foster City, Silicon Valley, California, USA) *(RecSys '14).* Association for Computing Machinery, New York, NY, USA, 169–176. https://doi.org/10.1145/2645710.2645745

[9] Liqiang Geng and Howard J Hamilton. 2006. Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR)* 38, 3 (2006), 9–es.

[10] Olivier Jeunen, Koen Verstrepen, and Bart Goethals. 2018. Fair Offline Evaluation Methodologies for Implicit-Feedback Recommender Systems with MNAR Data. In *Proc. of the REVEAL 18 Workshop on Ofine Evaluation for Recommender Systems (RecSys' 18).*

[11] Zachary C Lipton. 2018. The mythos of model interpretability. *Queue* 16, 3 (2018), 31–57.

[12] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17).* Curran Associates Inc., Red Hook, NY, USA, 4768–4777.

[13] Andrii Maksai, Florent Garcin, and Boi Faltings. 2015. Predicting Online Performance of News Recommender Systems Through Richer Evaluation Metrics. In *Proceedings of the 9th ACM Conference on Recommender Systems* (Vienna, Austria) *(RecSys '15).* Association for Computing Machinery, New York, NY, USA, 179–186. https://doi.org/10.1145/2792838.2800184

[14] Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems* (Montréal, Québec, Canada) *(CHI EA '06).* Association for Computing Machinery, New York, NY, USA, 1097–1101. https://doi.org/10.1145/1125451.1125659

[15] Sandy Moens, Olivier Jeunen, and Bart Goethals. 2019. Interactive Evaluation of Recommender Systems with SNIPER: An Episode Mining Approach. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19).* Association for Computing Machinery, New York, NY, USA, 538–539. https://doi.org/10.1145/3298689.3346965

[16] Christoph Molnar. 2019. *Interpretable machine learning.* Lulu. com.

[17] Yanou Ramon, David Martens, Foster Provost, and Theodoros Evgeniou. 2019. Counterfactual Explanation Algorithms for Behavioral and Textual Data. arXiv:1912.01819 [cs.AI]

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16).* Association for Computing Machinery, New York, NY, USA, 1135–1144. https://doi.org/10.1145/2939672.2939778

[19] Marco Rossetti, Fabio Stella, and Markus Zanker. 2016. Contrasting Offline and Online Results When Evaluating Recommendation Algorithms. In *Proceedings of the 10th ACM Conference on Recommender Systems* (Boston, Massachusetts, USA) *(RecSys '16).* Association for Computing Machinery, New York, NY, USA, 31–34. https://doi.org/10.1145/2959100.2959176

[20] Alan Said, Jimmy Lin, Alejandro Bellogín, and Arjen de Vries. 2013. A Month in the Life of a Production News Recommender System. In *Proceedings of the 2013 Workshop on Living Labs for Information Retrieval Evaluation* (San Francisco, California, USA) *(LivingLab '13).* Association for Computing Machinery, New York, NY, USA, 7–10. https://doi.org/10.1145/2513150.2513159

[21] Oren Sar Shalom, Noam Koenigstein, Ulrich Paquet, and Hastagiri P. Vanchinathan. 2016. Beyond Collaborative Filtering: The List Recommendation Problem. In *Proceedings of the 25th International Conference on World Wide Web* (Montréal, Québec, Canada) *(WWW '16).* International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 63–72. https://doi.org/10.1145/2872427.2883057

[22] D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. 2015. Hidden Technical Debt in Machine Learning Systems. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2* (Montreal, Canada) *(NIPS'15).* MIT Press, Cambridge, MA, USA, 2503–2511.

[23] Harald Steck, Maria Dimakopoulou, Nickolai Riabov, and Tony Jebara. 2020. ADMM SLIM: Sparse Recommendations for Many Users. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (Houston, TX, USA) *(WSDM '20).* Association for Computing Machinery, New York, NY, USA, 555–563. https://doi.org/10.1145/3336191.3371774

[24] Joaquin Vanschoren. 2018. Meta-Learning: A Survey. arXiv:1810.03548 [cs.LG]

[25] Ricardo Vilalta and Youssef Drissi. 2002. A perspective view and survey of meta-learning. *Artificial intelligence review* 18, 2 (2002), 77–95.

[26] Jilles Vreeken, Matthijs Van Leeuwen, and Arno Siebes. 2011. Krimp: mining itemsets that compress. *Data Mining and Knowledge Discovery* 23, 1 (2011), 169–214.

[27] Geoffrey I Webb. 2010. Self-sufficient itemsets: An approach to screening potentially interesting associations between items. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 4, 1 (2010), 1–20.

[28] Geoffrey I Webb, Loong Kuan Lee, Bart Goethals, and François Petitjean. 2018. Analyzing concept drift and shift from sample data. *Data Mining and Knowledge Discovery* 32, 5 (2018), 1179–1199.

[29] David H Wolpert and William G Macready. 1997. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation* 1, 1 (1997), 67–82.

[30] Indrė Žliobaitė, Mykola Pechenizkiy, and Joao Gama. 2016. An overview of concept drift applications. In *Big data analysis: new algorithms for a new society.* Springer, 91–114.