

本当に進歩しているのでしょうか?最近のニューラル・レコメンデーション・アプローチの心配な分析

Maurizio Ferrari Dacrema
Politecnico di Milano, Italy
maurizio.ferrari@polimi.it

Paolo Cremonesi
イタリア、ミラノのパオロ・クレモネ
シ政治家 paolo.cremonesi@polimi.it

Dietmar Jannach
クラゲンフルト大学ディートマー、ヤンナ
ッハオーストリア dietmar.jannach@aau.at

ABSTRACT

ディープラーニング技術は、レコメンダーシステムのアルゴリズムの側面に取り組む研究者にとって選択される手法となっている。機械学習全般への関心が強くなるにつれ、その結果、例えばトップn推薦タスクのように、現時点で最先端に行くことが難しくなっている。同時に、最近のいくつかの出版物は、応用機械学習における今日の研究実践における問題を指摘している。例えば、新しいモデルを提案する際の結果の再現性やベースラインの選択などである。本研究では、トップn推薦タスクのためのアルゴリズム提案の系統的分析の結果を報告する。具体的には、ここ数年のトップレベルの研究会議で発表された18のアルゴリズムを検討した。そのうち7つだけが、妥当な努力で再現できた。しかし、これらの手法では、最近傍やグラフベースの手法など、比較的単純な発見的手法で6つが上回ることができることが多いことが判明した。残りの1つは、ベースラインを明らかに上回ったが、よく調整された非ニューラル線形ランキング法を一貫して上回らなかった。全体として、我々の研究は、今日の機械学習研究における多くの潜在的な問題に光を当て、この分野における科学的実践の改善を求めている。

CCS CONCEPTS

– 情報システム → 協調フィルタリング; 推薦システム; – 一般と参照 → 評価。

KEYWORDS

推薦システム; ディープラーニング; 評価; 再現性

ACMリファレンス形式:

マウリツィオ・フェラーリ・ダクレマ、パオロ・クレモネシ、ディートマー・ヤナツラインは、与えられたタスクとデータセットに対して、一般的に弱す。2019. 私たちは本当に多くの進歩を遂げているのか?最近のニューラル・レコメンデーション・アプローチの心配な分析。第13回ACMレコメンダーシステム会議(RecSys '19)、2019年9月16日~20日、コペンハーゲン、デンマーク。ACM, ニューヨーク, NY 2019. 10ページ. <https://doi.org/10.1145/3298689.3347058>

この著作物の全部または一部を、個人的または教室で使用するために、デジタルまたはハードコピーを作成する許可を無償で与える。ただし、営利目的または商業的利益のためにコピーを作成または配布せず、コピーにはこの通知と最初のページに引用の全文を記載することを条件とする。ACM以外の者が所有する本作品の構成要素の著作権は尊重されなければならない。クレジットによる抽象化が許可される。それ以外の方法でコピーしたり、再出版したり、サーバーに投稿したり、リストに再配布したりするには、事前の特別な許可と/または料金が必要である。permissions@acm.org. RecSys '19, 2019年9月16日~20日, デンマーク, コペンハーゲン © 2019 計算機学会. ACM ISBN 978-1-4503-6243-6/19/09.

1 INTRODUCTION

わずか数年の間に、ディープラーニング技術がレコメンダーシステムにおけるアルゴリズム研究の状況を支配し始めた。長期的な嗜好プロファイルに基づくトップn推薦や、セッションベースの推薦シナリオなど、様々な設定やアルゴリズムタスクに対して新しい手法が提案された[36]。一般的な機械学習への関心の高まり、それに伴う最近の研究発表の数、視覚や言語処理などの他の分野での深層学習技術の成功を考えると、推薦システムの分野でもこれらの研究が大きな進歩を遂げたと期待できる。しかし、機械学習の他の応用分野では、達成された進歩(既存のモデルに対する精度の向上という観点から測定)が、必ずしも期待されたほど強いとは限らないという指摘が存在する。

例えば、Lin [25]は、トップレベルの学会で発表された、情報検索の分野における最近の2つのニューラル・アプローチについて論じている。彼の分析によると、新しい手法は、これらを注意深く調整した場合、既存のベースライン手法を有意に上回らないことが明らかになった。推薦システムの文脈では、[29]で提示された詳細な分析により、セッションベースの推薦のための非常に最近のニューラル手法でさえ、ほとんどの場合、最近傍技術などに基づく非常に単純な手法に負けることができることが示されている。一般に、このような応用的な機械学習の場面で達成される真の進歩に関する疑問は新しいものではなく、また深層学習に基づく研究とも結びついているものでもない。すでに2009年にArmstrongら[2]は、アドホック検索タスクの文脈での分析から、多くの論文が発表されているにもかかわらず、次のように結論付けている、

the reported improvements “don’t add up”.

このような現象には、(i)弱いベースライン、(ii)弱い手法を新しいベースラインとして確立すること、(iii)論文間で結果を比較したり再現したりすることの難しさなど、さまざまな要因が寄与している。最初の1つの問題は、比較に使用されるベースラインの選択にある。ベースラインとのみ比較される場合などである。この挙動は弱いベースラインの伝搬を強制する。以前の深層学習アルゴリズムが弱すぎるベースラインに対して評価された場合、新しい深層学習アルゴリズムは必ずしも強力な非ニューラルベースラインよりも改善されない。さらに、近年、論文の流れが絶え間なく続く中、最先端のベースラインを示すものを追跡することは、ますます困難になっている。

ベースラインに関する問題の他に、研究者が様々なタイプのデータセット、評価プロトコル、性能測定、データの前処理ステップを使用することが追加の課題であり、異なるアプリケーションシナリオにおいてどの方法が最適であるかを結論づけることを困難にしている。

Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches

Maurizio Ferrari Dacrema
Politecnico di Milano, Italy
maurizio.ferrari@polimi.it

Paolo Cremonesi
Politecnico di Milano, Italy
paolo.cremonesi@polimi.it

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

ABSTRACT

Deep learning techniques have become the method of choice for researchers working on algorithmic aspects of recommender systems. With the strongly increased interest in machine learning in general, it has, as a result, become difficult to keep track of what represents the state-of-the-art at the moment, e.g., for top-n recommendation tasks. At the same time, several recent publications point out problems in today's research practice in applied machine learning, e.g., in terms of the reproducibility of the results or the choice of the baselines when proposing new models.

In this work, we report the results of a systematic analysis of algorithmic proposals for top-n recommendation tasks. Specifically, we considered 18 algorithms that were presented at top-level research conferences in the last years. Only 7 of them could be reproduced with reasonable effort. For these methods, it however turned out that 6 of them can often be outperformed with comparably simple heuristic methods, e.g., based on nearest-neighbor or graph-based techniques. The remaining one clearly outperformed the baselines but did not consistently outperform a well-tuned non-neural linear ranking method. Overall, our work sheds light on a number of potential problems in today's machine learning scholarship and calls for improved scientific practices in this area.

CCS CONCEPTS

• **Information systems** → **Collaborative filtering; Recommender systems**; • **General and reference** → **Evaluation**.

KEYWORDS

Recommender Systems; Deep Learning; Evaluation; Reproducibility

ACM Reference Format:

Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19)*, September 16–20, 2019, Copenhagen, Denmark. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3298689.3347058>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6243-6/19/09...\$15.00

<https://doi.org/10.1145/3298689.3347058>

1 INTRODUCTION

Within only a few years, deep learning techniques have started to dominate the landscape of algorithmic research in recommender systems. Novel methods were proposed for a variety of settings and algorithmic tasks, including top-n recommendation based on long-term preference profiles or for session-based recommendation scenarios [36]. Given the increased interest in machine learning in general, the corresponding number of recent research publications, and the success of deep learning techniques in other fields like vision or language processing, one could expect that substantial progress resulted from these works also in the field of recommender systems. However, indications exist in other application areas of machine learning that the achieved progress—measured in terms of accuracy improvements over existing models—is not always as strong as expected.

Lin [25], for example, discusses two recent neural approaches in the field of information retrieval that were published at top-level conferences. His analysis reveals that the new methods do *not* significantly outperform existing baseline methods when these are carefully tuned. In the context of recommender systems, an in-depth analysis presented in [29] shows that even a very recent neural method for session-based recommendation can, in most cases, be outperformed by very simple methods based, e.g., on nearest-neighbor techniques. Generally, questions regarding the true progress that is achieved in such applied machine learning settings are not new, nor tied to research based on deep learning. Already in 2009, Armstrong et al. [2] concluded from an analysis in the context of ad-hoc retrieval tasks that, despite many papers being published, the reported improvements “don’t add up”.

Different factors contribute to such phenomena, including (i) weak baselines; (ii) establishment of weak methods as new baselines; and (iii) difficulties in comparing or reproducing results across papers. One first problem lies in the choice of the baselines that are used in the comparisons. Sometimes, baselines are chosen that are too weak in general for the given task and dataset, and sometimes the baselines are not properly fine-tuned. Other times, baselines are chosen from the same family as the newly proposed algorithm, e.g., when a new deep learning algorithm is compared only against other deep learning baselines. This behaviour enforces the propagation of weak baselines. When previous deep learning algorithms were evaluated against too weak baselines, the new deep learning algorithm will not necessarily improve over strong non-neural baselines. Furthermore, with the constant flow of papers being published in recent years, keeping track of what represents a state-of-the-art baseline becomes increasingly challenging.

Besides issues related to the baselines, an additional challenge is that researchers use various types of datasets, evaluation protocols, performance measures, and data preprocessing steps, which makes

ソースコードとデータが共有されない場合、特に問題となる。研究者がアルゴリズムのソースコードを公開する傾向が強まっているが、トップレベルの公開媒体であっても、これは今日では一般的なルールではない。また、コードが公開されている場合でも、不完全な場合があり、例えば、[15]でも指摘されているように、データの前処理、パラメータのチューニング、正確な評価手順のためのコードが含まれていない。

最後に、もう一つの一般的な問題は、応用機械学習全般における今日の研究実践にあるかもしれない。いくつかの「牽引傾向」が[27]で議論されており、その中には査読者ブールの薄さや、ある種の研究を刺激する可能性のある著者に対するインセンティブのずれなどがある。先行研究[46]では、抽象的な精度測定や機械学習研究の狭い焦点に焦点を当てたコミュニティについて、一流出版物で何が「公開可能」であるかという観点から論じている。

この研究成果により、我々の目標は、上記で報告された問題が深層学習ベースの推薦アルゴリズムの領域にも存在するかどうかという疑問に光を当てることである。具体的には、主に2つの研究課題に取り組む：

- (1) 再現性: (1)再現性:この分野の最近の研究は、どの程度再現可能か(妥当な努力で再現可能か)?
- (2) *Progress*: To what extent are recent algorithms actually leading to better performance results when compared to relatively simple, but well-tuned, baseline methods?

これらの疑問に答えるため、我々は、ディープラーニング手法を用いたトップn推薦タスクのための新しいアルゴリズムアプローチを提案した研究論文を分析する体系的な研究を行った。そのために、KDD, SIGIR, TheWebConf (WWW), RecSysの最近の会議録をスキャンし、対応する研究成果を探した。関連する論文を18件特定した。

最初のステップでは、著者によってソースコードが利用可能になり、実験に使用したデータにアクセスできるようになったケースについて、論文で報告された結果を再現することを試みた。最終的に、発表された結果を再現できたのは、わずか7論文で、許容できる程度の確実性であった。したがって、我々の研究の第一の貢献は、この分野における現在の研究の再現性のレベルの評価である。

本研究の第2部では、原著論文で報告された実験を再実行したが、比較にはベースライン手法も追加した。具体的には、ユーザーベースとアイテムベースの最近傍に基づく発見的手法と、単純なグラフベースのアプローチの2つのバリエーションを使用した。我々の研究は、少し驚いたことに、調査されたケースの大部分(7つ中6つ)において、提案された深層学習技術は、単純だが微調整されたベースライン手法を一貫して上回らなかったことを明らかにした。あるケースでは、最も人気のあるアイテムを誰にでも推薦する非パーソナライズド手法でさえ、ある精度指標では最良のものであった。したがって、我々の第二の貢献は、機械学習における現在の研究実践に関連する、より広範囲に及ぶ可能性のある問題を特定したことにある。本稿の構成は以下の通りである。次に、セクション 2 では、我々の研究方法と既存研究の再現方法について述べる。

セクション3では、ベースラインを追加しながら実験を再実行した結果を示す。最後にセクション4で我々の研究の意味について述べる。

2 研究方法

2.1 再現可能な論文の収集

我々の研究が、最近発表された研究の個々の例に基づくだけでなく、関連する長い論文の科学的会議のプロシーディングを手作業で体系的にスキャンした。具体的には、以下の4つの学会シリーズで2015年から2018年の間に発表された長い論文を分析に含めた：KDD, SIGIR, TheWebConf (WWW), RecSys.¹ (a)深層学習ベースの手法を提案し、(b)トップn推薦問題に焦点を当てた論文に関連性があるとみなした。その他の推薦タスク、例えば、グループ推薦やセッションベース推薦に関する論文は、我々の分析では考慮しなかった。トップn推薦への関心を考慮し、Precision, Recall, MAPなどの評価分類やランキングメトリクスに使用した論文のみを対象とした。このスクリーニングの結果、最終的に18の関連論文が集まった。

次のステップでは、これらの論文で報告された結果を再現することを試みた²。再現性を高めるための我々のアプローチは、著者自身によって提供された成果物、すなわち、そのソースコードと実験に使用されたデータに可能な限り依存することである。理論的には、論文中の技術的な記述のみを用いて、発表された結果を再現することが可能であるはずである。現実には、アルゴリズムの実装や評価手順、例えばデータ分割に関する些細な詳細が、実験結果に影響を与える可能性があるが、実際には、多くの詳細が存在する[39]。

そこで、著者からすべての関連論文のコードとデータを取得することを試みた。これらの成果物がすでに公開されていない場合は、論文の著者全員に連絡を取り、回答待ち30日間とした。結局、以下の条件を満たした論文を再現性があると判断した。

- ソースコードの動作バージョンが利用可能であるか、正しく動作するために最小限の方法でコードを変更するだけでよい。
- 原著論文で使用されたデータセットが少なくとも1つ利用可能である。ここでの更なる要件は、元々使用されていた訓練とテストの分割が公開されているか、論文の情報に基づいて再構成できることである。

それ以外の場合は、我々の特定の再現アプローチからすると、論文は再現不可能であると考えている。ソースコードが公開されているが、多くの部分と詳細が欠落しているモデルのスケルトン版のみを含んでいる場合にも、再現不可能であると考えたことに注意されたい。データセットに関しては、企業が所有する非公開データや、ウェブから何らかの形で収集されたが公に共有されていないデータのみに基づく研究も再現性があるとはみなされなかった。

学会シリーズごとに、比較的厳格な基準で再現可能な論文の割合を表 1 に示す。

¹すべての会議は、Australian Core RankingのA*とみなされるか、特に推薦システムの研究に特化している。つまり、著者から提供された成果物と、私たち自身の成果物の両方を使用した。読みやすくするために、本稿では「再現性」という用語のみを使用する。³ コアアルゴリズムに修正を加えなかった。

it difficult to conclude which method is the best across different application scenarios. This is in particular problematic when source code and data are not shared. While we observe an increasing trend that researchers publish the source code of their algorithms, this is not the common rule today even for top-level publication outlets. And even in cases when the code is published, it is sometimes incomplete and, for instance, does not include the code for data pre-processing, parameter tuning, or the exact evaluation procedures, as pointed out also in [15].

Finally, another general problem might lie in today's research practice in applied machine learning in general. Several "troubling trends" are discussed in [27], including the thinness of reviewer pools or misaligned incentives for authors that might stimulate certain types of research. Earlier work [46] also discusses the community's focus on abstract accuracy measures or the narrow focus of machine learning research in terms of what is "publishable" at top publication outlets.

With this research work, our goal is to shed light on the question if the problems reported above also exist in the domain of deep learning-based recommendation algorithms. Specifically, we address two main research questions:

- (1) *Reproducibility*: To what extent is recent research in the area reproducible (with reasonable effort)?
- (2) *Progress*: To what extent are recent algorithms actually leading to better performance results when compared to relatively simple, but well-tuned, baseline methods?

To answer these questions, we conducted a systematic study in which we analyzed research papers that proposed new algorithmic approaches for top-n recommendation tasks using deep learning methods. To that purpose, we scanned the recent conference proceedings of KDD, SIGIR, TheWebConf (WWW), and RecSys for corresponding research works. We identified 18 relevant papers.

In a first step, we tried to reproduce the results reported in the paper for those cases where the source code was made available by the authors and where we had access to the data used in the experiments. In the end, we could reproduce the published results with an acceptable degree of certainty for only 7 papers. A first contribution of our work is therefore an assessment of the reproducibility level of current research in the area.

In the second part of our study, we re-executed the experiments reported in the original papers, but also included additional baseline methods in the comparison. Specifically, we used heuristic methods based on user-based and item-based nearest neighbors as well as two variants of a simple graph-based approach. Our study, to some surprise, revealed that in the large majority of the investigated cases (6 out of 7) the proposed deep learning techniques did not consistently outperform the simple, but fine-tuned, baseline methods. In one case, even a non-personalized method that recommends the most popular items to everyone was the best one in terms of certain accuracy measures. Our second contribution therefore lies in the identification of a potentially more far-reaching problem related to current research practices in machine learning.

The paper is organized as follows. Next, in Section 2, we describe our research method and how we reproduced existing works.

The results of re-executing the experiments while including additional baselines are provided in Section 3. We finally discuss the implications of our research in Section 4.

2 RESEARCH METHOD

2.1 Collecting Reproducible Papers

To make sure that our work is not only based on individual examples of recently published research, we systematically scanned the proceedings of scientific conferences for relevant long papers in a manual process. Specifically, we included long papers in our analysis that appeared between 2015 and 2018 in the following four conference series: KDD, SIGIR, TheWebConf (WWW), and RecSys.¹ We considered a paper to be relevant if it (a) proposed a deep learning based technique and (b) focused on the top-n recommendation problem. Papers on other recommendation tasks, e.g., group recommendation or session-based recommendation, were not considered in our analysis. Given our interest in top-n recommendation, we considered only papers that used for evaluation classification or ranking metrics, such as Precision, Recall, MAP. After this screening process, we ended up with a collection of 18 relevant papers.

In a next step, we tried to reproduce² the results reported in these papers. Our approach to reproducibility is to rely as much as possible on the artifacts provided by the authors themselves, i.e., their source code and the data used in the experiments. In theory, it should be possible to reproduce published results using only the technical descriptions in the papers. In reality, there are, however many tiny details regarding the implementation of the algorithms and the evaluation procedure, e.g., regarding data splitting, that can have an impact on the experiment outcomes [39].

We therefore tried to obtain the code and the data for all relevant papers from the authors. In case these artifacts were not already publicly provided, we contacted all authors of the papers and waited 30 days for a response. In the end, we considered a paper to be *reproducible*, if the following conditions were met:

- A working version of the *source code* is available or the code only has to be modified in minimal ways to work correctly.³
- At least one *dataset* used in the original paper is available. A further requirement here is that either the originally-used train-test splits are publicly available or that they can be reconstructed based on the information in the paper.

Otherwise, we consider a paper to be *non-reproducible* given our specific reproduction approach. Note that we also considered works to be non-reproducible when the source code was published but contained only a skeleton version of the model with many parts and details missing. Concerning the datasets, research based solely on non-public data owned by companies or data that was gathered in some form from the web but not shared publicly, was also not considered reproducible.

The fraction of papers that were reproducible according to our relatively strict criteria per conference series are shown in Table 1.

¹All of the conferences are either considered A* in the Australian Core Ranking or specifically dedicated to research in recommender systems.

²Precisely speaking, we used a mix of replication and reproduction [12, 35], i.e., we used both artifacts provided by the authors and our own artifacts. For the sake of readability, we will only use the term "reproducibility" in this paper.

³We did not apply modifications to the core algorithms.

表1: 2015年から2018年までの会議シリーズごとのトップn推薦のための深層学習アルゴリズムに関する再現可能な作品。

Conference	Rep. ratio	Reproducible
KDD	3/4 (75%)	[17], [23], [48]
RecSys	1/7 (14%)	[53]
SIGIR	1/3 (30%)	[10]
WWW	2/4 (50%)	[14], [24]
Total	7/18 (39%)	
Non-reproducible: KDD: [43], RecSys: [41], [6], [38], [44], [21], [45], SIGIR: [32], [7], WWW: [42], [11]		

全体として、再現性が低いというこれまでの議論を裏付けるように、研究の約3分の1しか再現できなかった。会議シリーズ間の差について信頼できる結論を出すには、サンプルサイズが小さすぎる。しかし、紙面の都合上、ここでは示していないが、年ごとの詳細な統計は、再現率が年々増加していることを示している。

2.2 評価方法

測定方法。ベースラインのセットに対して新しい方法によって達成される進歩の検証は、少なくとも2つの方法で行うことができる。一つは、[29]で行われたのと同じデータセットと、すべてのアルゴリズムについて全く同じ評価手順を用いて、同じ定義された環境内で検討されたすべての手法を評価することである。このようなアプローチは、データセット間で異なる手法がどのように比較されるかを知るのに役立つが、実装された評価手順は、元の論文で使われたものとは若干異なるかもしれない。そのため、このアプローチでは、本研究の目的である、当初報告されたものを正確に再現することはできない。

そこで、本研究では、原著論文で使われたものと同じ評価手順を適用できるような方法で、原著の実装を再ファクタリングすることで、この研究を再現する。具体的には、学習、ハイパーパラメータ最適化、予測のための元のコードが評価コードから分離されるようにリファクタリングが行われる。この評価コードはベースラインにも使用される。

個々の実験で考慮されたすべての再現アルゴリズムについて、各データセットについて、原著論文で著者らが報告した最適なハイパーパラメータを使用した。これは、原著論文と同じデータセット、アルゴリズム実装、評価手順を使用したため適切である⁴。実験に使用した全てのコードとデータ、ベースラインの最終的なアルゴリズム(ハイパー)パラメータの詳細、およびオンラインの全実験結果を共有する⁵。

ベースライン。我々の実験では以下のベースライン手法を検討したが、これらはすべて概念的に単純である。

TopPopular(トピラ)。最も人気のあるアイテムを誰にでも推薦する非パーソナライズドメソッド。人気度は、明示的または暗黙的な評価の数によって測定される。
ItemKNN: k-最近傍探索(KNN)と項目-項目類似度に基づく伝統的な協調フィルタリング(CF)アプローチ[49]。として計算された項目*i*と*j*の間の余弦類似度 s_{ij} を用いた。

$$s_{ij} = \frac{\mathbf{r}_i \cdot \mathbf{r}_j}{\|\mathbf{r}_i\| \|\mathbf{r}_j\| + h} \quad (1)$$

ここで、 $\text{vectors } \mathbf{r}_{i,j} \in \mathbb{R}^{|\mathcal{U}|}$ はそれぞれアイテム*i*と*j*に対するユーザの暗黙の評価を表し、 $|\mathcal{U}|$ はユーザの数である。評価は、[50]に記述されているように、TF-IDFまたはBM25で任意に重み付けすることができる。さらに、類似度はベクトルノルムの積によって正規化される場合とされない場合がある。パラメータ*h*(収縮項)は、相互作用が少ない項目間の類似度を下げるために使用される[5]。この方法のもう一つのパラメータは、近傍サイズ*k*である。

UserKNN: 協調的なユーザー-ユーザー類似度を用いた近傍ベースの手法。ハイパーパラメータはItemKNN [40]で使ったものと同じである。

ItemKNN-CBF: 項目内容特徴(属性)を用いて項目の類似度を計算する近傍内容ベースフィルタリング(CBF)アプローチ

$$s_{ij} = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{\|\mathbf{f}_i\| \|\mathbf{f}_j\| + h} \quad (2)$$

ここで、ベクトル $\mathbf{f}_{i,j} \in \mathbb{R}^{|\mathcal{F}|}$ はそれぞれ項目*i*と*j*の特徴を表し、 $|\mathcal{F}|$ は特徴の数である。特徴量はTF-IDFまたはBM25で重み付けできる。その他のパラメータはItemKNN [28]と同じである。

ItemKNN-CFCBF: 項目項目の類似性に基づくハイブリッドCF+CBFアルゴリズム。類似度は、まず各項目*i*について、評価ベクトルと特徴ベクトル $[\mathbf{r}_i, \mathbf{w}_i]$ を連結し、後に連結されたベクトル間の余弦類似度を計算することで計算される。ハイパーパラメータはItemKNNで使ったものと同じで、さらに評価に関してコンテンツ特徴を重み付けするパラメータ*w*を加えたものである。

$P^3 \alpha$: ユーザとアイテム間のランダムウォークを実装した単純なグラフベースのアルゴリズム[8]。ユーザ*u*の項目は、ユーザ*u*から始まる3つのステップを持つランダムウォークの確率に基づいてランク付けされる。ユーザ*u*からアイテム*i*にジャンプする確率 p_{ui} は、暗黙のユーザ評価行列から $p_{ui} = (r_{ui} / N_u)^\alpha$ ここで、 r_{ui} はアイテム*i*に対するユーザ*u*の評価、 N_u はユーザ*u*の評価数、 α は減衰係数として計算される。後方にジャンプする確率 p_{iu} は $p_{iu} = (r_{ui} / N_i)^\alpha$ として計算される。この方法は、KNN項目ベースのCFアルゴリズムと等価であり、類似度行列は次のように定義される。

$$s_{ij} = \sum_v p_{jv} p_{vi} \quad (3)$$

この手法のパラメータは、近傍の数*k*と α の値である。このアルゴリズムは、低い計算コストで良好な推薦品質を提供するため、このアルゴリズムを含める。
 $RP^3 \beta$: [34]で提案された $P^3 \alpha$ のバージョン。ここで、 $P^3 \alpha$ の結果は、類似度を係数 β のべき乗に上げた各アイテムの人気度で割ることによって修正される。

⁴We will re-run parameter optimization for the reproduced algorithms as part of our future work in order to validate the parameter optimization procedures used by the authors. This step was, however, outside the scope of our current work.

⁵https://github.com/MaurizioFD/RecSys2019_DeepLearning_Evaluation

Table 1: Reproducible works on deep learning algorithms for top-n recommendation per conference series from 2015 to 2018.

Conference	Rep. ratio	Reproducible
KDD	3/4 (75%)	[17], [23], [48]
RecSys	1/7 (14%)	[53]
SIGIR	1/3 (30%)	[10]
WWW	2/4 (50%)	[14], [24]
Total	7/18 (39%)	
<i>Non-reproducible:</i> KDD: [43], RecSys: [41], [6], [38], [44], [21], [45], SIGIR: [32], [7], WWW: [42], [11]		

Overall, we could reproduce only about one third of the works, which confirms previous discussions about limited reproducibility, see, e.g., [3]. The sample size is too small to make reliable conclusions regarding the difference between conference series. The detailed statistics per year—not shown here for space reasons—however indicate that the reproducibility rate increased over the years.

2.2 Evaluation Methodology

Measurement Method. The validation of the progress that is achieved through new methods against a set of baselines can be done in at least two ways. One is to evaluate all considered methods within the same defined environment, using the same datasets and the exact same evaluation procedure for all algorithms as done in [29]. While such an approach helps us obtain a picture of how different methods compare across datasets, the implemented evaluation procedure might be slightly different from the one used in the original papers. As such, this approach would not allow us to *exactly* reproduce what has been originally reported, which is the goal in this present work.

In this work, we therefore reproduce the work by refactoring the original implementations in a way that allows us to apply the same evaluation procedure that was used in the original papers. Specifically, refactoring is done in a way that the original code for training, hyper-parameter optimization and prediction are separated from the evaluation code. This evaluation code is then also used for the baselines.

For all reproduced algorithms considered in the individual experiments, we used the optimal hyper-parameters that were reported by the authors in the original papers for each dataset. This is appropriate because we used the same datasets, algorithm implementation, and evaluation procedure as in the original papers.⁴ We share all the code and data used in our experiments as well as details of the final algorithm (hyper-)parameters of our baselines along with the full experiment results online.⁵

Baselines. We considered the following baseline methods in our experiments, all of which are conceptually simple.

TopPopular: A non-personalized method that recommends the most popular items to everyone. Popularity is measured by the number of explicit or implicit ratings.

ItemKNN: A traditional Collaborative-Filtering (CF) approach based on k -nearest-neighborhood (KNN) and item-item similarities [49]. We used the cosine similarity s_{ij} between items i and j computed as

$$s_{ij} = \frac{\mathbf{r}_i \cdot \mathbf{r}_j}{\|\mathbf{r}_i\| \|\mathbf{r}_j\| + h} \quad (1)$$

where vectors $\mathbf{r}_i, \mathbf{r}_j \in \mathbb{R}^{|U|}$ represent the implicit ratings of a user for items i and j , respectively, and $|U|$ is the number of users. Ratings can be optionally weighted either with TF-IDF or BM25, as described in [50]. Furthermore the similarity may or not be normalized via the product of vector norms. Parameter h (the *shrink term*) is used to lower the similarity between items having only few interactions [5]. The other parameter of the method is the neighborhood size k .

UserKNN: A neighborhood-based method using collaborative user-user similarities. Hyper-parameters are the same as used for ItemKNN [40].

ItemKNN-CBF: A neighborhood content-based-filtering (CBF) approach with item similarities computed by using item content features (attributes)

$$s_{ij} = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{\|\mathbf{f}_i\| \|\mathbf{f}_j\| + h} \quad (2)$$

where vectors $\mathbf{f}_i, \mathbf{f}_j \in \mathbb{R}^{|F|}$ describe the features of items i and j , respectively, and $|F|$ is the number of features. Features can be optionally weighted either with TF-IDF or BM25. Other parameters are the same used for ItemKNN [28].

ItemKNN-CFCBF: A hybrid CF-CBF algorithm based on item-item similarities. The similarity is computed by first concatenating, for each item i , the vector of ratings and the vector of features $[\mathbf{r}_i, \mathbf{w}\mathbf{f}_i]$ – and by later computing the cosine similarity between the concatenated vectors. Hyper-parameters are the same used for ItemKNN, plus a parameter w that weights the content features with respect to the ratings.

P³ α : A simple graph-based algorithm which implements a random walk between users and items [8]. Items for user u are ranked based on the probability of a random walk with three steps starting from user u . The probability p_{ui} to jump from user u to item i is computed from the implicit user-rating-matrix as $p_{ui} = (r_{ui}/N_u)^\alpha$, where r_{ui} is the rating of user u on item i , N_u is the number of ratings of user u and α is a damping factor. The probability p_{iu} to jump backward is computed as $p_{iu} = (r_{ui}/N_i)^\alpha$, where N_i is the number of ratings for item i . The method is equivalent to a KNN item-based CF algorithm, with the similarity matrix defined as

$$s_{ij} = \sum_v p_{jv} p_{vi} \quad (3)$$

The parameters of the method are the numbers of neighbors k and the value of α . We include this algorithm because it provides good recommendation quality at a low computational cost.

RP³ β : A version of P³ α proposed in [34]. Here, the outcomes of P³ α are modified by dividing the similarities by each item's popularity raised to the power of a coefficient β . If β is 0, the algorithm

⁴We will re-run parameter optimization for the reproduced algorithms as part of our future work in order to validate the parameter optimization procedures used by the authors. This step was, however, outside the scope of our current work.

⁵https://github.com/MaurizioFD/RecSys2019_DeepLearning_Evaluation

β が0の場合、アルゴリズムは P^3 α と等価である。そのパラメータは、近傍の数 k と α と β の値である。

すべてのベースラインアルゴリズムとデータセットについて、Scikit-Optimize⁶の実装を用いたベイズ探索[1]により最適なパラメータを決定した。各アルゴリズムについて35ケースを検討し、最初の5ケースを初期ランダムポイントに使用した。近傍サイズ k は5から800まで、縮小項 h は0から1000の間、 α と β は0から2の間の実数値をとった。

3 ベースラインに対する検証

本節では、再現性のある作品と、説明したベースライン手法を比較した結果をまとめる。詳細な統計、結果、最終的なパラメータをオンラインで共有する。

3.1 協調記憶ネットワーク(CMN)

CMN法はSIGIR '18で発表され、記憶ネットワークと神経注意メカニズムを潜在因子モデルと近傍モデルと組み合わせたものである[10]。著者らは、彼らのアプローチを評価するために、異なる行列分解とニューラル推薦アプローチ、およびItemKNNアルゴリズム(収縮なし)と比較している。評価には3つのデータセットを用いる: Epinions, CiteULike-a, Pinterest。提案手法の最適なハイパーパラメータが報告されているが、ベースラインがどのように調整されるかについての情報は提供されていない。ヒット率とNDCGは、leave-one-out手順で使用される性能指標である。報告された結果は、CMNがすべての指標において他のすべてのベースラインを上回ることを示している。

我々は彼らの実験をすべてのデータセットについて再現することができた。単純なベースラインを用いた追加実験では、ヒット率(HR@5)メトリックに対してベースラインのパラメータを最適化した。3つのデータセットに対する結果を表2に示す。

我々の分析によると、ベースラインの最適化後、CMNはどのデータセットでも単一のケースで最も性能の良い手法ではない。CiteULike-aとPinterestデータセットでは、パーソナライズされたベースライン手法のうち少なくとも2つが、どの指標においてもCMN手法を上回った。多くの場合、すべてのパーソナライズされたベースラインでさえ、CMNよりも優れていた。Epinionsデータセットでは、驚いたことに、原著論文に含まれていない非パーソナライズドTopPopularメソッドが、他のすべてのアルゴリズムよりも大きなマージンをもって優れていた。このデータセットでは、CMNは我々のベースラインよりもはるかに優れていた。したがって、約660kのオブザベーションを持つこの比較的に小さく非常に疎なデータセットでのCMNの成功は、データセットの特殊性、またはCMNの人気バイアスに結びつく可能性がある。分析の結果、Epinionsデータセットは他のデータセットよりも人気度分布が非常に不均一であることがわかった(CiteULike-aのジニ指数0.69対0.37)。このデータセットに対して、CMNはトップ n のリストでも、ベースラインが推奨するアイテムよりも平均して8%から25%人気のあるアイテムを推奨している。

表2:原著論文で報告された指標とカットオフ値を用いたCMN法の実験結果。数字は、最良の結果に対応する場合、またはベースラインがCMNを上回った場合に太字で表示される。

	CiteULike-a			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	0.1803	0.1220	0.2783	0.1535
UserKNN	0.8213	0.7033	0.8935	0.7268
ItemKNN	0.8116	0.6939	0.8878	0.7187
$P^3\alpha$	0.8202	0.7061	0.8901	0.7289
$RP^3\beta$	0.8226	0.7114	0.8941	0.7347
CMN	0.8069	0.6666	0.8910	0.6942

	Pinterest			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	0.1668	0.1066	0.2745	0.1411
UserKNN	0.6886	0.4936	0.8527	0.5470
ItemKNN	0.6966	0.4994	0.8647	0.5542
$P^3\alpha$	0.6871	0.4935	0.8449	0.5450
$RP^3\beta$	0.7018	0.5041	0.8644	0.5571
CMN	0.6872	0.4883	0.8549	0.5430

	Epinions			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	0.5429	0.4153	0.6644	0.4547
UserKNN	0.3506	0.2983	0.3922	0.3117
ItemKNN	0.3821	0.3165	0.4372	0.3343
$P^3\alpha$	0.3510	0.2989	0.3891	0.3112
$RP^3\beta$	0.3511	0.2980	0.3892	0.3103
CMN	0.4195	0.3346	0.4953	0.3592

3.2 メタパスに基づくREcommendation(MCRec)のコンテキスト

KDD '18で発表されたMCRec [17]は、top- n 推薦のために映画ジャンルのような補助情報を活用するメタパスベースのモデルである。技術的観点から、著者らは、より高品質なパスインスタンスを選択するための優先度ベースのサンプリング技術を提案し、メタパスベースのコンテキスト、ユーザー、アイテムの表現を改善するための新しい共同注意メカニズムを提案する。著者らは、3つの小さなデータセット(MovieLens100k, LastFm, Yelp)において、複雑さの異なる様々なモデルに対して、その手法の4つのバリエーションをベンチマークしている。評価は、80/20のランダムな訓練-テスト分割を作成し、そのような評価実行を10回実行することによって行われる。評価手順は再現可能であり、公開されているトレーニングとテストの分割はMovieLensデータセットに対してのみ提供された。彼らの論文で使用されたMFとNeuMF [14]のベースラインでは、アーキテクチャとハイパーパラメータは元の論文から引用された。他のベースラインでは、ハイパーパラメータのチューニングに関する情報は提供されていない。性能指標として、Precision, Recall, NDCGを用い、推薦リストは長さ10とする。しかし、NDCGの指標は、本論文では言及されていない、一般的でない、疑わしい方法で実装されている。そこで、NDCGの標準版を使用する。

⁶<https://scikit-optimize.github.io/>

⁷We report the results for CMN-3 as the version with the best results.

is equivalent to $P^3\alpha$. Its parameters are the numbers of neighbors k and the values for α and β .

For all baseline algorithms and datasets, we determined the optimal parameters via Bayesian search [1] using the implementation of Scikit-Optimize⁶. We explored 35 cases for each algorithm, where the first 5 were used for the initial random points. We considered neighborhood sizes k from 5 to 800; the shrink term h was between 0 and 1000; and α and β took real values between 0 and 2.

3 VALIDATION AGAINST BASELINES

This section summarizes the results of comparing the reproducible works with the described baseline methods. We share the detailed statistics, results, and final parameters online.

3.1 Collaborative Memory Networks (CMN)

The CMN method was presented at SIGIR '18 and combines memory networks and neural attention mechanisms with latent factor and neighborhood models [10]. To evaluate their approach, the authors compare it with different matrix factorization and neural recommendation approaches as well as with an ItemKNN algorithm (with no *shrinkage*). Three datasets are used for evaluation: Epinions, CiteULike-a, and Pinterest. Optimal hyper-parameters for the proposed method are reported, but no information is provided on how the baselines are tuned. Hit rate and NDCG are the performance measures used in a leave-one-out procedure. The reported results show that CMNs outperform all other baselines on all measures.

We were able to reproduce their experiments for all their datasets. For our additional experiments with the simple baselines, we optimized the parameters of our baselines for the hit rate (HR@5) metric. The results for the three datasets are shown in Table 2.

Our analysis shows that, after optimization of the baselines, CMN⁷ is in no single case the best-performing method on any of the datasets. For the CiteULike-a and Pinterest datasets, at least two of the personalized baseline techniques outperformed the CMN method on any measure. Often, even all personalized baselines were better than CMN. For the Epinions dataset, to some surprise, the unpersonalized TopPopular method, which was not included in the original paper, was better than all other algorithms by a large margin. On this dataset, CMN was indeed much better than our baselines. The success of CMN on this comparably small and very sparse dataset with about 660k observations could therefore be tied to the particularities of the dataset or to a popularity bias of CMN. An analysis reveals that the Epinions dataset has indeed a much more uneven popularity distribution than the other datasets (Gini index of 0.69 vs. 0.37 for CiteULike-a). For this dataset, CMN also recommends in its top-n lists items that are, on average, 8% to 25% more popular than the items recommended by our baselines.

Table 2: Experimental results for the CMN method using the metrics and cutoffs reported in the original paper. Numbers are printed in bold when they correspond to the best result or when a baseline outperformed CMN.

	CiteULike-a			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	0.1803	0.1220	0.2783	0.1535
UserKNN	0.8213	0.7033	0.8935	0.7268
ItemKNN	0.8116	0.6939	0.8878	0.7187
$P^3\alpha$	0.8202	0.7061	0.8901	0.7289
$RP^3\beta$	0.8226	0.7114	0.8941	0.7347
CMN	0.8069	0.6666	0.8910	0.6942
	Pinterest			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	0.1668	0.1066	0.2745	0.1411
UserKNN	0.6886	0.4936	0.8527	0.5470
ItemKNN	0.6966	0.4994	0.8647	0.5542
$P^3\alpha$	0.6871	0.4935	0.8449	0.5450
$RP^3\beta$	0.7018	0.5041	0.8644	0.5571
CMN	0.6872	0.4883	0.8549	0.5430
	Epinions			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	0.5429	0.4153	0.6644	0.4547
UserKNN	0.3506	0.2983	0.3922	0.3117
ItemKNN	0.3821	0.3165	0.4372	0.3343
$P^3\alpha$	0.3510	0.2989	0.3891	0.3112
$RP^3\beta$	0.3511	0.2980	0.3892	0.3103
CMN	0.4195	0.3346	0.4953	0.3592

3.2 Metapath based Context for REcommendation (MCRec)

MCRec [17], presented at KDD '18, is a meta-path based model that leverages auxiliary information like movie genres for top-n recommendation. From a technical perspective, the authors propose a priority-based sampling technique to select higher-quality path instances and propose a novel co-attention mechanism to improve the representations of meta-path based context, users, and items.

The authors benchmark four variants of their method against a variety of models of different complexity on three small datasets (MovieLens100k, LastFm, and Yelp). The evaluation is done by creating 80/20 random training-test splits and by executing 10 of such evaluation runs. The evaluation procedure could be reproduced; public training-test splits were provided only for the MovieLens dataset. For the *MF* and *NeuMF* [14] baselines used in their paper, the architecture and hyper-parameters were taken from the original papers; no information about hyper-parameter tuning is provided for the other baselines. Precision, Recall, and the NDCG are used as performance measures, with a recommendation list of length 10. The NDCG measure is however implemented in an uncommon and questionable way, which is not mentioned in the paper. Here, we therefore use a standard version of the NDCG.

⁶<https://scikit-optimize.github.io/>

⁷We report the results for CMN-3 as the version with the best results.

公開ソフトウェアでは、MovieLensのメタパスはハードコードされており、メタパスの前処理や構築のためのコードは提供されていない。ここでは、MovieLens データセットの結果のみを詳細に示す。我々は、[17]で明らかに行われたように、Precisionのためにベースラインを最適化した。MCRecについては、完全モデルの結果を報告する。

表3:MCRecと我々のベースラインの比較(MovieLens100k)

	PREC@10	REC@10	NDCG@10
TopPopular	0.1907	0.1180	0.1361
UserKNN	0.2913	0.1802	0.2055
ItemKNN	0.3327	0.2199	0.2603
$P^3\alpha$	0.2137	0.1585	0.1838
$RP^3\beta$	0.2357	0.1684	0.1923
MCRec	0.3077	0.2061	0.2363

表3は、従来のItemKNN法が正しく構成された場合、すべての性能指標でMCRecを上回ることが示している。本論文では、一般的でないNDCG尺度の使用以外にも、方法論的な問題点があることがわかった。MFとNeuMFのベースラインのハイパーパラメータは、前述のように、与えられたデータセットに対して最適化されたものではないが、原著論文[17]から引用した。さらに、提供されたソースコードを見ると、著者らはテストセットで選択された異なるエポックにわたって、各メトリックに対する手法の最良の結果を報告していることがわかる。

3.3 協調変分オートエンコーダ(CVAE)

KDD '18で発表されたCVAE法[23]は、コンテンツと評価情報の両方を考慮するハイブリッド技法である。このモデルは、教師無しでコンテンツデータから深い潜在表現を学習し、また、コンテンツと評価の両方からアイテムとユーザーの間の暗黙の関係を学習する。

本手法は、比較的に小さな2つのCiteULikeデータセット(135kと205kのインタラクション)で評価される。両データセットとも、スパース版とデンス版をテストした。23]のベースラインには、3つの最近の深層学習モデルと、Collaborative Topic Regression (CTR)が含まれる。各手法のパラメータは、検証セットに基づいて調整される。異なるリスト長(50から300)でのRecallが評価指標として使用される。ランダムな訓練とテストのデータ分割が適用され、測定が5回繰り返される。

彼らのコードと評価手順を使って、彼らの結果を再現することができた。データセットは著者らも共有している。ベースラインを微調整した結果、[47]の高密度CiteULike-aデータセットに対して表4に示すような結果が得られた。最短リスト長50の場合、純粋なCFベースラインの大部分でさえ、このデータセットでCVAE法を上回った。より長いリスト長では、ハイブリッドItemKNN-CFCBF法が最良の結果をもたらした。同様の結果が、疎なCiteULike-tデータセットでも得られた。一般に、リスト長50では、ItemKNN-CFCBFは一貫してCVAEを上回った

⁸In our evaluations, we did not use this form of measurement.

表4:CVAE(CiteULike-a)の実験結果。

	REC@50	REC@100	REC@300
TopPopular	0.0044	0.0081	0.0258
UserKNN	0.0683	0.1016	0.1685
ItemKNN	0.0788	0.1153	0.1823
$P^3\alpha$	0.0788	0.1151	0.1784
$RP^3\beta$	0.0811	0.1184	0.1799
ItemKNN-CFCBF	0.1837	0.2777	0.4486
CVAE	0.0772	0.1548	0.3602

をテストした。より長いリスト長(100以上)においてのみ、CVAEは2つのデータセットで我々の手法を上回ることができた。全体として、CVAEは特定の構成においてのみベースラインより有利であり、比較的に長く、むしろ珍しい推奨カットオフ閾値においてのみ有利であった。しかし、このような長いリストサイズの使用は、この論文では正当化されなかった。

3.4 協調深層学習(CDL)

議論されたCVAE法は、KDD '15の先行するよく引用されるCDL法[48]をベースラインの1つとして考慮し、著者らも同じ評価手順とCiteULikeデータセットを使用する。CDLは、積層ノイズ除去オートエンコーダ(SDAE)と協調フィルタリングの共同学習のための確率的フィードフォワードモデルである。ディープラーニング技術を適用し、コンテンツ情報と協調情報の深い表現を共同で学習する。48]のCDLの評価では、特にスパースデータ状況において、広く参照されているCTR法[47]と比較して有利であることが示された。

表5:密なCiteULike-aデータセットに対するCDLの実験結果。

	REC@50	REC@100	REC@300
TopPopular	0.0038	0.0073	0.0258
UserKNN	0.0685	0.1028	0.1710
ItemKNN	0.0846	0.1213	0.1861
$P^3\alpha$	0.0718	0.1079	0.1777
$RP^3\beta$	0.0800	0.1167	0.1815
ItemKNN-CBF	0.2135	0.3038	0.4707
ItemKNN-CFCBF	0.1945	0.2896	0.4620
CDL	0.0543	0.1035	0.2627

我々は[48]の研究を再現し、密なCiteULike-aデータセットに対する表5に示す結果を得た。当然のことながら、前節のCVAEよりも優れたベースラインはCDLよりも優れており、短いリスト長でも、すでに純粋なCF手法はハイブリッドCDLアプローチよりも優れていた。しかし、ここでもCDLは、4つのデータセット構成のうち2つにおいて、リスト長が100を超えると高いRecallをもたらす。CVAEとCDLの詳細な結果を比較すると、新しいCVAE法はCDLよりも確かに常に優れていることがわかり、これは進歩があったことを示している。しかし、どちらの方法も、大半のケースで単純なベースラインの1つより優れているわけではない。

In the publicly shared software, the meta-paths are hard-coded for MovieLens, and no code for preprocessing and constructing the meta-paths is provided. Here, we therefore only provide the results for the MovieLens dataset in detail. We optimized our baselines for Precision, as was apparently done in [17]. For MCRc, the results for the complete model are reported.

Table 3: Comparing MCRc against our baselines (MovieLens100k)

	PREC@10	REC@10	NDCG@10
TopPopular	0.1907	0.1180	0.1361
UserKNN	0.2913	0.1802	0.2055
ItemKNN	0.3327	0.2199	0.2603
$P^3\alpha$	0.2137	0.1585	0.1838
$RP^3\beta$	0.2357	0.1684	0.1923
MCRc	0.3077	0.2061	0.2363

Table 3 shows that the traditional ItemKNN method, when configured correctly, outperforms MCRc on all performance measures.

Besides the use of an uncommon NDCG measure, we found other potential methodological issues in this paper. Hyper-parameters for the *MF* and *NeuMF* baselines were, as mentioned, not optimized for the given datasets but taken from the original paper [17]. In addition, looking at the provided source code, it can be seen that the authors report the best results of their method for each metric across different epochs chosen on the test set, which is inappropriate.⁸

3.3 Collaborative Variational Autoencoder (CVAE)

The CVAE method [23], presented at KDD '18, is a hybrid technique that considers both content as well as rating information. The model learns deep latent representations from content data in an unsupervised manner and also learns implicit relationships between items and users from both content and ratings.

The method is evaluated on two comparably small CiteULike datasets (135k and 205k interactions). For both datasets, a sparse and a dense version is tested. The baselines in [23] include three recent deep learning models and as well as Collaborative Topic Regression (CTR). The parameters for each method are tuned based on a validation set. Recall at different list lengths (50 to 300) is used as an evaluation measure. Random train-test data splitting is applied and the measurements are repeated five times.

We could reproduce their results using their code and evaluation procedure. The datasets are also shared by the authors. Fine-tuning our baselines led to the results shown in Table 4 for the dense CiteULike-a dataset from [47]. For the shortest list length of 50, even the majority of the pure CF baselines outperformed the CVAE method on this dataset. At longer list lengths, the hybrid *ItemKNN-CFCBF* method led to the best results. Similar results were obtained for the sparse *CiteULike-t* dataset. Generally, at list length 50, *ItemKNN-CFCBF* was consistently outperforming CVAE

Table 4: Experimental results for CVAE (CiteULike-a).

	REC@50	REC@100	REC@300
TopPopular	0.0044	0.0081	0.0258
UserKNN	0.0683	0.1016	0.1685
ItemKNN	0.0788	0.1153	0.1823
$P^3\alpha$	0.0788	0.1151	0.1784
$RP^3\beta$	0.0811	0.1184	0.1799
ItemKNN-CFCBF	0.1837	0.2777	0.4486
CVAE	0.0772	0.1548	0.3602

in all tested configurations. Only at longer list lengths (100 and beyond), CVAE was able to outperform our methods on two datasets.

Overall, CVAE was only favorable over the baselines in certain configurations and at comparably long and rather uncommon recommendation cutoff thresholds. The use of such long list sizes was however not justified in the paper.

3.4 Collaborative Deep Learning (CDL)

The discussed CVAE method considers the earlier and often-cited CDL method [48] from KDD '15 as one of their baselines, and the authors also use the same evaluation procedure and CiteULike datasets. CDL is a probabilistic feed-forward model for joint learning of stacked denoising autoencoders (SDAE) and collaborative filtering. It applies deep learning techniques to jointly learn a deep representation of content information and collaborative information. The evaluation of CDL in [48] showed that it is favorable in particular compared to the widely referenced CTR method [47], especially in sparse data situations.

Table 5: Experimental results for CDL on the dense CiteULike-a dataset.

	REC@50	REC@100	REC@300
TopPopular	0.0038	0.0073	0.0258
UserKNN	0.0685	0.1028	0.1710
ItemKNN	0.0846	0.1213	0.1861
$P^3\alpha$	0.0718	0.1079	0.1777
$RP^3\beta$	0.0800	0.1167	0.1815
ItemKNN-CBF	0.2135	0.3038	0.4707
ItemKNN-CFCBF	0.1945	0.2896	0.4620
CDL	0.0543	0.1035	0.2627

We reproduced the research in [48], leading to the results shown in Table 5 for the dense *CiteULike-a* dataset. Not surprisingly, the baselines that were better than CVAE in the previous section are also better than CDL, and again for short list lengths, already the pure CF methods were better than the hybrid CDL approach. Again, however, CDL leads to higher Recall for list lengths beyond 100 in two out of four dataset configurations. Comparing the detailed results for CVAE and CDL, we see that the newer CVAE method is indeed always better than CDL, which indicates that progress was made. Both methods, however, are not better than one of the simple baselines in the majority of the cases.

⁸In our evaluations, we did not use this form of measurement.

3.5 ニューラル協調フィルタリング(NCF)

WWW '17で発表されたニューラルネットワークベースの協調フィルタリング[14]は、内積をデータから任意の関数を学習できるニューラルアーキテクチャに置き換えることで、行列因子分解を一般化したものである。提案するハイブリッド手法(NeuMF)を、それぞれ100万と150万の相互作用を含む2つのデータセット(MovieLens1MとPinterest)で評価した。評価にはleave-one out手順を使用し、元のデータ分割は著者によって公開されている。その結果、NeuMFは、ヒット率とNDCGを、最大10までの異なるリスト長を使用した評価尺度として使用した場合、例えば、既存の行列分解モデルよりも有利であることが示された。パラメータの最適化は、トレーニングセットから作成された検証セットで行われる。上記のMCRecの実装と同様に、提供されたソースコードは、著者がテストセットで得られた結果に基づいてエポック数を選択したことを示している。しかし、エポック数は調整すべきパラメータであり、テストセットに基づいて決定されるべきではないので、検証セットでこのパラメータを見つける、より適切な実装を使用する。ItemKNN法では、著者らは近傍サイズを変化させただけで、他の変化はテストしていない。

表6:NCFの実験結果。

	Pinterest			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	0.1663	0.1065	0.2744	0.1412
UserKNN	0.7001	0.5033	0.8610	0.5557
ItemKNN	0.7100	0.5092	0.8744	0.5629
P ³ _α	0.7008	0.5018	0.8667	0.5559
RP ³ _β	0.7105	0.5116	0.8740	0.5650
NeuMF	0.7024	0.4983	0.8719	0.5536
	Movielens 1M			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	0.3043	0.2062	0.4531	0.2542
UserKNN	0.4916	0.3328	0.6705	0.3908
ItemKNN	0.4829	0.3328	0.6596	0.3900
P ³ _α	0.4811	0.3331	0.6464	0.3867
RP ³ _β	0.4922	0.3409	0.6715	0.3991
NeuMF	0.5486	0.3840	0.7120	0.4369
SLIM	0.5589	0.3961	0.7161	0.4470

公開されている情報があれば、[14]の結果を再現することができる。実験の結果を表6に示す。Pinterestデータセットでは、パーソナライズされたベースラインのうち2つが、すべてのメトリクスでNeuMFよりも優れていた。MovieLensデータセットでは、NeuMFは我々の単純なベースラインを非常に明確に上回った。MovieLensデータセットは、新しいモデルの評価に過去数十年にわたって広く使用されてきたため、[33]に記述されている単純な線形手法であるSLIMで追加実験を行った。SLIMを実装するために、Pythonのscikit-learnパッケージ(ElasticNet)で提供されている標準的なElastic Netの実装を使用した。検証セットでハイパーパラメータを調整するために、

他のベースラインと同様に近傍サイズを考慮した;11と12の正則化の比率は 10^{-5} から1.0;正則化の大きさの係数は 10^{-3} から1.0である。表6から、SLIMは予想通り我々のベースラインよりも確かに優れているが、このデータセットでもNeuMFを上回っていることがわかる。

3.6 スペクトル協調フィルタリング(SpectralCF)

SpectralCF [53]はRecSys '18で発表されたもので、コールドスタート問題に特化して設計されたもので、スペクトルグラフ理論の概念に基づいている。その推薦文は、二分割されたユーザーとアイテムの関係グラフと、スペクトル領域で直接協調推薦を行うために使用される新しい畳み込み演算に基づいている。この手法は3つの公開データセット(MovieLens1M, HetRec, Amazon Instant Video)で評価され、最近のニューラルアプローチや確立された因数分解・ランキング技術を含む様々な手法に対してベンチマークされた。評価は、ランダムに作成された80/20のトレーニングテスト分割に基づき、異なるカットオフでRecallと平均平均精度(MAP)を使用した⁹。

MovieLensデータセットについては、著者が使用したトレーニングデータセットとテストデータセットをコードとともに共有した。他のデータセットについては、データ分割は公開されていないため、論文の説明に従って、我々自身が分割を作成した。驚くべきことに、著者らは論文中のハイパーパラメータ値のセットを1つだけ報告しており、それは明らかにすべてのデータセットに使用されている。そこで、提供されたハイパーパラメータと、すべてのデータセットで独自に決定したハイパーパラメータ設定の両方でコードを実行した。HetRecとAmazon Instant Videoデータセットでは、TopPopular法も含めた我々の驚きのベースラインはすべて、すべての指標でSpectralCFを上回った。しかし、提供されたMovieLensのデータ分割でコードを実行すると、SpectralCFがすべてのベースラインよりも大きなマージンで優れていることがわかった。Recall@20は、例えば、我々の最良のベースラインより50%高かった。

そこで、MovieLensデータセットについて公表されている訓練とテストの分割を分析したところ、テストセットのアイテムの人気度分布は、ランダムサンプリング手順から生じられる分布とは大きく異なることが観察された¹⁰。次に、MovieLensデータセットについても、論文に記載されている分割手順を用いて、独自の訓練とテストの分割で実験を行った。公正な比較を確実にするため、データ分割のパラメータを最適化した。実験結果を表7に示す。原著論文で説明したように作成されたデータ分割を使用した場合、MovieLensデータセットの結果は、他の2つのデータセットに対する我々の実験と一致している。すなわち、全ての構成におけるSpectralCFは、我々のベースライン手法よりも性能が悪く、TopPopular手法でも上回った。

図1はデータ分割問題を可視化したものである。青いデータ点は、トレーニングセットの各アイテムの正規化された人気値を示し、対応する分割の最も人気のあるアイテムは、人気値の小さい順に値1を持つ。評価のランダムサンプリングの場合、テストセットからのオレンジ色の点は、対応する青色の点に非常に近いことがほとんどである。

⁹To assess the cold-start behavior, additional experiments are performed with fewer data points per user in the training set.

¹⁰We contacted the authors on this issue, but did not receive an explanation for this phenomenon.

3.5 Neural Collaborative Filtering (NCF)

Neural network-based Collaborative Filtering [14], presented at WWW '17, generalizes Matrix Factorization by replacing the inner product with a neural architecture that can learn an arbitrary function from the data. The proposed hybrid method (NeuMF) was evaluated on two datasets (MovieLens1M and Pinterest), containing 1 million and 1.5 million interactions, respectively. A leave-one out procedure is used in the evaluation and the original data splits are publicly shared by the authors. Their results show that NeuMF is favorable, e.g., over existing matrix factorization models, when using the hit rate and the NDCG as an evaluation measure using different list lengths up to 10.

Parameter optimization is done on a validation set created from the training set. Similar to the implementation of *MCR* above, the provided source code shows that the authors chose the number of epochs based on the results obtained for the test set. Since the number of epochs, however, is a parameter to tune and should not be determined based on the test set, we use a more appropriate implementation that finds this parameter with the validation set. For the ItemKNN method, the authors only varied the neighborhood sizes but did not test other variations.

Table 6: Experimental results for NCF.

	Pinterest			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	0.1663	0.1065	0.2744	0.1412
UserKNN	0.7001	0.5033	0.8610	0.5557
ItemKNN	0.7100	0.5092	0.8744	0.5629
$P^3\alpha$	0.7008	0.5018	0.8667	0.5559
$RP^3\beta$	0.7105	0.5116	0.8740	0.5650
NeuMF	0.7024	0.4983	0.8719	0.5536
	Movielens 1M			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	0.3043	0.2062	0.4531	0.2542
UserKNN	0.4916	0.3328	0.6705	0.3908
ItemKNN	0.4829	0.3328	0.6596	0.3900
$P^3\alpha$	0.4811	0.3331	0.6464	0.3867
$RP^3\beta$	0.4922	0.3409	0.6715	0.3991
NeuMF	0.5486	0.3840	0.7120	0.4369
SLIM	0.5589	0.3961	0.7161	0.4470

Given the publicly shared information, we could reproduce the results from [14]. The outcomes of the experiment are shown in Table 6. On the Pinterest dataset, two of the personalized baselines were better than NeuMF on all metrics. For the MovieLens dataset, NeuMF outperformed our simple baselines quite clearly.

Since the MovieLens dataset has been extensively used over the last decades for evaluating new models, we made additional experiments with *SLIM*, a simple linear method described in [33]. To implement *SLIM*, we took the standard Elastic Net implementation provided in the *scikit-learn* package for Python (ElasticNet). To tune the hyper-parameters on the validation set, we considered neighborhood sizes as in the other baselines; the ratio of l1 and l2

regularization between 10^{-5} and 1.0; and the regularization magnitude coefficient between 10^{-3} and 1.0. Table 6 shows that *SLIM* is indeed better than our baselines, as expected, but also outperforms NeuMF on this dataset.

3.6 Spectral Collaborative Filtering (SpectralCF)

SpectralCF [53], presented at RecSys '18, was designed to specifically address the cold-start problem and is based on concepts of Spectral Graph Theory. Its recommendations are based on the bipartite user-item relationship graph and a novel convolution operation, which is used to make collaborative recommendations directly in the *spectral domain*. The method was evaluated on three public datasets (MovieLens1M, HetRec, and Amazon Instant Video) and benchmarked against a variety of methods, including recent neural approaches and established factorization and ranking techniques. The evaluation was based on randomly created 80/20 training-test splits and using Recall and the Mean Average Precision (MAP) at different cutoffs.⁹

For the MovieLens dataset, the training and test datasets used by the authors were shared along with the code. For the other datasets, the data splits were not published therefore we created the splits by ourself following the descriptions in the paper.

Somehow surprisingly, the authors report only one set of hyper-parameter values in the paper, which they apparently used for all datasets. We therefore ran the code both with the provided hyper-parameters and with hyper-parameter settings that we determined by our own on all datasets. For the HetRec and Amazon Instant Video datasets, all our baselines, to our surprise also including the TopPopular method, outperformed SpectralCF on all measures. However, when running the code on the provided MovieLens data splits, we found that SpectralCF was better than all our baselines by a huge margin. Recall@20 was, for example, 50% higher than our best baseline.

We therefore analyzed the published train-test split for the MovieLens dataset and observed that the popularity distribution of the items in the test set is very different from a distribution that would likely result from a random sampling procedure.¹⁰ We then ran experiments with our own train-test splits also for the MovieLens dataset, using the splitting procedure described in the paper. We optimized the parameters for our data split to ensure a fair comparison. The results of the experiment are shown in Table 7. When using data splits that were created as described in the original paper, the results for the MovieLens dataset are in line with our own experiments for the other two datasets, i.e., SpectralCF in all configurations performed worse than our baseline methods and was outperformed even by the TopPopular method.

Figure 1 visualizes the data splitting problem. The blue data points show the normalized popularity values for each item in the training set, with the most popular item in the corresponding split having the value 1, ordered by decreasing popularity values. In case of random sampling of ratings, the orange points from the

⁹To assess the cold-start behavior, additional experiments are performed with fewer data points per user in the training set.

¹⁰We contacted the authors on this issue, but did not receive an explanation for this phenomenon.

表7:SpectralCFの実験結果(MovieLens1M、独自のランダム分割と5回の繰り返し測定を使用)。

	Cutoff 20		Cutoff 60		Cutoff 100	
	REC	MAP	REC	MAP	REC	MAP
TopPopular	0.1853	0.0576	0.3335	0.0659	0.4244	0.0696
UserKNN CF	0.2881	0.1106	0.4780	0.1238	0.5790	0.1290
ItemKNN CF	0.2819	0.1059	0.4712	0.1190	0.5737	0.1243
$P^3\alpha$	0.2853	0.1051	0.4808	0.1195	0.5760	0.1248
$RP^3\beta$	0.2910	0.1088	0.4882	0.1233	0.5884	0.1288
SpectralCF	0.1843	0.0539	0.3274	0.0618	0.4254	0.0656

しかし、テストセットの多くのアイテムの人気値は大きく異なることがわかる。ジニ指数やシャノンエントロピーのような尺度で分布を分析すると、共有テストセットのデータセット特性がランダムな分割から大きく乖離していることが確認できる。真のランダム分割のジニ指数は、訓練分割とテスト分割の両方で約0.79である。提供された訓練分割のジニ指数は我々のものと同様であるが、提供されたテスト分割のジニ指数ははるかに高く(0.92)、これは分布がランダムな分割よりもはるかに高い人気バイアスを持つことを意味する。

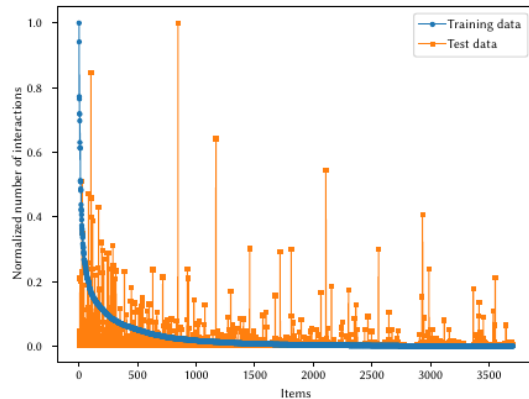


図1:提供されたトレーニングとテストの分割の人気度分布。ランダムな分割の場合、正規化された値は平均して両方の分割で近いはずである。

3.7 協調フィルタリングのための変分オートエンコーダ(Mult-VAE)

Mult-VAE[24]は変分オートエンコーダに基づく暗黙のフィードバックのための協調フィルタリング手法である。この研究はWWW'18で発表された。Mult-VAEにより、著者らは多項尤度を持つ生成モデルを導入し、学習目的に対して異なる正則化パラメータを提案し、パラメータ推定にベイズ推論を用いる。彼らは、もともと映画の評価や曲の再生回数を含む3つの二値化されたデータセットで、彼らの方法を評価する。実験のベースラインには、2008年の行列分解法[18]、2011年の線形モデル[33]、そして最近のニューラル法[51]の両方が含まれる。

報告された実験によると、提案手法は、RecallとNDCGの点で、最良のベースラインよりも一般的に約3%良い精度結果を導く。

提案手法は、そのコードとデータセットを用いて、非常に単純なベースライン手法を常に凌駕していることがわかった。得られた精度の結果は、我々の最良のベースラインよりも10%から20%良好であった。このように、Mult-VAEでは、より複雑な方法が、すべての構成において、我々のどのベースライン技法よりも、大きなマージンをもって、優れている例を、検討された文献から1つ発見した。

Mult-VAEが複雑な非ニューラルモデルよりも有利であることを検証するために、[24]で報告されているように、MovieLensとNetflixデータセットに対して、重み付き行列分解技法[18]と線形モデル[33](Elastic Netを用いたSLIM)のパラメータを我々自身で最適化した。我々は以下の観察を行った。両データセットとも、結果を再現することができ、原著論文で報告された様々な尺度において、SLIMに対して最大5%の改善が観察された。表8は、[24]と同様にNDCG@100で最適化した後、オリジナル実験の測定値とカットオフ値を使用したNetflixデータセットの結果を示している。

表8:Mult-VAE(Netflixデータ)の実験結果(原著論文で報告された指標とカットオフを使用)。

	REC@20	REC@50	NDCG@100
TopPop	0.0782	0.1643	0.1570
ItemKNN CF	0.2088	0.3386	0.3086
$P^3\alpha$	0.1977	0.3346	0.2967
$RP^3\beta$	0.2196	0.3560	0.3246
SLIM	0.2551	0.3995	0.3745
Mult-VAE	0.2626	0.4138	0.3756

最適化目標であるNDCGの観点から、Mult-VAEとSLIMの差は非常に小さい。しかし、Recallの観点からは、Mult-VAEはSLIMより優れているように見える。使用したカットオフ値(Recallは20と50、NDCGは100)の選択は[24]ではあまり一貫していないため、異なるカットオフ長で追加測定を行った。結果を表9に示す。NDCGを最適化目標として、また性能指標として使用した場合、このデータセットではSLIMとMult-VAEの差はなくなり、SLIMの方が若干良くなることがあることがわかる。同様の現象はMovieLensデータセットでも観察できる。したがって、この特殊なケースでは、ニューラル・アプローチによって達成される進歩は部分的なものに過ぎず、選択された評価尺度に依存する。

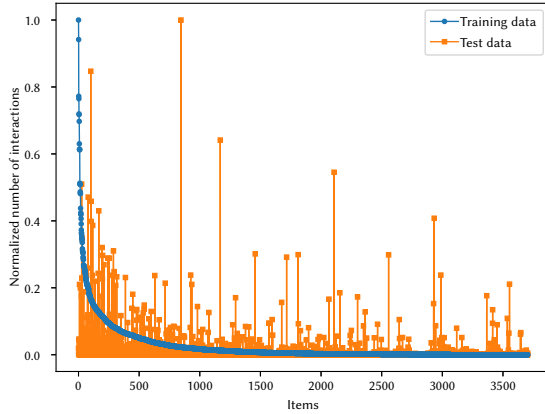
表9:Netflixデータセットに対するカットオフ長を追加したMult-VAEの実験結果。

	NDCG@20	NDCG@50	REC@100	NDCG@100
SLIM	0.2473	0.3196	0.5289	0.3745
Mult-VAE	0.2448	0.3192	0.5476	0.3756

Table 7: Experimental results for SpectralCF (MovieLens1M, using own random splits and five repeated measurements).

	Cutoff 20		Cutoff 60		Cutoff 100	
	REC	MAP	REC	MAP	REC	MAP
TopPopular	0.1853	0.0576	0.3335	0.0659	0.4244	0.0696
UserKNN CF	0.2881	0.1106	0.4780	0.1238	0.5790	0.1290
ItemKNN CF	0.2819	0.1059	0.4712	0.1190	0.5737	0.1243
$P^3\alpha$	0.2853	0.1051	0.4808	0.1195	0.5760	0.1248
$RP^3\beta$	0.2910	0.1088	0.4882	0.1233	0.5884	0.1288
SpectralCF	0.1843	0.0539	0.3274	0.0618	0.4254	0.0656

test set would mostly be very close to the corresponding blue ones. Here, however, we see that the popularity values of many items in the test set differ largely. An analysis of the distributions with measures like the Gini index or Shannon entropy confirms that the dataset characteristics of the shared test set diverge largely from a random split. The Gini index of a true random split lies at around 0.79 for both the training and test split. While the Gini index for the provided training split is similar to ours, the Gini index of the provided test split is much higher (0.92), which means that the distribution has a much higher popularity bias than a random split.

**Figure 1: Popularity distributions of the provided training and test splits. In case of a random split, the normalized values should, on average, be close for both splits.**

3.7 Variational Autoencoders for Collaborative Filtering (Mult-VAE)

Mult-VAE [24] is a collaborative filtering method for implicit feedback based on variational autoencoders. The work was presented at WWW '18. With Mult-VAE, the authors introduce a generative model with multinomial likelihood, propose a different regularization parameter for the learning objective, and use Bayesian inference for parameter estimation. They evaluate their method on three binarized datasets that originally contain movie ratings or song play counts. The baselines in the experiments include both a matrix factorization method from 2008 [18], a linear model from 2011 [33], and a more recent neural method [51]. According to the

reported experiments, the proposed method leads to accuracy results that are typically around 3% better than the best baseline in terms of Recall and the NDCG.

Using their code and datasets, we found that the proposed method indeed consistently outperforms our quite simple baseline techniques. The obtained accuracy results were between 10% and 20% better than our best baseline. Thus, with Mult-VAE, we found one example in the examined literature where a more complex method was better, by a large margin, than any of our baseline techniques in all configurations.

To validate that Mult-VAE is advantageous over the complex non-neural models, as reported in [24], we optimized the parameters for the weighted matrix factorization technique [18] and the linear model [33] (SLIM using Elastic Net) for the MovieLens and Netflix datasets by ourselves. We made the following observations. For both datasets, we could reproduce the results and observe improvements over SLIM of up to 5% on the different measures reported in the original papers. Table 8 shows the outcomes for the Netflix datasets using the measurements and cutoffs from the original experiments after optimizing for NDCG@100 as in [24].

Table 8: Experimental results for Mult-VAE (Netflix data), using metrics and cutoffs reported in the original paper.

	REC@20	REC@50	NDCG@100
TopPop	0.0782	0.1643	0.1570
ItemKNN CF	0.2088	0.3386	0.3086
$P^3\alpha$	0.1977	0.3346	0.2967
$RP^3\beta$	0.2196	0.3560	0.3246
SLIM	0.2551	0.3995	0.3745
Mult-VAE	0.2626	0.4138	0.3756

The differences between Mult-VAE and SLIM in terms of the NDCG, the optimization goal, are quite small. In terms of the Recall, however, Mult-VAE improvements over SLIM seem solid. Since the choice of the used cutoffs (20 and 50 for Recall, and 100 for NDCG) is not very consistent in [24], we made additional measurements at different cutoff lengths. The results are provided in Table 9. They show that when using the NDCG as an optimization goal *and* as a performance measure, the differences between SLIM and Mult-VAE disappear on this dataset, and SLIM is actually sometimes slightly better. A similar phenomenon can be observed for the MovieLens dataset. In this particular case, therefore, the progress that is achieved through the neural approach is only partial and depends on the chosen evaluation measure.

Table 9: Experimental results for Mult-VAE using additional cutoff lengths for the Netflix dataset.

	NDCG@20	NDCG@50	REC@100	NDCG@100
SLIM	0.2473	0.3196	0.5289	0.3745
Mult-VAE	0.2448	0.3192	0.5476	0.3756

4 DISCUSSION

4.1 再現性とスケーラビリティ

ある意味で、応用機械学習における再現性の確立は、他の科学分野やコンピュータサイエンスの他のサブフィールドよりもはるかに容易であるはずである。多くの推薦アルゴリズムは完全な決定論的アルゴリズムではないが、例えば、パラメータのランダムな初期化を何らかの形で使用しているため、全く同じ実験構成を数回繰り返したときの結果のばらつきは、ほとんどの場合、おそらく非常に小さい。したがって、研究者がコードと使用したデータを提供する場合、誰もが多かれ少なかれまったく同じ結果を再現できるはずである。今日の研究者が、一般に入手可能なソフトウェアや学術機関から提供されるソフトウェアに依存することが多いことを考えると、技術的要件に関する障壁もほとんど低い。特に、仮想化技術は、他の研究者が非常によく似た条件下で実験を繰り返しやすくするはずである。

それにもかかわらず、我々の研究は、再現性のレベルが実際に高くないことを示している。コアとなるアルゴリズムのコードは、過去よりも研究者によって共有されることが多かったようであるが、これはおそらく、再現性が学会の評価基準となっていることが原因であろう。しかし、多くの場合、ハイパーパラメータの最適化、評価、データ前処理、ベースラインに使用されるコードは共有されない。このため、報告された知見を検証することは、他の研究者にとって困難である。

再現性を困難にする可能性のある直交要因の1つは、提案された手法の多くの計算複雑性である。Netflix Prizeとその1億のレーティングデータセットから10年後、研究者は2019年に、数十万のレーティングを含むデータセットを一般的に使用する。数年前には許容できないほど小さいと考えられていたこのような小さなデータセットであっても、研究者がGPUコンピューティングにアクセスできる場合でも、ハイパーパラメータの最適化には数日から数週間かかることがある。明らかに、本論文で議論されているような最近傍法は、スケーラビリティの問題にもつながる可能性がある。しかし、適切なデータ前処理とデータサンプリングメカニズムにより、学術的・産業的環境の両方において、このような手法のスケーラビリティを確保することもできる[19, 26]。

4.2 進捗状況の評価

計算量が多いにもかかわらず、最近提案されたいくつかのニューラル手法は、概念的にも計算的にも、より単純な、時には古くから知られているアルゴリズムをも凌駕しないことが、我々の分析によって示された。したがって、ニューラル手法の分野で達成される進捗のレベルは、少なくとも我々の論文で議論されているアプローチを考慮すると、不明確である。この幻の進捗の主な理由の一つは、我々の研究が示すように、ベースラインの選択と、ベースラインの適切な最適化の欠如にある。調査したケースの大半において、考慮したベースラインの最適化について十分な情報が示されていない。また、データ分割や特定の評価指標やプロトコルの実装に関して間違いを犯すことがあった。もう一つの興味深い観察は、最近の多くの論文が、最先端のベースラインの1つとして、ニューラル協調フィルタリング法(NCF)[14]を使用していることである。しかし、我々の分析によると、この方法は、あるデータセットでは単純なベースラインよりも優れており、別のデータセットではあまり良い結果をもたらさない。

線形回帰法の標準的な実装でも優れている。したがって、複雑なニューラル・モデルと他のニューラル・モデルを比較することで、進歩が主張されることが多い。セッションベースの推薦の分野でも同様の観察が可能であり、リカレントニューラルネットワークに基づく最近の手法[16]は、ほとんどの場合、ほとんど些細な手法の方が優れているにもかかわらず、競争力のあるベースラインと考えられている[29, 30]。

この分野の進歩を評価することを難しくしているもう一つの側面は、研究者が使用するデータセット、評価プロトコル、測定基準、ベースラインの多様性にある。例えば、データセットに関しては、20以上の公開データセットに加え、MovieLensとYelpデータセットのいくつかのバリエーションが見つかった。その結果、ほとんどのデータセットは1つか2つの論文でしか使われていない。あらゆる種類の評価指標(例: Precision, Recall, Mean Average Precision, NDCG, MRRなど)と、様々な評価手順(例: ランダムホールドアウト80/20, leave-last-out, leave-one-out, 100ネガティブ項目、50ネガティブ項目)が使用される。しかし、ほとんどの場合、これらの選択は、他の人が以前から使っていたという事実以上には正当化されない。現実には、メトリックの選択はアプリケーションのコンテキストに依存すべきである。アプリケーションによっては、例えば、少なくとも1つの関連項目が推薦の上位にあることが重要であり、これはMRRのようなランクベースのメトリクスの使用を示唆する。他の領域では、できるだけ多くの関連項目をユーザに見せることが目的である場合、高いRecallがより重要になるかもしれない。測定法の選択が不明確であることに加え、測定法のカットオフ・サイズも説明されていないことが多く、トップ3やトップ5のリストから数百の要素まである。

しかし、これらの現象はニューラル推薦アプローチと結びついておらず、推薦システムのアルゴリズム研究においても、プレニューラルな時間において見出すことができる。27, 46]の議論を考慮すると、このような発展は、機械学習研究者が精度測定に強く焦点を当て、「最良の」モデルを探索することによって促進される。現在の研究では、少なくとも1つか2つの公開データセットにおいて、新しい手法が1つか2つの確立された精度尺度において、既存のアルゴリズムのセットを上回ることができることを示すのに十分であると考えられることが多い¹¹。しかし、評価尺度とデータセットの選択は、しばしば恣意的であるように思われる。

このような不明確な研究手法の例として、暗黙のフィードバックデータセットに対するアルゴリズムの評価にMovieLensレーティングデータセットを使用することが挙げられる。このような実践は、研究がいかなる仮説によっても導かれるものでも、与えられた問題の解決を目指すものでもないという根本的な問題を指し示している。推薦消費者や推薦者にとって付加価値の向上という点で、やや高い精度値が関連するかどうかさえ明らかではないが、より良い精度値を求めることが、この分野の研究活動を支配している[20, 22, 52]。実際、精度が高くて必ずしも推奨事項が見直されるわけではないことを示す研究成果が数多く存在する[4, 9, 13, 31, 37]。

¹¹本研究に関連すると考えられる18の論文のうち、新しいDLアーキテクチャを提案した論文が少なくとも2つあり、それらは単一のプライベートデータセットで評価され、ソースコードが提供されていない。

4 DISCUSSION

4.1 Reproducibility and Scalability

In some ways, establishing reproducibility in applied machine learning should be much easier than in other scientific disciplines and also other subfields of computer science. While many recommendation algorithms are not fully deterministic, e.g., because they use some form of random initialization of parameters, the variability of the obtained results when repeating the exact same experiment configuration several times is probably very low in most cases. Therefore, when researchers provide their code and the used data, everyone should be able to reproduce more or less the exact same results. Given that researchers today often rely on software that is publicly available or provided by academic institutions, the barriers regarding technological requirements are mostly low as well. In particular, virtualization technology should make it easier for other researchers to repeat an experiment under very similar conditions.

Nonetheless, our work shows that the level of reproducibility is actually not high. The code of the core algorithms seems to be more often shared by researchers than in the past, probably also due to the fact that reproducibility has become an evaluation criterion for conferences. However, in many cases, the code that is used for hyper-parameter optimization, evaluation, data pre-processing, and for the baselines is not shared. This makes it difficult for others to validate the reported findings.

One orthogonal factor that can make reproducibility challenging is the computational complexity of many of the proposed methods. Ten years after the Netflix Prize and its 100 million rating dataset, researchers, in the year 2019, commonly use datasets containing only a few hundred thousand ratings. Even for such tiny datasets, which were considered unacceptably small a few years ago, hyper-parameter optimization can take days or weeks, even when researchers have access to GPU computing. Clearly, nearest-neighbor methods, as discussed in our paper, can also lead to scalability issues. However, with appropriate data pre-processing and data sampling mechanisms, scalability can also be ensured for such methods, both in academic and industrial environments [19, 26].

4.2 Progress Assessment

Despite their computational complexity, our analysis showed that several recently proposed neural methods do not even outperform conceptually or computationally simpler, sometimes long-known, algorithms. The level of progress that is achieved in the field of neural methods is, therefore, unclear, at least when considering the approaches discussed in our paper.

One main reason for this *phantom progress*, as our work shows, lies in the choice of the baselines and the lack of a proper optimization of the baselines. In the majority of the investigated cases, not enough information is given about the optimization of the considered baselines. Sometimes, we also found that mistakes were made with respect to data splitting and the implementation of certain evaluation measures and protocols.

Another interesting observation is that a number of recent papers use the neural collaborative filtering method (NCF) [14] as one of their state-of-the-art baselines. According to our analysis, this method is however outperformed by simple baselines on one

dataset and does not lead to much better results on another, where it is also outperformed by a standard implementation of a linear regression method. Therefore, progress is often claimed by comparing a complex neural model against another neural model, which is, however, not necessarily a strong baseline. Similar observations can be made for the area of session-based recommendation, where a recent method based on recurrent neural networks [16] is considered a competitive baseline, even though almost trivial methods are in most cases better [29, 30].

Another aspect that makes it difficult to assess progress in the field lies in the variety of datasets, evaluation protocols, metrics, and baselines that are used by researchers. Regarding datasets, for example, we found over 20 public datasets that were used, plus several variants of the MovieLens and Yelp datasets. As a result, most datasets are only used in one or two papers. All sorts of metrics are used (e.g., Precision, Recall, Mean Average Precision, NDCG, MRR etc.) as well as various evaluation procedures (e.g., random holdout 80/20, leave-last-out, leave-one-out, 100 negative items or 50 negative items for each positive). In most cases, however, these choices are not well justified beyond the fact that others used them before. In reality, the choice of the metric should depend on the application context. In some applications, for example, it might be important to have at least one relevant item at the top of the recommendations, which suggests the use of rank-based metrics like MRR. In other domains, high Recall might be more important when the goal is to show as many relevant items as possible to the user. Besides the unclear choice of the measure, often also the cutoff sizes for the measurement are not explained and range from top-3 or top-5 lists to several hundred elements.

These phenomena are, however, not tied to neural recommendation approaches, but can be found in algorithmic research in recommender systems also in pre-neural times. Considering the arguments from [27, 46], such developments are fueled by the strong focus of machine learning researchers on accuracy measures and the hunt for the “best” model. In our current research practice, it is often considered sufficient to show that a new method can outperform a set of existing algorithms on at least one or two public datasets on one or two established accuracy measures.¹¹ The choice of the evaluation measure and dataset however often seems arbitrary.

An example of such unclear research practice is the use of MovieLens rating datasets for the evaluation of algorithms for implicit feedback datasets. Such practices point to the underlying fundamental problem that research is not guided by any hypothesis or aim at the solution of a given problem. The hunt for better accuracy values dominates research activities in this area, even though it is not even clear if slightly higher accuracy values are relevant in terms of adding value for recommendation consumers or providers [20, 22, 52]. In fact, a number of research works exist that indicate that higher accuracy does not necessarily translate into better-received recommendations [4, 9, 13, 31, 37].

¹¹From the 18 papers considered relevant for our study, there were at least two papers which proposed new DL architectures which were evaluated on a single private dataset and for which no source code was provided.

5 SUMMARY

本研究では、トップ n 推薦のための最近のニューラルアルゴリズムを多数分析した。我々の分析は、発表された研究を再現することはまだ困難であることを示している。さらに、レビューされた研究のほとんどは、少なくともいくつかのデータセットにおいて、概念的にも計算的にもより単純なアルゴリズムで上回ることができることが判明した。したがって、我々の研究は、この分野におけるアルゴリズムの貢献の評価に関して、より厳密でより良い研究実践を求めている。

これまでの分析は、特定の学会シリーズで発表された論文に限定している。現在進行中および将来の研究において、我々は他の出版物や他のタイプの推薦問題に我々の分析を拡張する予定である。さらに、より伝統的なアルゴリズムをベースラインとして、例えば行列分解に基づくアルゴリズムを検討する予定である。

REFERENCES

- [1] S. Antenucci, S. Boglio, E. Chioso, E. Dervishaj, K. Shuwen, T. Scarlatti, and M. Ferrari Dacrema. 2018. Artist-driven layering and user's behaviour impact on recommendations in a playlist continuation scenario. In *Proceedings of the ACM Recommender Systems Challenge 2018 (RecSys 2018)*. <https://doi.org/10.1145/3267471.3267475> Source: <https://github.com/MaurizioFD/spotify-recsys-challenge>.
- [2] Timothy G. Armstrong, Alistair Moffat, William Webber, and Justin Zobel. 2009. Improvements That Don't Add Up: Ad-hoc Retrieval Results Since 1998. In *Proceedings CIKM '09*. 601–610.
- [3] Joeran Beel, Corinna Breiteringer, Stefan Langer, Andreas Lommatzsch, and Bela Gipp. 2016. Towards reproducibility in recommender-systems research. *User Modeling and User-Adapted Interaction* 26, 1 (2016), 69–101.
- [4] Jöran Beel and Stefan Langer. 2015. A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems. In *Proceedings TPDL '15*. 153–168.
- [5] Robert M Bell and Yehuda Koren. 2007. Improved neighborhood-based collaborative filtering. In *KDD cup and workshop at the KDD '07*. Citeseer, 7–14.
- [6] Homanga Bharadhwaj, Homin Park, and Brian Y. Lim. 2018. RecGAN: Recurrent Generative Adversarial Networks for Recommendation Systems. In *Proceedings RecSys '18*. 372–376.
- [7] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention. In *Proceedings SIGIR '17*. 335–344.
- [8] Colin Cooper, Sang Hyuk Lee, Tomasz Radzik, and Yiannis Siantos. 2014. Random walks in recommender systems: exact computation and simulations. In *Proceedings WWW '14*. 811–816.
- [9] Paolo Cremonesi, Franca Garzotto, and Roberto Turrin. 2012. Investigating the Persuasion Potential of Recommender Systems from a Quality Perspective: An Empirical Study. *Transactions on Interactive Intelligent Systems* 2, 2 (2012), 1–41.
- [10] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative Memory Network for Recommendation Systems. In *Proceedings SIGIR '18*. 515–524.
- [11] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings WWW '15*. 278–288.
- [12] Association for Computing Machinery. 2016. Artifact Review and Badging. Available online at: <https://www.acm.org/publications/policies/artifact-review-badging> (Accessed March, 2018).
- [13] Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Brutin, and Amr Huber. 2014. Offline and Online Evaluation of News Recommender Systems at Swissinfo.Ch. In *Proceedings RecSys '14*. 169–176.
- [14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings WWW '17*. 173–182.
- [15] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep Reinforcement Learning That Matters. In *Proceedings AAAI '18*. 3207–3214.
- [16] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *Proceedings ICLR '16*.
- [17] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *Proceedings KDD '18*. 1531–1540.
- [18] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings ICDM '08*. 263–272.
- [19] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks Meet the Neighborhood for Session-Based Recommendation. In *Proceedings RecSys '17*. 306–310.
- [20] Dietmar Jannach, Paul Resnick, Alexander Tuzhilin, and Markus Zanker. 2016. Recommender Systems - Beyond Matrix Completion. *Commun. ACM* 59, 11 (2016), 94–102.
- [21] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional Matrix Factorization for Document Context-Aware Recommendation. In *Proceedings RecSys '16*. 233–240.
- [22] Joseph A. Konstan and John Riedl. 2012. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction* 22, 1 (2012), 101–123.
- [23] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings KDD '17*. 305–314.
- [24] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings WWW '18*. 689–698.
- [25] Jimmy Lin. 2019. The Neural Hype and Comparisons Against Weak Baselines. *SIGIR Forum* 52, 2 (Jan. 2019), 40–51.
- [26] G. Linden, B. Smith, and J. York. 2003. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing* 7, 1 (2003), 76–80.
- [27] Zachary C. Lipton and Jacob Steinhardt. 2018. Troubling Trends in Machine Learning Scholarship. arXiv:arXiv:1807.03341
- [28] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*. Springer, 73–105.
- [29] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of Session-based Recommendation Algorithms. *User-Modeling and User-Adapted Interaction* 28, 4–5 (2018), 331–390.
- [30] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2019. Performance Comparison of Neural and Non-Neural Approaches to Session-based Recommendation. In *Proceedings RecSys '19*. <https://doi.org/10.1145/3298689.3347041>
- [31] Andrii Maksai, Florent Garcin, and Boi Faltings. 2015. Predicting Online Performance of News Recommender Systems Through Richer Evaluation Metrics. In *Proceedings RecSys '15*. 179–186.
- [32] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2018. A Contextual Attention Recurrent Architecture for Context-Aware Venue Recommendation. In *Proceedings SIGIR '18*. 555–564.
- [33] Xia Ning and George Karypis. 2011. SLIM: Sparse linear methods for top-n recommender systems. In *Proceedings ICDM '11*. 497–506.
- [34] Bibek Paudel, Fabian Christoffel, Chris Newell, and Abraham Bernstein. 2017. Updatable, Accurate, Diverse, and Scalable Recommendations for Interactive Applications. *ACM Transactions on Interactive Intelligent Systems* 7, 1 (2017), 1.
- [35] Hans Ekehard Plesser. 2017. Reproducibility vs. Replicability: A Brief History of a Confused Terminology. *Frontiers in Neuroinformatics* 11, 76 (2017).
- [36] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *Comput. Surveys* 51, 4 (2018), 1–36.
- [37] Marco Rossetti, Fabio Stella, and Markus Zanker. 2016. Contrasting Offline and Online Results when Evaluating Recommendation Algorithms. In *Proceedings RecSys '16*. 31–34.
- [38] Naveen Sachdeva, Kartik Gupta, and Vikram Pudi. 2018. Attentive Neural Architecture Incorporating Song Features for Music Recommendation. In *Proceedings RecSys '18*. 417–421.
- [39] Alan Said and Alejandro Bellogin. 2014. Rival: A Toolkit to Foster Reproducibility in Recommender System Evaluation. In *Proceedings RecSys '14*. 371–372.
- [40] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings WWW '01*. 285–295.
- [41] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent Knowledge Graph Embedding for Effective Recommendation. In *Proceedings RecSys '18*. 297–305.
- [42] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings WWW '18*. 729–739.
- [43] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Multi-Pointer Co-Attention Networks for Recommendation. In *Proceedings SIGKDD '18*. 2309–2318.
- [44] Trinh Xuan Tuan and Tu Minh Phuong. 2017. 3D Convolutional Networks for Session-based Recommendation with Content Features. In *Proceedings RecSys '17*. 138–146.
- [45] Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016. Meta-Prod2Vec: Product Embeddings Using Side-Information for Recommendation. In *Proceedings RecSys '16*. 225–232.
- [46] Kiri Wagstaff. 2012. Machine Learning that Matters. In *Proceedings ICML '12*. 529–536.
- [47] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings KDD '11*. 448–456.

5 SUMMARY

In this work, we have analyzed a number of recent neural algorithms for top-n recommendation. Our analysis indicates that reproducing published research is still challenging. Furthermore, it turned out that most of the reviewed works can be outperformed at least on some datasets by conceptually and computationally simpler algorithms. Our work therefore calls for more rigor and better research practices with respect to the evaluation of algorithmic contributions in this area.

Our analyses so far are limited to papers published in certain conference series. In our ongoing and future work, we plan to extend our analysis to other publication outlets and other types of recommendation problems. Furthermore, we plan to consider more traditional algorithms as baselines, e.g., based on matrix factorization.

REFERENCES

- [1] S. Antenucci, S. Boglio, E. Chioso, E. Dervishaj, K. Shuwen, T. Scarlatti, and M. Ferrari Dacrema. 2018. Artist-driven layering and user's behaviour impact on recommendations in a playlist continuation scenario. In *Proceedings of the ACM Recommender Systems Challenge 2018 (RecSys 2018)*. <https://doi.org/10.1145/3267471.3267475> Source: <https://github.com/MaurizioFD/spotify-recsys-challenge>.
- [2] Timothy G. Armstrong, Alistair Moffat, William Webber, and Justin Zobel. 2009. Improvements That Don't Add Up: Ad-hoc Retrieval Results Since 1998. In *Proceedings CIKM '09*, 601–610.
- [3] Joeran Beel, Corinna Breiter, Stefan Langer, Andreas Lommatzsch, and Bela Gipp. 2016. Towards reproducibility in recommender-systems research. *User Modeling and User-Adapted Interaction* 26, 1 (2016), 69–101.
- [4] Jöran Beel and Stefan Langer. 2015. A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems. In *Proceedings TPD L '15*, 153–168.
- [5] Robert M Bell and Yehuda Koren. 2007. Improved neighborhood-based collaborative filtering. In *KDD cup and workshop at the KDD '07*. Citeseer, 7–14.
- [6] Homanga Bharadhwaj, Homin Park, and Brian Y. Lim. 2018. RecGAN: Recurrent Generative Adversarial Networks for Recommendation Systems. In *Proceedings RecSys '18*, 372–376.
- [7] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings SIGIR '17*, 335–344.
- [8] Colin Cooper, Sang Hyuk Lee, Tomasz Radzik, and Yiannis Siantos. 2014. Random walks in recommender systems: exact computation and simulations. In *Proceedings WWW '14*, 811–816.
- [9] Paolo Cremonesi, Franca Garzotto, and Roberto Turrin. 2012. Investigating the Persuasion Potential of Recommender Systems from a Quality Perspective: An Empirical Study. *Transactions on Interactive Intelligent Systems* 2, 2 (2012), 1–41.
- [10] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative Memory Network for Recommendation Systems. In *Proceedings SIGIR '18*, 515–524.
- [11] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings WWW '15*, 278–288.
- [12] Association for Computing Machinery. 2016. Artifact Review and Badging. Available online at: <https://www.acm.org/publications/policies/artifact-review-badging> (Accessed March, 2018).
- [13] 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 RecSys '14*, 169–176.
- [14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings WWW '17*, 173–182.
- [15] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep Reinforcement Learning That Matters. In *Proceedings AAAI '18*, 3207–3214.
- [16] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *Proceedings ICLR '16*.
- [17] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *Proceedings KDD '18*, 1531–1540.
- [18] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings ICDM '08*, 263–272.
- [19] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks Meet the Neighborhood for Session-Based Recommendation. In *Proceedings RecSys '17*, 306–310.
- [20] Dietmar Jannach, Paul Resnick, Alexander Tuzhilin, and Markus Zanker. 2016. Recommender Systems - Beyond Matrix Completion. *Commun. ACM* 59, 11 (2016), 94–102.
- [21] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional Matrix Factorization for Document Context-Aware Recommendation. In *Proceedings RecSys '16*, 233–240.
- [22] Joseph A. Konstan and John Riedl. 2012. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction* 22, 1 (2012), 101–123.
- [23] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings KDD '17*, 305–314.
- [24] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings WWW '18*, 689–698.
- [25] Jimmy Lin. 2019. The Neural Hype and Comparisons Against Weak Baselines. *SIGIR Forum* 52, 2 (Jan. 2019), 40–51.
- [26] G. Linden, B. Smith, and J. York. 2003. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing* 7, 1 (2003), 76–80.
- [27] Zachary C. Lipton and Jacob Steinhardt. 2018. Troubling Trends in Machine Learning Scholarship. [arXiv:arXiv:1807.03341](https://arxiv.org/abs/1807.03341)
- [28] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*. Springer, 73–105.
- [29] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of Session-based Recommendation Algorithms. *User-Modeling and User-Adapted Interaction* 28, 4–5 (2018), 331–390.
- [30] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2019. Performance Comparison of Neural and Non-Neural Approaches to Session-based Recommendation. In *Proceedings RecSys '19*. <https://doi.org/10.1145/3298689.3347041>
- [31] Andrii Maksai, Florent Garcin, and Boi Faltings. 2015. Predicting Online Performance of News Recommender Systems Through Richer Evaluation Metrics. In *Proceedings RecSys '15*, 179–186.
- [32] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2018. A Contextual Attention Recurrent Architecture for Context-Aware Venue Recommendation. In *Proceedings SIGIR '18*, 555–564.
- [33] Xia Ning and George Karypis. 2011. SLIM: Sparse linear methods for top-n recommender systems. In *Proceedings ICDM '11*, 497–506.
- [34] Bibek Paudel, Fabian Christoffel, Chris Newell, and Abraham Bernstein. 2017. Updatable, Accurate, Diverse, and Scalable Recommendations for Interactive Applications. *ACM Transactions on Interactive Intelligent Systems* 7, 1 (2017), 1.
- [35] Hans Eklund Plesser. 2017. Reproducibility vs. Replicability: A Brief History of a Confused Terminology. *Frontiers in Neuroinformatics* 11, 76 (2017).
- [36] Massimo Quadroni, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *Comput. Surveys* 51, 4 (2018), 1–36.
- [37] Marco Rossetti, Fabio Stella, and Markus Zanker. 2016. Contrasting Offline and Online Results when Evaluating Recommendation Algorithms. In *Proceedings RecSys '16*, 31–34.
- [38] Naveen Sachdeva, Kartik Gupta, and Vikram Pudi. 2018. Attentive Neural Architecture Incorporating Song Features for Music Recommendation. In *Proceedings RecSys '18*, 417–421.
- [39] Alan Said and Alejandro Bellogin. 2014. Rival: A Toolkit to Foster Reproducibility in Recommender System Evaluation. In *Proceedings RecSys '14*, 371–372.
- [40] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings WWW '01*, 285–295.
- [41] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent Knowledge Graph Embedding for Effective Recommendation. In *Proceedings RecSys '18*, 297–305.
- [42] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings WWW '18*, 729–739.
- [43] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Multi-Pointer Co-Attention Networks for Recommendation. In *Proceedings SIGKDD '18*, 2309–2318.
- [44] Trinh Xuan Tuan and Tu Minh Phuong. 2017. 3D Convolutional Networks for Session-based Recommendation with Content Features. In *Proceedings RecSys '17*, 138–146.
- [45] Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016. Meta-Prod2Vec: Product Embeddings Using Side-Information for Recommendation. In *Proceedings RecSys '16*, 225–232.
- [46] Kiri Wagstaff. 2012. Machine Learning that Matters. In *Proceedings ICML '12*, 529–536.
- [47] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings KDD '11*, 448–456.

- [48] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings KDD '15*. 1235–1244.
- [49] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. 2006. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings SIGIR '06*. 501–508.
- [50] Jun Wang, Stephen Robertson, Arjen P de Vries, and Marcel JT Reinders. 2008. Probabilistic relevance ranking for collaborative filtering. *Information Retrieval* 11, 6 (2008), 477–497.
- [51] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings WSDM '16*. 153–162.
- [52] Bo Xiao and Izak Benbasat. 2007. E-commerce Product Recommendation Agents: Use, Characteristics, and Impact. *MIS Quarterly* 31, 1 (March 2007), 137–209.
- [53] Lei Zheng, Chun-Ta Lu, Fei Jiang, Jiawei Zhang, and Philip S. Yu. 2018. Spectral Collaborative Filtering. In *Proceedings RecSys '18*. 311–319.

- [48] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings KDD '15*. 1235–1244.
- [49] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. 2006. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings SIGIR '06*. 501–508.
- [50] Jun Wang, Stephen Robertson, Arjen P de Vries, and Marcel JT Reinders. 2008. Probabilistic relevance ranking for collaborative filtering. *Information Retrieval* 11, 6 (2008), 477–497.
- [51] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings WSDM '16*. 153–162.
- [52] Bo Xiao and Izak Benbasat. 2007. E-commerce Product Recommendation Agents: Use, Characteristics, and Impact. *MIS Quarterly* 31, 1 (March 2007), 137–209.
- [53] Lei Zheng, Chun-Ta Lu, Fei Jiang, Jiawei Zhang, and Philip S. Yu. 2018. Spectral Collaborative Filtering. In *Proceedings RecSys '18*. 311–319.