Received: 1 February 2021 | Revised: 2 September 2021 | Accepted: 6 October 2021

DOI: 10.1111/exsy.12871

ORIGINAL ARTICLE

Expert Systems WILEY

A replication study on implicit feedback recommender systems with application to the data visualization recommendation

Parisa Lak[1] | Aysun Bozanta[1] | Can Kavaklioglu[1] | Mucahit Cevik[1] | Ayse Basar[1] | Martin Petitclerc[2] | Graham Wills[2]

[1]Data Science Lab, Mechanical Industrial Engineering Department, Ryerson University, Ontario, Canada

[2]Watson Analytics, IBM US, New York, USA

Correspondence
Aysun Bozanta, Data Science Lab, Mechanical Industrial Engineering Department, Ryerson University, Toronto, ON M5B 2K3, Canada.
Email: aysun.bozanta@ryerson.ca

Funding information
IBM CAS Project 919; NSERC CRD Grant 490782 2015

Abstract

In this study, we compare the Bayesian personalized ranking (BPR) algorithms with two recent state-of-the-art algorithms, namely, noisy-label robust Bayesian point-wise optimization (NBPO) and Light Graph Convolution Network (LightGCN) algorithms, to validate and generalize their performance by using six publicly available datasets and one proprietary dataset containing web-based data visualization usage records. We follow the guidelines explained in the original studies to pre-process the input data and evaluate these algorithms using various evaluation metrics. We also perform hyperparameter tuning for the recommendation algorithms to determine the optimal configuration resulting in the best recommendation quality. We observe that the best hyperparameter configuration varies based on the algorithms and the datasets. The results of our analysis show some similarities with the results of the original studies while differing in certain respects. We observe that adaptive oversampling BPR (AOBPR) and LightGCN algorithms generate higher quality recommendations than the other algorithms. However, algorithm convergence rates vary significantly for each dataset. We note that the AOBPR approach is particularly useful for data visualization recommendation task, and can contribute to the improved recommendations in practice.

KEYWORDS

Bayesian personalized ranking, LightGCN, matrix factorization, NBPO, negative sampling, ranking prediction

1 | INTRODUCTION

The goal of recommender systems is to provide users with relevant items (Jannach et al., 2010). Recommender systems can be implemented in many applications such as e-commerce, entertainment, tourism and healthcare. The algorithms behind recommender systems use the collected data about user–system interactions and then provide user specific item suggestions. User interaction data may be in the form of explicit user preferences such as ratings and reviews. Alternatively, user preferences may be derived implicitly based on the historical user interactions with the application such as in e-commerce website or a movie recommendation application (Jawaheer et al., 2014). Some examples of implicit data sources are user click (Gantner et al., 2011), purchase (Schafer et al., 1999) or time spent on a webpage (Claypool et al., 2001). User–system interaction data is generally regarded as positive feedback collected from a user. A large number of studies in the literature consider the items that are not listed in the positive feedback as not preferred or with negative user preferences (Claypool et al., 2001; Hu et al., 2008; Schafer et al., 1999), which might not always be true. This problem can be addressed by pair-wise comparison of the items.

https://doi.org/10.1111/exsy.12871

The problem of providing a list of sorted relevant items to the users can be considered as learning to rank (LtR) problem. The existing LtR algorithms can be grouped into three main categories: point-wise, pair-wise, and list-wise approaches (Liu, 2009). Bayesian personalized ranking (BPR) is one of the popular methods that is based on pair-wise user preferences for implicit user feedback (Rendle et al., 2009). Since the explicit source of information (i.e. positive-only feedback) is limited compared to the total number of available items, the pair-wise comparison is performed on a sample of positive–negative item pairs for each user. This approach is useful as there can be some items that a user might like but he/she has not seen those. In the BPR algorithm, items which have not been visited, rated, or clicked yet, are chosen as negative items using a uniform sampling distribution. Various versions of the BPR algorithm using different sampling distributions have been developed to overcome this potential issue.

Some of the more popular extensions of the BPR algorithm, namely, adaptive oversampling BPR (AOBPR) and static oversampling BPR (SCIBPR) are proposed by Rendle and Freudenthaler (2014). The AOBPR algorithm has two important features: being context-dependent and adaptive. The sampling probability depends on a given context, which implies that when the model recognizes more contexts, it becomes more context-aware and generates better results. Besides, the sampler is not static, and it changes as model parameters are learned. On the other hand, in the SCIBPR algorithm, the item distribution is static and independent of the context, which means that the sampler does not change while the model parameters are learned. Yu and Qin (2020) proposed noisy-label robust Bayesian point-wise optimization (NBPO) algorithm to address the weaknesses of the BPR algorithm in terms of sampling unvoted items as negative items uniformly. In particular, they improved the adaptive sampling technique based on noisy-label robust learning in place of uniform sampling of the unvoted items, and similar to BPR, they used matrix factorization for learning. Different from BPR variants and NBPO algorithm, He et al. (2020) consider deep models for personalized recommendations. Specifically, they simplified the Graph Convolution Network (GCN) based approach proposed by Kipf and Welling (2016) by only focusing on the neighbourhood aggregation.

1.1 | Motivation

One of the most frequently encountered problems in the field of recommendation systems is to incorporate items that have not yet been scored by the users into the recommendation system algorithm, and include these items in the candidate set of recommendations. This problem is generally addressed in previous studies by either ignoring these items or evaluating them as negative items. Because the underlying problem is inherently difficult and the existing methods typically fail to generate highly accurate recommendations, this problem has been the subject of many studies in recent literature, and still maintains its importance in the field of recommendation systems. Studies proposing new recommendation algorithms typically provide a comparative performance analysis with some of the existing methods and available datasets. On the other hand, an extensive analysis with a large number of models and datasets can help to provide deeper insights into the capabilities of existing methods in this domain, which we set out to achieve in this study. Furthermore, reproducibility is an important concept in recommendation systems as well as in much AI-related research that includes computational components, and testing the reproducibility of the available results helps to ensure higher adaptability of these methods in practice. Nevertheless, we note that there is a limited number of such replication studies in the literature for comparing recommendation system algorithms. Lastly, our extensive analysis can be used as a benchmark for future research in this field and can facilitate associated comparative analysis.

1.2 | Research goals and scope

In this study, we aim to replicate the results of the selected studies (He et al., 2020; Rendle & Freudenthaler, 2014; Yu & Qin, 2020) to validate the presented results and test out their generalizability. Specifically, we consider the BPR algorithm (Rendle et al., 2009), which has been frequently used as a benchmark in the recommender systems research, its two extensions (SCIBPR and AOBPR), and two recent state-of-art algorithms (LightGCN and NBPO). First, we aim to show the reproducibility of the results presented in the original studies by experimenting with the proposed algorithms on the same datasets used in these studies. Then, we test these algorithms on additional datasets, and examine their performance on datasets with different characteristics, using evaluation metrics that were not previously used in the original studies. We also identify the best-performing hyperparameters and assess the sensitivity of the algorithms to different hyperparameters. Accordingly, through this detailed empirical analysis, we aim to provide insights on the capabilities of various recommendation systems algorithms for handling implicit feedback, and create benchmark results to facilitate future research in this field. In our numerical study, we compare the performances of five recommender system algorithms which generate recommendations using implicit feedback on six publicly available and one propriety datasets from different domains. Some of these datasets have been also used in the original studies (e.g. see ECML'09 dataset in Rendle et al. (2009)'s study and Gowalla dataset in He et al. (2020)'s study). As the propriety dataset, we consider the IBM Watson data visualization dataset. IBM Watson analytics is a tool that enables users to discover patterns and meaning in their data. Therefore, recommending suitable analysis and visualization techniques to examine the user data is crucial to increase the usage of this tool. However, it is difficult to understand user reactions to make suggestions because Watson Analytics (WA) dataset is generated using implicit feedback similar to the most datasets in the literature. In addition, the data is heavy-tailed in terms of item popularity. Therefore, one or more of these algorithms may be suitable for this task as they are shown to provide a solution for the convergence issues in recommender systems while preserving the prediction quality and run time performance for similar datasets.

1.3 | Contribution

The contributions of our study to the literature can be summarized as follows:

1. We conduct a replication study with the recommendation algorithms utilizing implicit feedback to generate recommendations. We verify and validate the results of the original studies using various evaluation metrics. We experiment with six publicly available datasets from different domains, and a propriety dataset about data visualization recommendations (which is obtained from IBM Watson Analytics, and henceforth referred to as IBM Watson dataset) to generalize these algorithms' results to other domains and application problems.
2. We present detailed hyperparameter tuning results for the algorithms to identify the best parameter configurations and assess their overall robustness.
3. Our analysis sheds light on some details about the original studies that were not previously reported such as the convergence behaviours and the training times. Our results indicate that AOBPR algorithm proposed by Rendle and Freudenthaler (2014) is able to overperform more recent algorithms such as LightGCN and NBPO for the majority of the datasets. Furthermore, AOBPR performance is particularly promising for data visualization recommendations. IBM Watson Analytics currently relies on recommendations that are based on a set of rules, and more precise recommendations for its users through AOBPR contributes to the practice.

1.4 | Structure of the paper

This paper is structured based on the replication study framework suggested by Carver (2010). In Section 2, we review the importance of the replication studies in information retrieval (IR) and provide an overview of previous studies on recommender systems and LtR. In Section 3, we review the original studies that proposed the algorithms considered in our analysis. In Section 4, we provide details on the datasets and the experimental setup. We compare our findings with those of the original studies in Section 5. We also discuss the results for the IBM Watson dataset and investigate the best hyperparameter configuration for each algorithm on this dataset. Finally, we describe the threats to study validity from the replication viewpoint in Section 6 and conclude the paper with a summary of our findings, the limitations of the study, and future work in Section 7.

2 | BACKGROUND

In this section, we review the literature on both the value of replication studies in the field of IR as well as algorithmic approaches for recommender systems. We also briefly review IBM Watson Analytics, which provides data visualization recommendations.

Reproducibility is one of the key elements of empirical research (Kazai & Fuhr, 2011). There are numerous studies discussing different aspects of reproducibility such as the infrastructure for managing experimental data (Agosti, Di Buccio, et al., 2012; Agosti, Ferro, & Thanos, 2012), use of private data in evaluation (Callan & Moffat, 2012), evaluation as a service (Hanbury et al., 2015) and reproducible baselines (Agosti et al., 2006; Di Buccio et al., 2015).

Research community in IR domain also emphasize the importance of reproducibility of research (Ferro et al., 2016; Zobel et al., 2011). Ferro (2017) outlined some of the challenges in reproducibility of the IR research. One of the challenges is reported as the "lack of commonly agreed formats for modelling and describing the experimental data as well as almost having no metadata (descriptive, administrative, copyright, etc.) for annotation". It is argued that in computational studies, the semantics of the data is often not explicit. That is, researchers develop unique scripts for processing the data, and these scripts are not well documented. Ferro (2017) further explained that all of these issues might be addressed by adapting solutions developed in other fields with similar problems. Shull et al. (2008) discussed the characteristics of a successful replication study. They suggested that both replication with similar results and different results can be considered successful.

2.1 | Recommender systems

The recommender systems have been an important research area since it emerged as an independent field in the mid-1990s (Adomavicius & Tuzhilin, 2005). The techniques used for recommender systems are usually classified into the following categories: content-based filtering, collaborative filtering, and hybrids (Adomavicius & Tuzhilin, 2005). The collaborative filtering approach can also be divided into subcategories as model-based (a learning-based) and memory-based filtering. A learning-based or model-based recommender system is described as a set of machine learning algorithms that takes users' historical preferences in an application as an input, and predicts what might be of their interest in their next interaction with the application (Jannach et al., 2010). The historical preferences are either provided by the user as explicit information (e.g. movie ratings) or can be derived from their interaction with the system as implicit information (e.g. clicks) (Oard & Kim, 1998). A significant body of work has been proposed both in academia and in industry on the advancements of these systems during the last two decades (Ricci et al., 2015). Despite the advances in computational power and the incorporation of cognitive computing techniques in recommender systems, there remain many research challenges to determine the relevancy of options presented to the users by these systems in the context of cold start, data sparsity, and modelling implicit user preferences (Ricci et al., 2015).

Few other studies in the literature conducted replication studies for recommender systems. Çoba and Zanker (2017) proposed a new R package for recommender system algorithms, and compared FunkSVD, SVD approximation, and WSlopOne algorithms on the MovieLens dataset. Polatidis et al. (2018); Polatidis et al. (2019) used Apache Mahout and Recommender101 libraries to compare different recommender system algorithms on the MovieLens dataset. However, these studies did not provide an extensive analysis with a large number of recommender system algorithms and diverse datasets, which we perform in this study for the task of generating recommendations from implicit feedback. In addition, to the best of our knowledge, there are no other replication studies for implicit feedback recommender systems in the literature.

Recent review papers on recommender systems mostly focus on methodological developments in the field. Mu (2018) provided a review of deep learning-based recommender systems algorithms, which have been shown to be effective for learning the latent representations of users and items from large datasets. Najafabadi et al. (2019) reviewed data mining techniques in recommender systems with a specific focus on collaborative filtering-based approaches. Zhang et al. (2020) pointed to the high efficiency of autoencoder architectures in information retrieval tasks and reviewed the recent work on autoencoder-based recommender systems, also highlighting the differences from traditional recommender systems.

Data visualization is one of the widely used functions by data analysts, and accordingly, there is an increasing interest in this area through data visualization tools such as PowerBI, Google charts, and Tableau. These data visualization tools have started to provide some graph, chart, or table recommendations based on the characteristics of a given data. Therefore, new recommendation systems are required to automatically recommend and construct visualizations that highlight patterns or trends of interest. Recently, there have been few studies focusing on visualization recommendation task (Dibia & Demiralp, 2019; Hu et al., 2019; Luo et al., 2018). For instance, DeepEye (Luo et al., 2018) used a binary classifier to classify the visualizations as good or bad, and then rank those. Data2V is (Dibia & Demiralp, 2019) handles the problem as a language translation problem and applied a neural machine translation approach to create a sequence-to-sequence model. VizML (Hu et al., 2019) considered a machine learning-based approach for visualization recommendation. They determine the design choices of the analyst and then train a neural network to predict the best design choices using 1 million data instances. In our study, we consider the data visualization recommendation task at a high level through a dataset consisting of user-item pairs, which implicitly signifies the preferences of the users for certain visualizations. Accordingly, implicit feedback recommender systems are suitable for our dataset to recommend data visualizations to the IBM Watson users. In addition, detailed empirical analysis with various recommender systems algorithms helps to understand the most suitable approach for our data visualization recommendation task.

2.2 | Recommendation with implicit feedback

Implicit feedback data is generally collected during the user-system interaction without users' consciousness (Das et al., 2017). This data may come from various applications such as browsing history, web purchase history, mouse movements, watched movies, played songs or search patterns. The most important advantage of implicit feedback is that large amounts of data can be collected in a short amount of time and at a lower cost. It also has its own drawbacks such as being difficult to interpret, vulnerable to noise, and less accurate compared to explicit feedback (Das et al., 2017; Hu et al., 2008). Modelling user preferences from the implicit information is considered to be the primary source of information for many recommender systems regardless of the selected technique or algorithm (Peska & Vojtas, 2017).

Implicit feedback data is considered to be one-class data, as only positive observations are accessible (Rendle & Freudenthaler, 2014). The general approach to handle this type of data is to assume all unvoted items as less interesting or negative items. Another deficiency of such data is being long-tailed. The item options are numerous and, generally, a small group of items continues to be even more popular, while the other items were either seen by a small number of people or not at all. For these types of datasets, algorithms generally learn based on sampling pairs for stochastic gradient descent (SGD) updates during training.

Table 1 summarizes the studies focusing on the recommender systems that generate recommendations using implicit feedback data, and the relative positioning of our work with respect to these studies. We specifically provide the research objectives, datasets, and algorithms in this table. Rendle et al. (2009) first proposed a BPR algorithm that is based on a uniform sampling of negative items for the SGD updates. Rendle and

TABLE 1 Summary of the relevant studies in the literature

Reference	Objective	Datasets	Tested algorithms
Rendle et al. (2009)	BPR	Rossman, Netflix	WR−MF, SVD−MF, Cosine−kNN, Item-pop, npmax
Rendle and Freudenthaler (2014)	SCIBPR, AOBPR	BBC, ECML	BPR, Item-pop
Wang et al. (2017)	IRGAN	Movielens, Netflix	MLE, BPR, LambdaFM
He et al. (2018)	AMF	Yelp, Pinterest, Gowalla	Item-pop, BPR, CDAE, IRGAN, NeuMF
Park and Chang (2019)	AdvIR	Letor, Movielens, Insurance QA	BPR, LambdaFM, IRGAN
He et al. (2020)	LightGCN	Gowalla, Yelp, Amazon-book	NGCF, Mult-VAE, GRMF
Yu and Qin (2020)	NBPO	Amazon, Movielens	Item-pop, Item-KNN, BPR, WBPR, ShiftMC
Our study	Replication and performance comp	ECML, Gowalla, Amazon, Movielens, Frappe, Jester, Watson	BPR, SCIBPR, AOBPR, NBPO, LightGCN

Freudenthaler (2014) improved the original BPR algorithm by employing a non-uniform sampling method for the negative items. They proposed Static Oversampling BPR (SCIBPR) as a baseline approach, which applies sampling according to the global item popularity. The potential drawback of this algorithm is that the sampling distribution does not change while model parameters are learned. Then, they proposed AOBPR with adaptive and context-aware sampling distribution that follows the belief of the ranking model. AOBPR updates the sampler while model parameters are learned. This adaptive mechanism also affects the scoring model and the final rankings, which are reportedly more accurate than those of BPR.

Various other studies consider different sampling techniques for the negative items. Wang et al. (2017) proposed IRGAN based on two information retrieval models (generative retrieval and discriminative retrieval) and employed a game-theoretical minimax game to iteratively optimize both models. Generative retrieval model predict relevant documents (e.g. items) given a query (e.g. a user), and discriminative retrieval model predict relevancy of a given query-document (e.g. user-item) pair. Their numerical study showed superior performance of IRGAN over other algorithms such as BPR and LambdaFM, which is a LambdaRank-based collaborative filtering algorithm. On the other hand, authors noted that gaps in theoretical foundations of GANs and the associated performance issues are also valid for IRGAN training. He et al. (2018) and Park and Chang (2019) also developed similar models to that of Wang et al. (2017) by performing adversarial training. Park and Chang (2019) proposed virtual adversarial training and selective virtual adversarial training, which do not require a labelled data to generate adversarial examples. One of the most recent recommender systems algorithms designed for implicit feedback is NBPO (Yu & Qin, 2020), which use Bayesian point-wise optimization (BPO), and consider the possibility of negative items as mislabelled and noisy data points rather than being simply unvoted items. However, authors noted various issues with BPO as an optimization routine, and reported weaker performance compared to BPR. Several other studies considered deep learning-based approaches including graph convolution networks for designing recommender systems (Berg et al., 2017; Ying et al., 2018). He et al. (2020) proposed the LightGCN algorithm, which is a simplified version of GCN algorithm (Wang et al., 2019) that only considers the neighbourhood aggregation, and do not incorporate feature transformation and nonlinear activation functions in the network. Using only these features of the GCN algorithm makes the LightGCN algorithm more effective, generalizable, and easily trainable. Their empirical analysis shows that the simplified deep model architecture leads to substantially improved performance. All these studies considered at most three different datasets in their comparative analysis, and often did not include a comprehensive list of existing methods in the numerical study. Such an approach to computational study is expected, as detailed numerical analysis is typically deemed out of scope for most studies. Accordingly, our study contributes to the literature by help filling this research gap.

2.3 | IBM Watson analytics

IBM Watson Analytics is a data visualization recommendation application. The users upload their dataset to the application, and the system provides them with visualization recommendations based on the data headers. The user may receive the recommendations on a dataset by either clicking on the dataset, or by submitting a specific question regarding the data in natural language. The current recommendation for this application is provided based on a set of rules. We refer to the current recommendation engine a "rule based recommendation system", rather than a "learning-based recommender system". For Watson, the task can be the recommendation of data fields along with the appropriate visualization type; the experience can be derived from the user interaction with the system, which provides user preferences through "selection" and not "selection actions". We define the items in Watson as visualization types and users as users of the Watson analytics.

3 | INFORMATION ABOUT THE ORIGINAL STUDIES

We briefly summarize the studies by Rendle et al. (2009), Rendle and Freudenthaler (2014), He et al. (2020), and Yu and Qin (2020), which introduce the algorithms used in our study.

Rendle et al. (2009) proposed a BPR algorithm for item recommendations from implicit feedback. They noted that commonly used generic methods such as matrix factorization and adaptive kNN are not designed to optimize the ranking of the recommended items. To address this issue, they proposed an optimization criterion, BPR-OPT, which is derived from the maximum posterior estimator for optimal personalized ranking. They defined the Bayesian formulation to obtain the personalized ranking for a set of items $\mathscr{I}$ as maximizing the following posterior probability:

$$p(\Theta| >_u) \propto p(>_u|\Theta)p(\Theta).$$

Here, Θ represents the parameter vector of the chosen model class such as matrix factorization, and $>_u$ denotes the latent preference structure for user $u \in \mathcal{U}$. In addition, $p(\Theta)$ corresponds to prior density, which is taken as normally distributed with zero mean and covariance matrix Σ_Θ. Assume that set of all pair-wise preferences in the dataset can be represented by $D_S \subseteq \mathcal{U} \times \mathscr{I} \times \mathscr{I}$. Then, the user-specified likelihood function can be obtained as

$$\prod_{u \in U} p(>_u|\Theta) = \prod_{(u,i,j) \in D_S} p(i >_u j|\Theta) = \prod_{(u,i,j) \in D_S} \sigma\left(\widehat{x}_{uij}(\Theta)\right),$$

where $\sigma\left(\widehat{x}_{uij}(\Theta)\right)$ specify the individual probability that user *u* prefer item *i* to item *j* using the logistic function σ, and $\widehat{x}_{uij}(\Theta)$ corresponds a real-valued function of Θ that models the relationship between user *u* and items *i* and *j*, which can be estimated using matrix factorization or adaptive kNN. Accordingly, BPR-OPT can be formulated as

$$\text{BPR-OPT} = \ln p(\Theta| >_u) = \sum_{(u,i,j) \in D_S} \ln\sigma\left(\widehat{x}_{uij}(\Theta)\right) - \lambda_\Theta \parallel \Theta\|^2,$$

where $\Sigma_\Theta = \lambda_\Theta I$, with λ_Θ denoting the model-specific regularization parameters. Authors develop an SGD-based learning algorithm, LEARNBPR, to maximize the BPR-OPT, which samples (u,i,j) triples uniformly.

Rendle and Freudenthaler (2014) extended the BPR algorithm that is previously proposed Rendle et al. (2009). They proposed a solution to deal with highly skewed data that is seen in many recommender system frameworks using implicit information (Adomavicius & Tuzhilin, 2005; Ricci et al., 2015). The skewed data is due to the fact that there is always a small number of items that receive users' feedback (i.e. source of implicit information), and are considered as of the interests to the user. Thus, the rest of the items corresponding to the majority of the item space are regarded as not interesting to the users. Accordingly, the authors note that the uniformly sampled pairs typically result in slow convergence, especially when the pool of items is large and the item popularity is tailed. Specifically, they observed that, when uniform sampling is employed for the negative items, in most SGD updates, the gradient vanishes and hence the update is not performed properly, resulting in epoch being wasted. This issue is attributed to the fact that the randomly selected negative items are very likely to be ranked correctly below a random positive item and thus the gradient of the positive–negative pair is near zero. Therefore, a non-uniform sampling distribution that adapts to the specific user and the current state of learning is considered as a way to overcome the slow convergence problem. Rendle and Freudenthaler (2014) proposed SCIBPR and AOBPR algorithms, which employ non-uniform sampling for the selection of the negative items. While SCIBPR is global (i.e., item distribution is identical for different users) and static (the distribution is fixed during training), AOBPR adapt to the user preferences and the current state of learning. The authors also presented an efficient sampling algorithm with constant amortized run-time for factorization methods such as matrix factorization and factorization machines. They conducted their numerical study on two datasets: a video dataset from BBC and a social tagging dataset from ECML'09, where they used Matrix Factorization and Tensor Factorization as their prediction algorithms for the two datasets, respectively. Their numerical study indicates that AOBPR algorithm improves the state of the art recommendation algorithm (i.e. BPR with uniform sampling) in terms of convergence rates while preserving the prediction quality and run-time performance.

Yu and Qin (2020) also investigated the issue of sampling of all negative items uniformly during SGD updates. They proposed the NBPO algorithm, in which they also considered the probability of mislabelling for each unobserved user-item pair. The authors argued that the BPR algorithm is not suitable for noisy-label robust learning, and accordingly, they considered BPO as the optimization method (Yu & Qin, 2020). The authors tested their algorithm on two datasets—the Amazon dataset consisting of user reviews of electronic products that are sold on Amazon.com and the Movie lens (1 M) dataset containing user-movie pairs, and compared its performance with five algorithms, namely, ItemPop, ItemKNN (Sarwar et al., 2001), BPR (Rendle et al., 2009), WBPR (Gantner et al., 2012), and ShiftMC (Hsieh et al., 2015). They obtained around 0.25% increase in

F1-Score, and less than 1% increase in normalized discounted cumulative gain (NDCG) score for the Amazon dataset, while for Movie lens (1 M) dataset, they achieved at most 1% increase in F1-score and less than 2% increase in NDCG score.

Many recent studies on recommender systems focus on deep learning models. GCNs are considered to be especially suitable for collaborative filtering as they enable integrating the user-item interactions as embedding in the network. The LightGCN algorithm proposed by He et al. (2020) is a variant of the GCN-based Neural Graph Collaborative Filtering (NGCF) algorithm (Wang et al., 2019), which is one of the recent state-of-the-art collaborative filtering techniques that exploits high-order connectivity from user-item interactions through embedding propagation layer. Two of the main features of the NGCF algorithm, feature transformation and nonlinear activation, were noted to make the training process difficult and also potentially decrease the recommendation performance. Therefore, the LightGCN algorithm does not incorporate these features of NGCF, instead it only adopts its neighbourhood aggregation component. Basically, the LightGCN algorithm learns user-item embedding on the user-item interaction graph by linearly propagating them. To find the final embedding, it calculates the weighted average of the embedding on all layers. He et al. (2020) tested their algorithms on the Gowalla, Yelp2018, and Amazon-book datasets and compared the performance of their algorithms with the other state-of-the-art algorithm, namely, NGCF (Wang et al., 2019), Mult-VAE (Liang et al., 2018), and GRMF (Rao et al., 2015), and they obtained around 1-2% increase in terms of recall and NDCG values.

4 | INFORMATION ABOUT REPLICATION

In this section, we first provide details on the datasets used in our analysis. Then, we present the experimental setup.

4.1 | Datasets

We performed our experiments on seven datasets. Our analysis with ECML'09, Gowalla, and Amazon datasets serve the purpose of verifying the correctness of our implementations as they were also separately used in the studies by Rendle and Freudenthaler (2014), He et al. (2020), and Yu and Qin (2020), respectively. The other four datasets, namely, Frappe, Movielens (100 K), Jester, and Watson, help explore the generalizability of the results of these studies. The number of users, items, interactions and the density values (i.e. the average number of rated items by each user) of all the datasets are presented in Table 2.[i] Overall, these datasets constitute a diverse set in terms of the application area and the dataset size to understand the capabilities of the personalized ranking algorithms.

A more detailed information about these datasets are provided below.

1. ECML'09: This is a publicly available social tagging dataset[ii] consisting of user-article-tag triplets. It consists of 1185 users, 7946 BibTeX, 22,852 BibTeX posts, 263,004 bookmark posts, 13,276 tags, and 253,615 tag assignments.
2. Gowalla: Gowalla is a location-based social network service, which was active between 2007 and 2012. This dataset consists of check-in data,[iii] which keeps the records of users' check-in information on this platform. We experiment with the same version of this dataset that was used by He et al. (2020). The dataset includes 29,858 users, 40,981 items, and 99,728 interactions. It is the most sparse dataset considered in our study with a density of 0.00084.
3. Amazon: This dataset consists of user reviews of electronic products that are sold on Amazon.com. We experiment with the same version of this dataset[iv] that was used in the study of (Yu & Qin, 2020). The dataset contains 1435 users, 1522 items, and 39,975 interactions, and its density is 0.0183.
4. Frappe: This dataset[v] consists of 957 users and 4082 items, and contains user preferences for different mobile applications (Baltrunas et al., 2015). This dataset also includes contextual features. However, such features are not considered within the scope of this study.

TABLE 2 Dataset specifications

Dataset	No. of users	No. of items	No. of interactions	Density
Gowalla	29,858	40,981	99,728	0.00084
Amazon	1435	1522	39,975	0.01830
Movielens (100 K)	943	1683	99,728	0.06284
Frappe	957	4082	96,203	0.02463
Jester	10,000	100	249,826	0.24983
Watson	317	2174	2503	0.00363

5. MovieLens (100 K): This dataset[vi] consists of 943 users and 1683 items, and contains an explicit rating information provided by each user on each item (Harper & Konstan, 2016). In order to follow the BPR setting for pair-wise comparison with implicit information, we take the appearance of each user-item pair as a positive source of information. Essentially, the ranking task in this application would be to find the next movie the user is going to rate. The frequency of rated movies is heavily tailed for popular movies.
6. Jester: The Jester is an online joke recommendation system. There are three versions of this dataset. We used the version[vii] in which there are 24,938 users who have rated between 15 and 35 jokes. There are 100 jokes and 1,810,455 interactions in the original version of this dataset. This is the most dense and the largest dataset among our datasets, and we randomly selected 10,000 users, and tested the algorithms on this subset to have reasonable run times.
7. Watson: This propriety dataset is from a web-based data visualization application. A data-extraction application was developed to query complete logs from the Logstash servers of the application. Then two main scripts were used to parse over a month long data from user's interaction with version 1 of the application during the month of October in 2015. There are 150 K observations with 971 real life users and 34,794 items. The items in the dataset are defined as the permutations of possible visualization types and the data headers provided within each data source by each user. We conducted an extensive exploratory data analysis that is presented elsewhere (Lak et al., 2016; 2017). The number of unique user-item pairs in WA dataset is 49,125 from which 3618 were positive feedback (i.e. selected items by the users) with frequencies higher than or equal to one, and the rest were not selected. As suggested by Rendle and Freudenthaler (2014), we only selected the users with more than five positive feedback activities. Accordingly, the dataset used in this experiment has 317 users and 2174 items.

In a typical recommender system, while some items are more popular, others may not have been preferred even once, therefore there is not enough information related to those, which causes data sparsity. The datasets with high data sparsity tend not to lead to high recommendation performance. Figure 1 provides an illustration for all the datasets that are used in our study, in which *x*-axis represents the item rank, and *y*-axis represents the selection frequency. If selection frequency is high, it implies that item popularity for those items are also high. Each graph in this figure has a long tail, and they are right skewed, which indicates that very few items has a high popularity in the datasets, and the datasets are very sparse. We note that while all the datasets are tailed with respect to popularity of the items, ECML'09 and Gowalla datasets are the most sparse datasets, and they show the highest similarity with respect to the popularity patterns of the items. On the other hand, Jester is the least sparse and the most dense dataset.

4.2 | Experimental setup

In our experiments with the recommendation algorithms, we mostly relied on the publicly available implementations. In all experiments, we applied fivefold cross-validation with an 80–20 training–testing split in each fold. Since the ECML'09 dataset is a three dimensional dataset, for only to replicate its results, we used the source code[viii] provided by Rendle (2012). The rest of the experiments with BPR algorithms were performed using Librec, a Java library for recommender systems.[ix] Note that our preliminary analysis shows that Librec library version of BPR algorithms and the original source codes provided by the authors generate comparable results for various datasets, however, we chose Librec library for ease of use. We detail the experimental setup with the BPR algorithms as follows. We consider a training epoch to be defined as $10 \cdot |S|$ single SGD update steps, *S* being the number of observed actions. While we experimented with factorization dimensions of 10, 20 and 64 in our preliminary analysis, the results are reported using 64 dimensions for all datasets as was the case in the original study. We take the learning rate as 0.01, and λ parameter for the Geometric distribution as 500. The number of training epochs are 100 for Watson and Movielens, 140 for Jester, 150 for Frappe, and 200 for Gowalla and Amazon. The hyperparameters for ECML'09 differ in the regularization, which is 0.005 for oversampling and 0.00005 for uniform sampling and 2000 training epochs as indicated in the original study (Rendle & Freudenthaler, 2014). In addition, Rendle and Freudenthaler (2014) state that since the ranking does not always change during the learning, we can only compute the ranking every $|I|\log|I|$ iterations. Accordingly, we also update the adaptive sampling evaluation at every $|I|\log|I|$ iterations.

In our experiments with LightGCN algorithm, we use the source codes provided by the authors[x] and take the hyperparameters values as specified in the original study (He et al., 2020) for the Gowalla dataset. Specifically, the embedding size is 64, regularization coefficient λ is 1e−4, learning rate is 0.001, batch size is 2048, and the number of epochs is 500. Similarly, for the NBPO algorithm, we use the source code[xi] provided by the authors and take the hyperparameters as specified in the original study (Yu & Qin, 2020). That is, the number of latent factors is 50, batch size is 5000, η is 0.05, λ_ϕ is 1, and the number of epochs is 200.

5 | RESULTS

We next provide our detailed numerical study with BPR algorithm and its variants, as well as NBPO and LightGCN, which are two of the most recent personalized ranking algorithms. We first focus on replicating the results provided in the original studies using the identical datasets. Then,

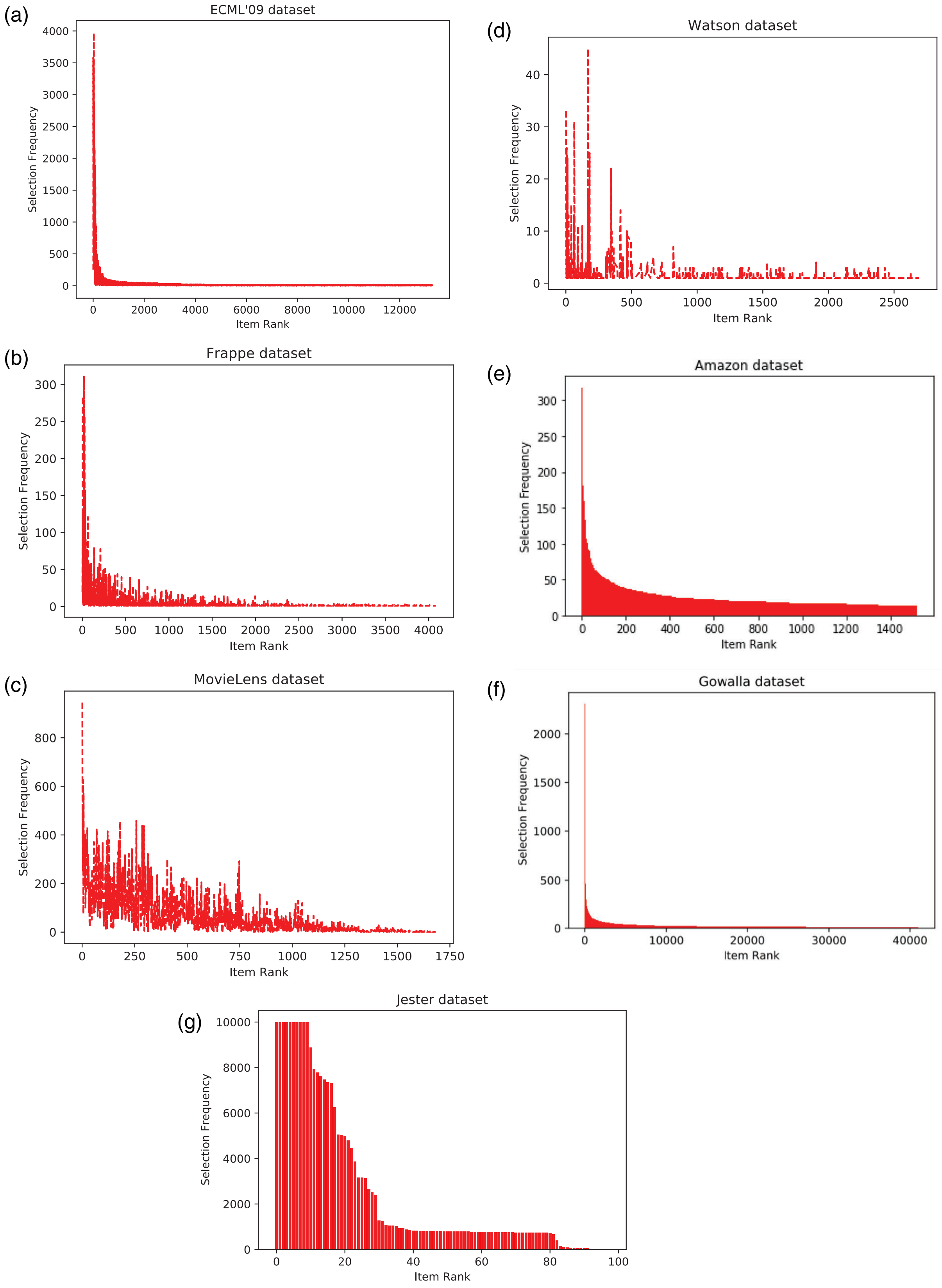


FIGURE 1 Item popularity in the dataset

we present the comparative results with the recommendation algorithms using all the datasets. Lastly, we assess the impact of the hyper-parameters using our propriety dataset, Watson.

We consider various performance metrics when comparing the algorithms. In particular, we report the performance values obtained using the test sets and measure the ranking quality using F1-Score, NDCG, and mean average precision (MAP) values. F1-Score is the harmonic mean of the recall and the precision values, where, in the context of recommender systems, precision is concerned with the number of recommendations that are relevant among the provided recommendations, and recall is concerned with the number of recommendations that are provided from all the relevant recommendations. That is,

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$

F1-score@*k* can be defined as the F1-score when *k* recommendations are provided. Recall@*k* and precision@*k* can be defined similarly. In addition, we compute MAP as the mean of the average precision values over all the users.

Lastly, we can obtain NDCG as the ratio of Discounted Cumulative Gain (DCG) of recommended order to DCG of the ideal order (iDCG). Specifically, DCG sums the relevance of the recommended item for the user while adding a (discounted) penalty for relevant items placed on later positions. That is, for *k* recommended items,

$$\text{NDCG}@k = \frac{\text{DCG}@k}{i\text{DCG}@k}, \quad \text{DCG}@k = \sum_{i=1}^{k} \frac{\text{rel}_i}{\log_2(i+1)}, \quad i\text{DCG}@k = \sum_{i=1}^{|\text{REL}_k|} \frac{\text{rel}_i}{\log_2(i+1)},$$

where rel_i is the true relevance of the recommendation at position *i* for the user, and $|\text{REL}_k|$ corresponds to the list of *k* relevant items that are sorted by their relevance. Note that in computing DCG and iDCG values, we discount the relevance score by dividing it with the logarithm of the corresponding position.

5.1 | Replication results

We first experiment with the BPR with uniform sampling, and the BPR with two oversampling strategies (i.e. SCIBPR and AOBPR) on ECML'09 dataset. We specifically report the convergence behaviour of the prediction quality metric (MAP) and gradient magnitudes. Gradient magnitude shows the effect of a training case on the BPR learning process. In Figure 2a, the green, red and blue lines show how many magnitudes are smaller than 0.01, 0.1 and 0.5, respectively. Similar to Rendle and Freudenthaler (2014), we observe that, after a few training epochs, almost all the samples have very small magnitudes indicating that most of the samples are useless in the SGD updates when BPR with uniform sampling is used, which explains the slow convergence of the algorithm. Figure 2 demonstrates the convergence of the BPR (with uniform sampling), SCIBPR, and AOBPR algorithms based on the MAP values calculated over the number of training epochs for the test set. We observe that AOBPR provides the best MAP values (≈0.36) while converging substantially faster than BPR with uniform sampling.

We use the LightGCN algorithm for the Gowalla dataset and report results on algorithm convergence and performance in Figure 3. Figure 3b, c show the recall and NDCG values of top 20 recommendations with respect to the number of training epochs. After training the algorithm for 500 epochs, we obtain the recall@20 and NDCG@20 values as 0.186 and 0.153, respectively, which are consistent with the values reported by He et al. (2020).

Figure 4 shows the results of our experiments for the NBPO algorithm using the Amazon dataset. We respectively present the F1-score and the NDCG values in Figure 4a,b for top-*k* recommendations using different *k* values ranging from 2 to 20. We observe that while the recall value peaks at 0.040 for $k = 5$, it starts falling afterwards and becomes 0.038 and 0.032, respectively, for the top-10 and 20 recommendations. On the other hand, NDCG values increase when the *k* value increases for top-*k* recommendations, starting at 0.05 for the top-2 recommendations reaching 0.085 for the top-20 recommendations. These performance values are in line with the results presented by Yu and Qin (2020).

5.2 | Comparative results with the recommendation algorithms

We compare the five recommendation algorithms using a variety of performance metrics including MAP, F1-score and NDCG. We exclude ECML'09 dataset from this comparison as it consists of user-item-tag triplets that are not immediately suitable for NBPO and LightGCN algorithms. In these experiments, we use the default hyperparameters for each algorithm, as discussed in Section 4.2. Table 3 provides the summary results for our analysis. We note that the reported performance values are calculated for top 10 recommendations (i.e. $k = 10$) and obtained using an independent test set for each dataset.

These results show that the best performance for each dataset is either provided by LightGCN or AOBPR algorithms. In particular, AOBPR provides the best overall performance as indicated by the average performance values over six datasets. For the Gowalla dataset, which is one of

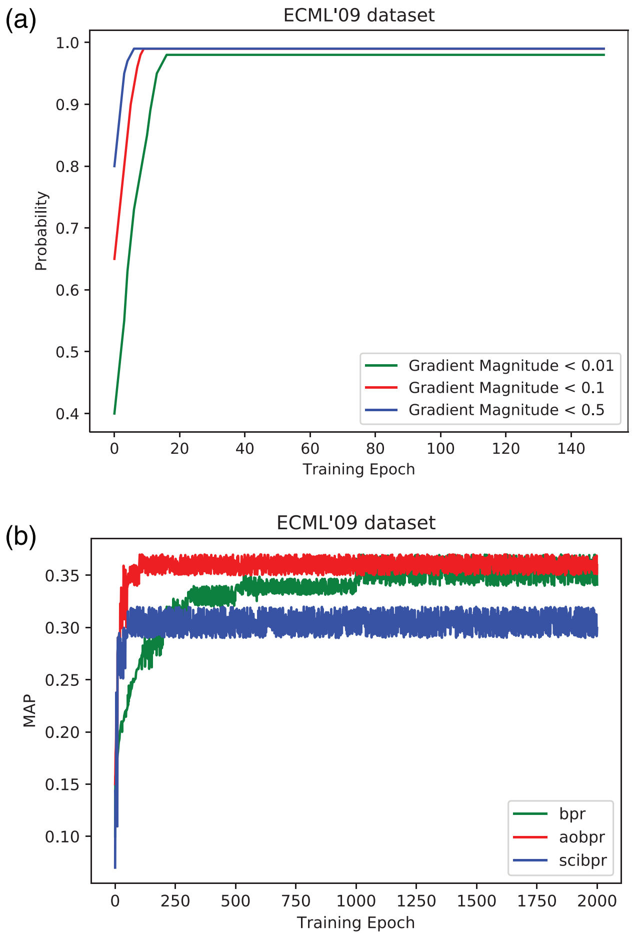


FIGURE 2 Gradient magnitude and mean average precision (MAP) values for Bayesian personalized ranking (BPR) and its extensions. (a) Gradient magnitude values (BPR with uniform sampling). (b) MAP

the datasets used by He et al. (2020), LightGCN performs best in terms of all three performance metrics. For the Frappe dataset, while LightGCN performs best in terms of F1-Score and NDCG values, AOBPR provides the best MAP values. For the remaining four datasets, AOBPR consistently performs the best as agreed by all three performance metrics. We also observe that NBPO algorithm performs substantially worse than AOBPR and LightGCN for all six datasets including the Amazon dataset that was used by Yu and Qin (2020). We note that all algorithms produced better recall scores than precision scores. The precision score closest to the recall score was obtained by LightGCN on the Frappe dataset. In addition, the AOBPR algorithm provides a better precision score than the recall score on the MovieLens dataset.

We note that BPR and AOBPR algorithms perform particularly well for dense datasets such as Jester and Movielens (100 k). For instance, AOBPR performance is impressive for Jester for which F1-score, NDCG and MAP values are 0.538, 0.828 and 0.743, respectively. On the other hand, for Gowalla, which is the most sparse dataset used in our analysis, AOBPR performance is quite poor (e.g. F1-score is 0.085) similar to all other algorithms. Among all the datasets, second best AOBPR performance (in terms of MAP and NDCG) is observed for our propriety Watson dataset. These results are promising for IBM Watson Analytics, which normally provides a recommendation based on a set of rules, to generate more precise recommendation for its users via AOBPR. Providing more precise recommendations to its users may grow the customer base for IBM Watson Analytics and increase its value in the market.

Figure 5 presents F1 scores of the algorithms obtained over fivefold cross-validation. We observe that LightGCN generates very consistent results over different subsets (i.e. folds) for all datasets. The NBPO shows less robustness compared to other algorithms, especially on the Frappe, Gowalla, and Amazon datasets. In addition, for the Amazon dataset, the performances of BPR, AOBPR, and LightGCN are very close to each other. On the other hand, BPR and AOBPR show similar performance except for Frappe and Gowalla datasets. Overall, we note that BPR/AOBPR achieves the best average F1 scores for four of the datasets (Amazon, Watson, MovieLens and Jester), and LightGCN performs best (i.e. in terms of average F1 scores) for the remaining two datasets (Frappe, Gowalla).

We also explore the convergence of the algorithms by examining the changes in MAP values over training epochs. Note that we use different number of training epochs for different datasets due to differences in algorithm convergence speeds by the datasets. Figure 6 shows that AOBPR

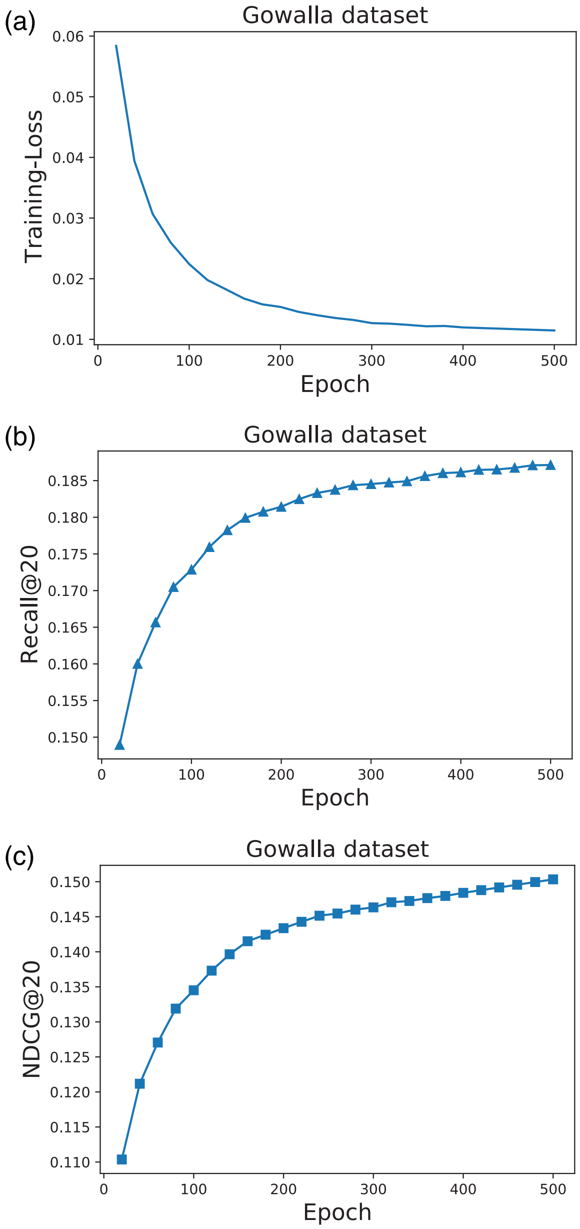


FIGURE 3 LightGCN performance on Gowalla dataset. (a) Training loss. (b) Recall. (c) normalized discounted cumulative gain

converges only marginally faster than BPR in the majority of the datasets except for Gowalla dataset where both overall performance and the convergence speed substantially favours AOBPR. This can be explained by the fact that Gowalla is the largest and most sparse dataset in our experiments, and, accordingly, performance/convergence differences between BPR and AOBPR is more pronounced for Gowalla compared to smaller datasets. We recognize that the algorithm convergence behaviour was primarily investigated by Rendle and Freudenthaler (2014) as AOBPR was in part proposed to improve the convergence speed of BPR (with uniform sampling), and our results mostly confirm their findings. The other algorithms largely follow similar convergence patterns with AOBPR and BPR especially for the datasets for which they were able to generate reasonable recommendations. As expected, for smaller datasets (e.g. MovieLens and Watson) algorithms generally require fewer training epochs to converge.

5.3 | Hyperparameter tuning results

In our hyperparameter tuning experiments, we focus on our propriety dataset, Watson, as the representative dataset since we aim to explore the performance of the recommendation algorithms for data visualization recommendation. Table 4 shows the training times of the algorithms on the

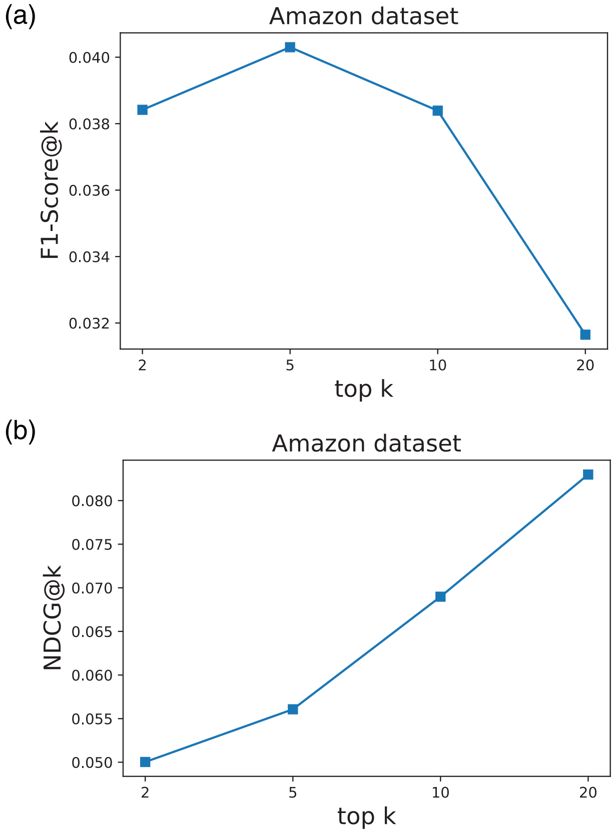


FIGURE 4 Noisy-label robust Bayesian Point-wise optimization performance on Amazon dataset. (a) F1-score. (b) Normalized discounted cumulative gain

Watson dataset. We observe that BPR variants have short training times (e.g. 7.7 s for BPR and 39.4 s for AOBPR), whereas NBPO has significantly long training time (4239.4 s). It is important to note that the provided training time comparison is purely empirical and do not reflect the theoretical complexity of the algorithms. In addition, while the experiments are run in a uniform environment (i.e. using identical CPU and RAM configurations) implementation specifications might contribute to the discrepancies in the training times. For instance, we use a Java implementation for the BPR variants, and python implementations for LightGCN and NBPO. Nonetheless, also considering its low performance along with long training times, we exclude the NBPO algorithm from our hyperparameter tuning experiments.

Hyperparameter tuning experiments are conducted according to various hyperparameter configurations of the recommendation algorithms on the Watson dataset to determine the configuration leading the highest quality recommendations. For the BPR variants, two hyperparameters, namely, learning rate and the number of latent factors are tuned. For the learning rate, the values 0.001, 0.01 and 0.05 are chosen, while for the number of latent factors 10, 20, 30, 64 and 120 are chosen. BPR, AOBPR and SCIBPR algorithms are run to generate results for 15 combinations of these hyperparameters.

For the LightGCN algorithm, we focus on two hyperparameters: learning rate and the regularization parameter. Specifically, we use 0.001, 0.01, and 0.05 for the learning rate and $1e^{-2}$, $1e^{-3}$, $1e^{-4}$, $1e^{-5}$ and $1e^{-6}$ for the regularization parameter. Hyperparameter tuning results are presented in Figure 7 where the black line represents the results with best performing hyperparameter configuration, red dotted line represents the average results, and shaded areas represent the range of MAP values obtained.

The best hyperparameter configuration varies over three BPR variants. The best hyperparameter configuration for SCIBPR consists of the learning rate of 0.01 and the number of latent factors as 64, which leads to an MAP value of 0.20. The best hyperparameter configuration for original BPR consists of the learning rate of 0.01 and the number of latent factors as 30. The highest MAP value for BPR is close to 0.27. The best hyperparameter configuration for AOBPR consists of the learning rate value of 0.05 and the number of latent factors as 64 with an MAP value of 0.29. We note that, for the BPR algorithms, the variation of MAP values are very high according to varying hyperparameter values especially for SCIBPR and AOBPR variants, meaning that those algorithms are very sensitive and require hyperparameter tuning for different datasets. LightGCN algorithm results are largely in line with the hyperparameter tuning results of He et al. (2020), showing that the algorithm is relatively insensitive to changes in the hyperparameters. Although the algorithm is robust, there still exists certain correlations between the

TABLE 3 The comparison of overall performances of the algorithms over top-10 recommendations

Algorithm	Ev. metric	Frappe	Gowalla	Watson	MovieLens	Amazon	Jester	Avg.	Std.
*BPR	Recall	0.288	0.081	0.440	0.182	0.081	**0.801**	0.312	0.276
	Precision	0.115	0.045	0.080	0.279	**0.045**	**0.404**	0.161	0.147
	F1	0.164	0.058	0.135	0.220	0.058	0.537	0.195	0.179
	NDCG	0.232	0.073	0.329	0.214	0.073	**0.828**	0.292	0.281
	MAP	0.144	0.037	0.258	0.103	0.035	**0.743**	0.220	0.269
*SCIBPR	Recall	0.192	0.025	0.251	0.152	0.053	0.789	0.249	0.293
	Precision	0.093	0.012	0.034	0.099	0.015	0.397	0.132	0.202
	F1	0.125	0.016	0.060	0.120	0.023	0.453	0.166	0.242
	NDCG	0.170	0.045	0.168	0.172	0.023	0.652	0.186	0.185
	MAP	0.123	0.009	0.145	0.075	0.015	0.538	0.137	0.164
*AOBPR	Recall	**0.316**	0.108	**0.455**	**0.197**	**0.083**	**0.801**	**0.327**	0.271
	Precision	0.132	0.060	**0.087**	**0.301**	**0.045**	**0.404**	**0.171**	0.147
	F1	0.186	0.077	**0.146**	**0.238**	**0.059**	**0.538**	**0.207**	0.175
	NDCG	0.263	0.099	**0.338**	**0.233**	**0.075**	**0.828**	**0.306**	0.275
	MAP	**0.174**	0.052	**0.266**	**0.115**	**0.036**	**0.743**	**0.231**	0.265
*LightGCN	Recall	0.233	**0.109**	0.020	0.020	0.077	0.143	0.101	**0.081**
	Precision	**0.209**	**0.088**	0.007	0.010	0.045	0.077	0.069	**0.074**
	F1	**0.220**	**0.093**	0.010	0.013	0.057	0.100	0.081	**0.077**
	NDCG	**0.270**	**0.125**	0.020	0.020	0.073	0.149	0.109	**0.095**
	MAP	0.143	**0.060**	0.008	0.005	0.032	0.079	0.055	**0.052**
*NBPO	Recall	0.226	0.097	0.234	0.022	0.096	0.256	0.155	0.096
	Precision	0.100	0.030	0.057	0.009	0.026	0.064	0.048	0.033
	F1	0.139	0.047	0.091	0.011	0.041	0.103	0.072	0.047
	NDCG	0.179	0.070	0.298	0.019	0.069	0.213	0.141	0.106
	MAP	0.061	0.036	0.032	0.003	0.021	0.098	0.042	0.034

hyperparameters values and the MAP. We observed that smaller values of the regularization parameter and the learning rate leads to better MAP values. We identify the best LightGCN configuration for the Watson dataset to be $1e^{-6}$ for the regularization parameter and 0.001 for the learning rate.

6 | THREATS TO VALIDITY

We next discuss the threats to the validity of our study.

For the internal validity aspect, the potential threat of this kind of study might be due to the script for the analysis. Librec—a Java library for recommender systems (Guo et al., 2015) was used for BPR and its variants for this study. SCIBPR method is not included in the Librec, and was constructed following the explanation in the original paper (Rendle & Freudenthaler, 2014). This threat is minimized through proofreading and debugging of the constructed code. For the LightGCN and NBPO algorithms, we used the codes that were shared by the authors, and we only added the MAP metric calculation into the NBPO code.

In the case of statistical validity threats, we ensured that we did not violate any assumptions of the statistical tests used in this work. To mitigate the risk of researcher bias, we strictly followed the original study step-by-step to the extent possible. In the case of a misunderstanding, we communicated with the main author of the original studies and ensured that we clearly understood the methodology and the study approach. Another possible threat that may violate the statistical validity is not performing a cross-validation, and to rely on a single train and test sets. Although we applied fivefold cross-validation for the BPR and its extensions, we did not apply it for the LightGCN and NBPO algorithms. Our main motivation for this was to be consistent with the original studies. This may jeopardize the generalizability of our results, and may cause overfitting or underfitting. We used more than one dataset to overcome this possible threat.

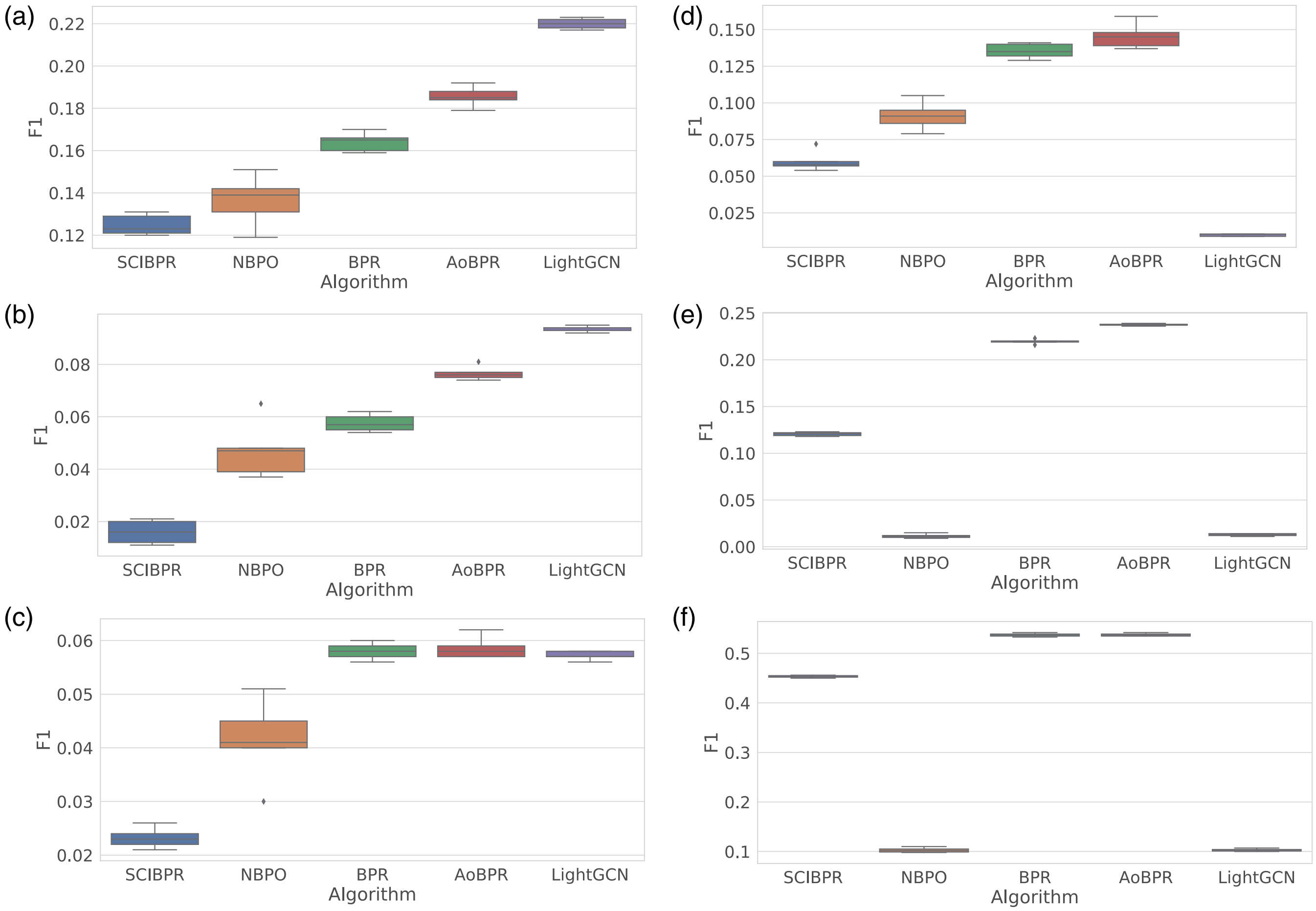


FIGURE 5 F1-scores of the algorithms obtained over fivefold cross-validation. (a) Frappe. (b) Gowalla. (c) Amazon. (d) Watson. (e) MovieLens. (f) Jester

For the external validity we can mention that our results are based on the data collected from real life applications, and they reflect the behaviour of users interacting with the system. Although the original study was conducted on a different type of datasets, we may cautiously extend our conclusion to other applications with different data specifications.

7 | CONCLUSION

In this paper, we applied the BPR algorithm, its extensions, LightGCN, and NBPO algorithms on a propriety dataset, and gain insights about this specific dataset to solve a business problem for IBM Watson Analytics. We replicated and compared these algorithms also using six publicly available datasets to test their generalizability of the presented results in the corresponding studies. That is, our goal was to replicate these approaches to gain a deeper understanding of the performance of the proposed methods on different recommendation applications that generate recommendations using implicit feedback data, and shed some light on the details that might not be provided in the original studies.

We observed similar results with respect to the BPR algorithm and its extensions according to recommendation quality of the algorithms as was reported by Rendle and Freudenthaler (2014). The authors claimed that the AOBPR algorithm would converge faster, while preserving the performance and prediction quality. We only observe the fast convergence of the AOBPR algorithm on the ECML and Gowalla datasets. The reason for this can be the different characteristics of these two datasets compared to other five datasets. For instance, based on item popularity, we observed that ECML and Gowalla datasets are more tailed than others. In addition, they are the largest datasets, therefore, it is easier to observe the fast convergence of the AOBPR algorithm. One of the main findings of this work is that the BPR algorithm and its versions are highly dependent on and sensitive to the hyperparameters for the matrix factorization. Our results show that some specific configurations should be used for each datasets. For instance, the hyperparameter configuration generating the best recommendation results differs among BPR and its extensions, as was observed in our analysis with Watson dataset. On the other hand the we observed that the LightGCN algorithm is more robust to changes in the values of the hyperparameters.

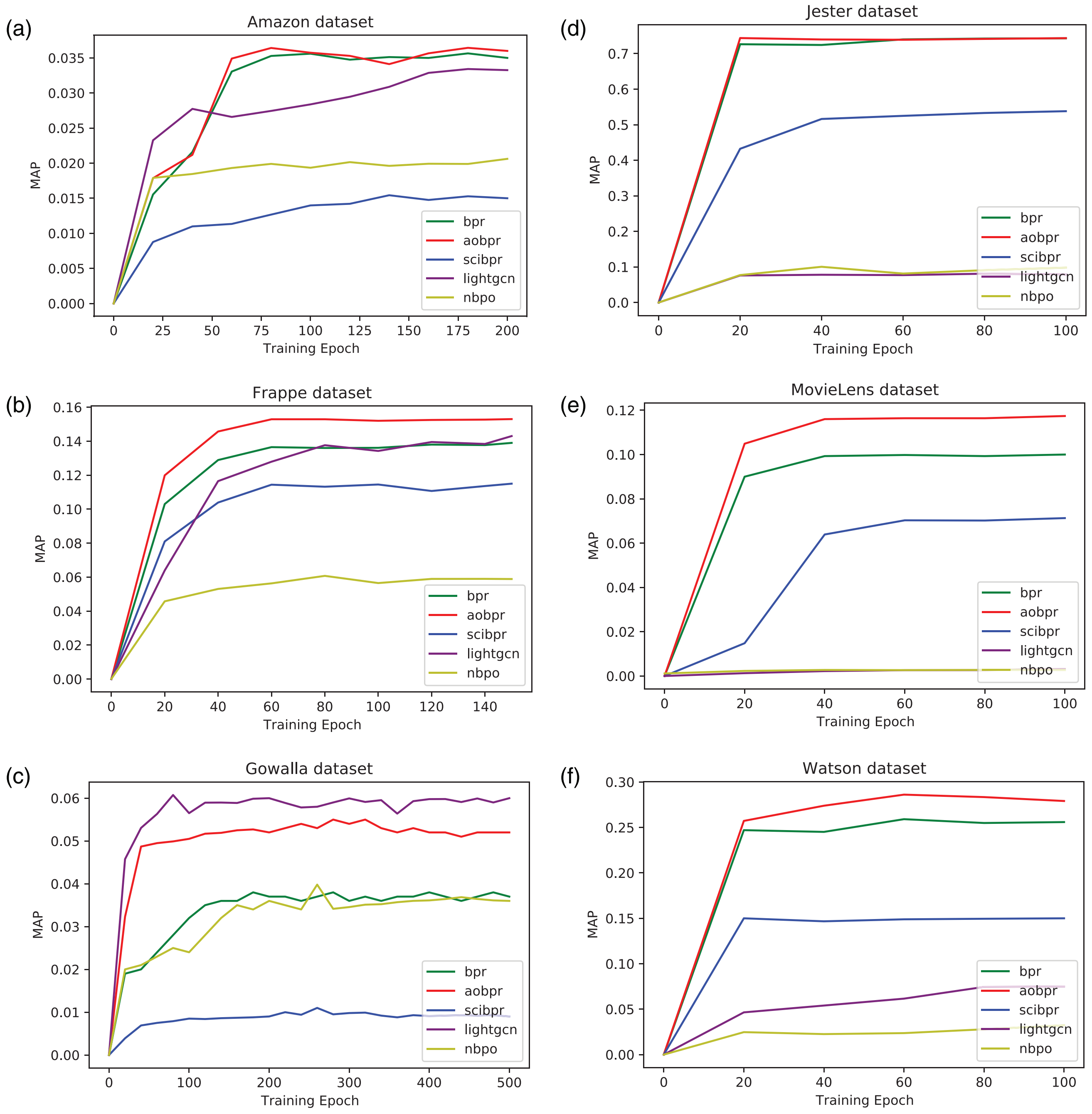


FIGURE 6 Convergence of mean average precision values with respect to number of training epochs

TABLE 4 Training times of the algorithms on the Watson dataset

	BPR	SCIBPR	AOBPR	LightGCN	NBPO
Training time (in seconds)	7.7	9.2	39.4	163.6	4239.4

We note that AOBPR algorithm is able to perform comparably or better than two recent recommendation algorithms, namely, NBPO (Yu & Qin, 2020) and LightGCN (He et al., 2020). LightGCN algorithm outperforms AOBPR only for the Gowalla and Frappe datasets, which might be taken as an evidence for LightGCN algorithm's capability to handle large and sparse datasets. On the other hand, NBPO algorithm consistently showed poor performance compared to other algorithms. These findings point to the importance of independent replication studies such as ours for assessing the overall performance of the notable algorithms in the literature. As supported by our results, conducting such replication studies at certain time intervals and comparing the algorithms developed in a particular field of study on publicly available data sets provide invaluable

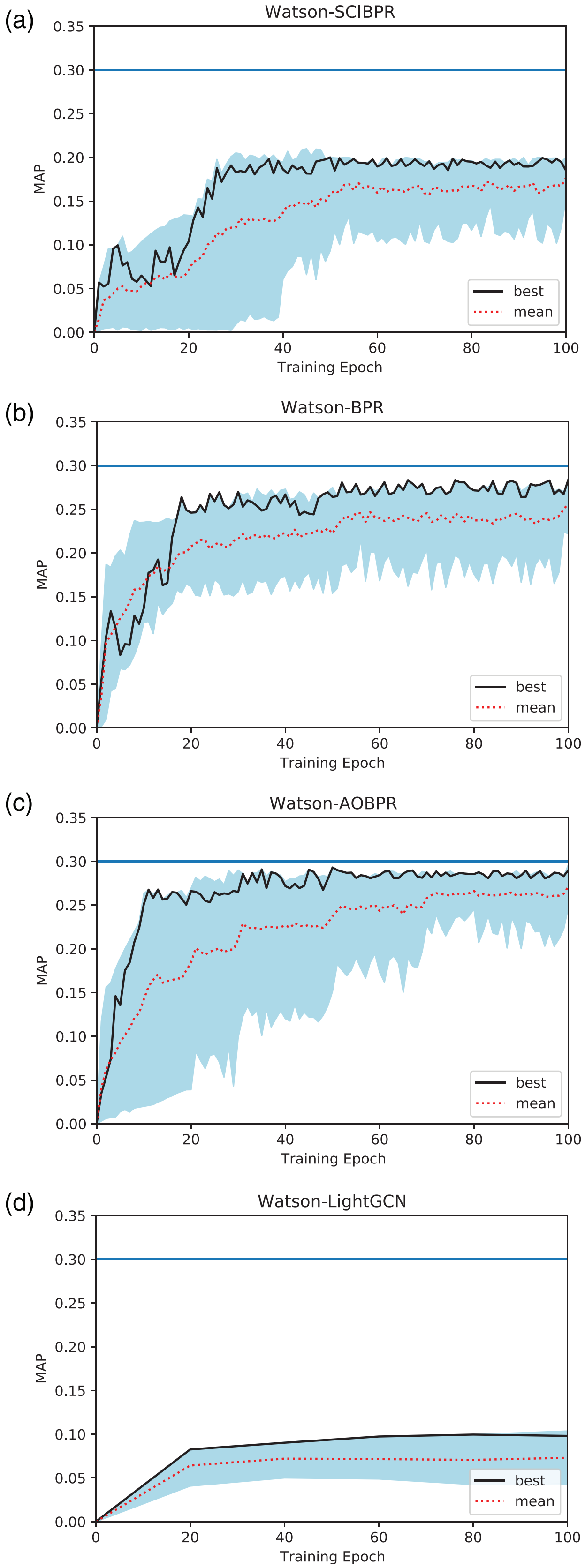


FIGURE 7 Hyperparameter tuning results with Watson dataset

insights for future studies. Our numerical study also points to the relative strengths and weaknesses of the existing methods, for example, through analysing which algorithms perform well for what type of datasets and whether they are sensitive to the algorithm hyperparameters. For instance, we observed that, even though the AOBPR algorithm was proposed much earlier, it can still outperform some of the most recent methods, yet its performance can be sensitive to hyperparameters. In addition, our detailed numerical results with a variety of datasets and performance metrics can be used as benchmarks for future studies.

There are some limitations to our study. First, while we considered some of the most recent algorithms in this field, our study does not include few other algorithms, namely, IRGAN (J.Wang et al., 2017), AMF (He et al., 2018), and AdVIR (Park & Chang, 2019). This is mainly because either the implementations of these methods are not publicly available or re-implementation is not trivial due to the complex algorithmic structures. Second, even though we closely followed the settings provided in the corresponding papers, there might be some minor differences in our implementations and settings when compared to the original studies. While we were able to replicate the reported results, algorithm performances for the new datasets can be potentially improved with certain adjustments in the settings and hyperparameters.

In our future works, we aim to extend our analysis to other datasets from different domains. Seven datasets considered in this study involve application problems in movie recommendation (MovieLens), marketing (Amazon), data visualization (Watson), point of interest recommendation (Gowalla), mobile application recommendation (Frappe), joke recommendation (Jester), and social tagging (ECML). Similarly, recommender systems designed for implicit feedback can be used for other areas such as news, song, video, event and website recommendation. We also aim to investigate alternative strategies to the non-uniform sampling approaches that provide more informative samples for the ranking task. In addition, other recommender system algorithms such as IRGAN (J.Wang et al., 2017), AMF (He et al., 2018), and AdVIR (Park & Chang, 2019) can be implemented and made public to facilitate future comparative analysis. Among these approaches, the performances of AdVIR (Park & Chang, 2019) and AMF (He et al., 2018) were compared to BPR (Rendle et al., 2009) and IRGAN (J. Wang et al., 2017), and they reportedly performed better than these methods. However, they were not compared among themselves and also with the other algorithms that we considered in this study, which can be the subject of future research.

ACKNOWLEDGEMENTS

This research is funded in part by IBM CAS Project 919, and NSERC CRD Grant 490782 2015.

CONFLICT OF INTEREST

The authors declare no conflict of interests.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from IBM US. Restrictions apply to the availability of these data, which were used under license for this study. Data are available from the author(s) with the permission of IBM US.

ORCID

Aysun Bozanta https://orcid.org/0000-0002-1768-6278

ENDNOTES

[i] ECML'09 properties are presented in the text since it has different components, which are not compatible with the table structure.

[ii] https://www.kde.cs.uni-kassel.de/wp-content/uploads/ws/dc09/dataset.html

[iii] https://github.com/kuandeng/LightGCN/tree/master/Data/gowalla

[iv] https://github.com/Wenhui-Yu/NBPO/tree/master/dataset/amazon

[v] https://www.baltrunas.info/context-aware

[vi] https://grouplens.org/datasets/movielens/

[vii] http://eigentaste.berkeley.edu/dataset/

[viii] http://www.libfm.org/

[ix] https://guoguibing.github.io/librec/index.html

[x] https://github.com/kuandeng/LightGCN

[xi] https://github.com/Wenhui-Yu/NBPO

REFERENCES

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749.

Agosti, M., Di Buccio, E., Ferro, N., Masiero, I., Peruzzo, S., & Silvello, G. (2012). Directions: design and specification of an ir evaluation infrastructure. *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 88–99.

Agosti, M., Di Nunzio, G. M., & Ferro, N. (2006). Scientific data of an evaluation campaign: Do we properly deal with them? *Proceedings of the Workshop of the Cross-Language Evaluation Forum for European Languages*, pp. 11–20.

Agosti, M., Ferro, N., & Thanos, C. (2012). Desire 2011: Workshop on data infrastructures for supporting information retrieval evaluation. In *ACM SIGIR Forum* (Vol. 46, pp. 51–55). ACM.

Baltrunas, L., Church, K., Karatzoglou, A., & Oliver, N. (2015). Frappe: Understanding the usage and perception of mobile app recommendations in-the-wild. arXiv preprint arXiv:1505.03014.

Berg, R., Kipf, T. N., & Welling, M. (2017). Graph convolutional matrix completion. arXiv preprint arXiv:1706.02263.

Callan, J., & Moffat, A. (2012). Panel on use of proprietary data. In *ACM SIGIR Forum* (Vol. 46, pp. 10–18). ACM.

Carver, J. C. (2010). Towards reporting guidelines for experimental replications: A proposal. *Proceedings of the 1st International Workshop on Replication in Empirical Software Engineering*.

Claypool, M., Le, P., Wased, M., & Brown, D. (2001). Implicit interest indicators. *Proceedings of the 6th International Conference on Intelligent User Interfaces* (pp. 33–40).

Çoba, L., & Zanker, M. (2017). Replication and reproduction in recommender systems research-evidence from a case-study with the rrecsys library. *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 305–314).

Das, D., Sahoo, L., & Datta, S. (2017). A survey on recommendation system. *International Journal of Computer Applications*, *160*(7), 6–10.

Di Buccio, E., Di Nunzio, G. M., Ferro, N., Harman, D., Maistro, M., & Silvello, G. (2015). Unfolding off-the-shelf ir systems for reproducibility. *Proceedings of the SIGIR Workshop on Reproducibility, Inexplicability, and Generalizability of Results* (RIGOR 2015).

Dibia, V., & Demiralp, Ç. (2019). Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. *IEEE Computer Graphics and Applications*, *39*(5), 33–46.

Ferro, N. (2017). Reproducibility challenges in information retrieval evaluation. *Journal of Data and Information Quality (JDIQ)*, *8*(2), 8.

Ferro, N., Fuhr, N., Järvelin, K., Kando, N., Lippold, M., & Zobel, J. (2016). Increasing reproducibility in ir: Findings from the dagstuhl seminar on reproducibility of data-oriented experiments in e-science. In *ACM SIGIR Forum* (Vol. 50, pp. 68–82). ACM.

Gantner, Z., Drumond, L., Freudenthaler, C., & Schmidt-Thieme, L. (2012). Personalized ranking for non-uniformly sampled items. *Proceedings of the KDD Cup 2011* (pp. 231–247).

Gantner, Z., Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. (2011). Mymedialite: A free recommender system library. *Proceedings of the Fifth ACM Conference on Recommender Systems* (pp. 305–308).

Guo, G., Xhang, J., Sun, Z., & Yorke-Smith, N. (2015). Librec: A java library for recommender systems. *Posters Demos, Late-Breaking Results and Workshop Proceedings of the 23rd Conference on User Modelling, Adaptation and Personalization* (UMAP) (pp. 1–4).

Hanbury, A., Müller, H., Balog, K., Brodt, T., Cormack, G. V., Eggel, I., … others (2015). Evaluation-as-a-service: Overview and outlook. arXiv preprint arXiv: 1512.07454.

Harper, F. M., & Konstan, J. A. (2016). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *5*(4), 19.

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., &Wang, M. (2020). Lightgcn: Simplifying and powering graph convolution network for recommendation. arXiv preprint arXiv:2002.02126.

He, X., He, Z., Du, X., & Chua, T.-S. (2018). Adversarial personalized ranking for recommendation. *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (pp. 355–364).

Hsieh, C.-J., Natarajan, N., & Dhillon, I. (2015). Pu learning for matrix completion. *Proceedings of the International Conference on Machine Learning* (pp. 2445–2453).

Hu, K., Bakker, M. A., Li, S., Kraska, T., & Hidalgo, C. (2019). Vizml: A machine learning approach to visualization recommendation. *Proceedings of the 2019 Chi Conference on Human Factors in Computing Systems* (pp. 1–12).

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. *Proceedings of the Data mining, 2008. Eighth IEEE International Conference on ICDM'08*. (pp. 263–272)

Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). *Recommender systems: An introduction*. Cambridge University Press.

Jawaheer, G., Weller, P., & Kostkova, P. (2014). Modeling user preferences in recommender systems: A classi_cation framework for explicit and implicit user feedback. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *4*(2), 8.

Kazai, G., & Fuhr, A. R. N. (2011). *Advances in information retrieval*. Springer.

Kipf, T. N., &Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Lak, P., Kavaklioglu, C., Sadat, M., Petitclerc, M., Miranskyy, A. V., Wills, G., & Bener, A. B. (2017). A probabilistic approach for modelling user preferences in recommender systems: A case study on IBM watson analytics. *Proceedings of the 27th Annual International Conference on Computer Science and Software Engineering 2017* (pp. 38-47).

Lak, P., Sadat, M., Barrelet, C. J., Petitclerc, M., Miranskyy, A. V., Statchuk, C., & Bener, A. B. (2016). Preliminary investigation on user interaction with ibm Watson analytics. *Proceedings of the 26th Annual International Conference on Computer Science and Software Engineering 2016* (pp. 218–225).

Liang, D., Krishnan, R. G., Hoffman, M. D., & Jebara, T. (2018). Variational autoencoders for collaborative filtering. *Proceedings of the 2018 World Wide Web Conference* (pp. 689–698).

Liu, T.-Y. (2009). Learning to rank for information retrieval. Foundations and trends®. *Information Retrieval*, *3*(3), 225–331.

Luo, Y., Qin, X., Tang, N., & Li, G. (2018). Deepeye: Towards automatic data visualization. *Proccedings of the 2018 ieee 34th international conference on data engineering (ICDE)* (pp. 101–112).

Mu, R. (2018). A survey of recommender systems based on deep learning. *IEEE Access*, *6*, 69009–69022.

Najafabadi, M. K., Mohamed, A. H., & Mahrin, M. N. (2019). A survey on data mining techniques in recommender systems. *Soft Computing*, *23*(2), 627–654.

Oard, D.W., Kim, J., (1998). Implicit feedback for recommender systems. *Proceedings of the AAAI Workshop on Recommender Systems* (Vol. *83*).

Park, D. H., & Chang, Y. (2019). Adversarial sampling and training for semi-supervised information retrieval. *Proceedings of the World Wide Web Conference* (pp. 1443–1453).

Peska, L., & Vojtas, P. (2017). Using implicit preference relations to improve recommender systems. *Journal on Data Semantics*, *6*(1), 15–30.

Polatidis, N., Kapetanakis, S., Pimenidis, E., & Kosmidis, K. (2018). Reproducibility of experiments in recommender systems evaluation. *Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations* (pp. 401–409).

Polatidis, N., Pimenidis, E., Fish, A., & Kapetanakis, S. (2019). A guideline-based approach for assisting with the reproducibility of experiments in recommender systems evaluation. *International Journal on Artificial Intelligence Tools*, *28*(08), 1960011.

Rao, N., Yu, H.-F., Ravikumar, P. K., & Dhillon, I. S. (2015). Collaborative filtering with graph information: Consistency and scalable methods. *Advances in Neural Information Processing Systems*, *28*, 2107–2115.

Rendle, S. (2012). Factorization machines with libFM. *ACM trans. International Journal of Intelligent*, *3*(3), 1–22.

Rendle, S., & Freudenthaler, C. (2014). Improving pairwise learning for item recommendation from implicit feedback. *Proceedings of the 7th ACM International Conference on Web Search and Data Mining* (pp. 273–282).

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (pp. 452–461).

Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems: Introduction and challenges. In *Recommender systems handbook* (pp. 1–34). Springer.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web* (pp. 285–295).

Schafer, J. B., Konstan, J., & Riedl, J. (1999). Recommender systems in e-commerce. *Proceedings of the 1st ACM Conference on Electronic Commerce* (pp. 158–166).

Shull, F. J., Carver, J. C., Vegas, S., & Juristo, N. (2008). The role of replications in empirical software engineering. *Empirical Software Engineering*, *13*(2), 211–218.

Wang, J., Yu, L., Zhang, W., Gong, Y., Xu, Y., Wang, B., Zhang, P., Zhang, D. (2017). Irgan: A minimax game for unifying generative and discriminative information retrieval models. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 515–524).

Wang, X., He, X., Wang, M., Feng, F., & Chua, T.-S. (2019). Neural graph collaborative filtering. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 165–174).

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph convolutional neural networks forweb-scale recommender systems. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 974–983).

Yu, W., & Qin, Z. (2020). Sampler design for implicit feedback data by noisy-label robust learning. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 861–870).

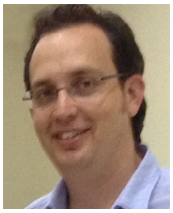
Zhang, G., Liu, Y., & Jin, X. (2020). A survey of autoencoder-based recommender systems. *Frontiers of Computer Science*, *14*(2), 430–450.

Zobel, J., Webber, W., Sanderson, M., & Moffat, A. (2011). Principles for robust evaluation infrastructure. *Proceedings of the 2011 Workshop on Data Infrastructures for Supporting Information Retrieval Evaluation* (pp. 3–6).

AUTHOR BIOGRAPHIES

Parisa Lak received her PhD from Data Science Laboratory, Department of Mechanical and Industrial Engineering, Ryerson University, Toronto, Canada. Parisa is also IBM's research partner being involved in IBM Watson Analytics projects. Her research focus is on topics related to machine learning, data mining and information retrieval and more specifically on Recommender Systems. She received her MBA from Sharif University of Technology and MMSc in Management of Innovation and Technology from Ted Rogers School of Management at Ryerson University. She is also a member of IEEE and ACM Special Interest Group of Information Retrieval.

Aysun Bozanta is a post-doctoral fellow at Data Science Laboratory, Department of Mechanical and Industrial Engineering, Ryerson University, Toronto, Canada. She received her Ph.D. and MA degrees from Management Information Systems, Bogazici University, Istanbul, Turkey. Her primary research interests are recommender systems, data analysis and reinforcement learning.



Can Kavaklioglu is a PhD candidate at Data Science Laboratory, Department of Mechanical and Industrial Engineering, Ryerson University, Toronto, Canada. He became a member of IEEE in 2007. His PhD thesis work is on computational aspects of Tensor Factorization.

Mucahit Cevik is an assistant professor in the Faculty of Engineering, Ryerson University. His current research focus is on reinforcement learning and its applications in healthcare. He is a member of INFORMS.

Ayse Basar is a professor and the director of Data Science Laboratory (DSL) in the Faculty of Engineering, Ryerson University. She is the director of Big Data in the Office of Provost and Vice President Academic at Ryerson University. She is a faculty research fellow of IBM Toronto Labs Centre for Advance Studies, and affiliate research scientist in St. Michael's Hospital in Toronto. Her current research focus is big data applications to tackle the problem of decision-making under uncertainty by using Bayesian machine learning methods to analyse complex structures in data to build recommender systems and predictive models. She is a member of AAAI, INFORMS, AIS, a senior member of IEEE.

Martin Petitclerc recently joined Coveo as a data scientist. He was previously senior software architect at IBM Canada, part of the Watson Analytic team. He did major contributions to the underline natural language processing (NLP) and data access layer and played key role in those areas, including large number of cognitive features. He worked on various business intelligence and business analytics products, over some of the world's leaders in the related domain. He led R&D technologies such as OLAP databases, reporting tools, and cognitive systems for more than 20 years, owning multiple related patents and publications. In addition, he holds a bachelor degree in Management Information system (1994—Laval University).

Graham Wills creates methods for interacting with data to find patterns and features. He has a PhD in Statistics and was a researcher at AT&T Bell Labs before joining SPSS in 2001, where he formed a statistical graphics visualization team. He was technical lead for the development of IBM's RAVE, an interactive web-based visualization system used throughout IBM, and is now responsible for developing cognitive numeric solutions in IBM Analytics. He has given many seminars and talks, and has twice chaired the IEEE InfoVis conference. He has a dozen patents, has written over 60 articles and a book, "Visualizing Time", is co-author of the open source vis project Brunel (brunelvis.org) and is an amateur actor, director and playwright.

How to cite this article: Lak, P., Bozanta, A., Kavaklioglu, C., Cevik, M., Basar, A., Petitclerc, M., & Wills, G. (2021). A replication study on implicit feedback recommender systems with application to the data visualization recommendation. *Expert Systems*, e12871. https://doi.org/10.1111/exsy.12871