User Modeling and User-Adapted Interaction (2022) 32:747–780
https://doi.org/10.1007/s11257-021-09303-w

NFC: a deep and hybrid item-based model for item cold-start recommendation

Cesare Bernardis[1] · **Paolo Cremonesi**[1]

Received: 11 March 2021 / Accepted in revised form: 18 September 2021 / Published online: 20 October 2021

Abstract

New items, also called cold-start items, are introduced every day in the catalogs of numerous online systems. Due to the absence of previous preferences, recommending these items is difficult but important task for a recommender system. For this reason, the item cold-start recommendation problem still represents an interesting research topic for the community. In this work, we propose Neural Feature Combiner (NFC), a novel deep learning, item-based approach for cold-start item recommendation. The model learns to map the content features of the items into a low-dimensional hybrid embedding space. The features that compose the embeddings are then combined in order to reproduce collaborative item similarity values. We compare NFC with three variants of the same model that learn from user feedback, showing the advantages of learning from similarities in terms of accuracy and convergence time. With an extensive set of experiments on four datasets, we show that NFC outperforms several cutting edge approaches in the top-n recommendation task of cold-start items. Results in extremely cold (i.e., with a very low amount of interactions for training) and cold-warm hybrid scenarios prove that NFC effectively exploits collaborative information, leading to state-of-the-art accuracy. We finally conduct a qualitative analysis of the embeddings generated by different models, and we provide an analysis of the importance that different models assign to the input features, empirically demonstrating the robustness of the hybrid representations produced by our new model.

Keywords Recommender systems · Cold-start · Item-based · Deep learning · Hybrid

✉ Cesare Bernardis
cesare.bernardis@polimi.it

Paolo Cremonesi
paolo.cremonesi@polimi.it

[1] DEIB, Politecnico di Milano, Milan, Italy

1 Introduction

New items are daily introduced in the catalogs of a number of online systems. Recommending these items is a non-trivial task of high interest, due to its relevance in the industrial environment. In this scenario, the absence of user feedback prevents recommender systems from producing recommendations based on the commonalities between the usage patterns of different users, known as *collaborative information*. Consequently, recommenders are forced to exploit alternative sources of information about these new items, also called *cold-start items*, often represented by the attributes or features that describe them, known as *content information*. However, the accuracy of these techniques is strongly dependent on the quality of the available features, and it is usually not comparable with the performance of algorithms that take advantage of past interactions between users and items (Lops et al. 2011; Desrosiers and Karypis 2011).

In the last years, several approaches have been specifically designed to face the cold-start item recommendation problem. Most of them propose hybrid approaches that transduce collaborative information about *warm items* (i.e., items that received preferences from users) to new items, leveraging content information as a means, in order to enhance the accuracy in the cold-start scenario (Gantner et al. 2010; Barkan et al. 2019; Barjasteh et al. 2015; Volkovs et al. 2017; Sharma et al. 2015; Bernardis et al. 2018; Saveski and Mantrach 2014; Elbadrawy and Karypis 2015; Wei et al. 2017; Cella et al. 2017; Dacrema et al. 2018). The most simple methods provide solutions for feature weighting that assess the importance of each content feature depending on the relationships between content and collaborative information on warm items. Other more complex models learn to map item features into a hybrid, low-dimensional embedding space, to exploit also combinations of features, which can be crucial to improve the accuracy.

A family of recent feature weighting techniques proposes to learn feature weights extracting collaborative information from a collaborative item-based similarity matrix instead of user-item interactions directly (Dacrema et al. 2018; Cella et al. 2017; Bernardis et al. 2018). Item similarity matrices based on collaborative information express the similarity between two items according to their respective feedback patterns: a high similarity score indicates that most of the users that provided positive feedback for one of the two items also provided positive feedback for the other one. Dacrema et al. (2018) assert that using similarity values as learning targets has a number of advantages over learning directly from the user-item interactions: similarity matrices are denser than user rating matrices; the item-based model used to generate the matrix can be chosen properly, depending on the characteristics required; the models are simpler and converge faster.

Inspired by this family of approaches, in this paper we propose a new item-based, deep learning model for cold-start item recommendation called Neural Feature Combiner (NFC), that learns to reproduce a given collaborative item similarity matrix using only the content representations of the items. The model takes as input the content features of an item, maps them in a low-dimensional embedding space, and generates the similarity values between that item and all the warm items in the

dataset. Different from previous techniques, the deep learning architecture enables the exploitation of nonlinear combinations of features, and the structure of the model allows to weigh them differently, depending on the item the similarity refers to. Moreover, NFC is able to predict the similarities between a new, cold item and all the warm items leveraging only the content representation of the new one.

We also propose three variants of NFC that learn from user-item interactions directly, using different optimization techniques, in order to assess the differences between learning from similarities or user ratings. With extensive experiments over four datasets, we show that the main version which learns from collaborative similarities largely outperforms the other variants in both accuracy and convergence time. We confront NFC with several baselines and state-of-the-art techniques for cold-start item recommendation, and we empirically demonstrate its superior ranking capabilities. We perform a sensitivity analysis varying the number of interactions available during the training process, and we report a comparative study in a cold-warm hybrid scenario similar to a real environment, showing the ability of NFC to effectively exploit collaborative information. Finally, we perform a qualitative study of the embeddings on the Yahoo! Movies dataset, comparing the representations provided by our model against those provided by a group of state-of-the-art approaches for cold-start recommendation. We conduct an analysis of the item neighborhoods generated by these models in the embedding space they learn, and we examine the importance they assign to the content features, to provide a clearer explanation of the outcomes produced by the models. For the sake of reproducibility, we also release the source code used to perform all the experiments.

The rest of the paper is organized as follows. In Sect. 3, we present some state-of-the-art techniques for cold-start recommendation. In Sect. 4, we describe our new NFC model. In Sect. 5, we propose the variants of NFC that learn from user feedback instead of similarity values. In Sect. 6, we describe the datasets and all the algorithms that we used in our experiments. In Sect. 7, we report the results in the cold-start scenarios. In particular, we assess the advantages of learning from similarity values, and we show the results of the comparison between NFC and the other state-of-the-art techniques. In Sect. 8, we provide the results in the cold-warm hybrid scenario. In Sect. 9, we show the results of our qualitative analysis of the model. In Sect. 10, we discuss the limitations of our work. We conclude the paper with some final remarks.

2 Related works

Recommending cold-start items is a very well-known problem in Recommender Systems literature, and many different approaches have been specifically proposed to face it. The lack of collaborative information about cold items led to the implementation of techniques that focus on the content information of the items, which is always available a priori, also for new items. One of the most simple and common techniques that exploit content information is the item nearest neighbors approach (Desrosiers and Karypis 2011). The content of each item is represented as a vector of features, and a similarity function, like cosine similarity or Pearson correlation,

is defined between couples of items. The algorithm recommends the most similar items to those in the user's profile.

To improve the accuracy performance of this branch of approaches, a variety of feature weighting algorithms have been proposed. In these techniques, a real value is assigned to every feature in order to estimate its importance. Filtering methods, like TF-IDF (Luhn 1957; Sparck Jones 1988) and BM25 (Robertson et al. 1994), are very common in information retrieval and try to improve the performance of content-based algorithms assessing the relevance of features from their statistical distribution among items. The idea behind these weighting techniques is that the importance of a feature is directly proportional to the frequency with which it appears in an item but inversely proportional to its popularity.

The main drawback of these methods is that they ignore the collaborative information available between different users, and they consider only the interaction history of the user involved in the recommendation. For this reason, they are also called *non-collaborative* user personalized models. A branch of more complex approaches called embedded methods, exploits this type of information to estimate the relevance of item features. User-specific Feature-based Similarity Models (UFSM) in Elbadrawy and Karypis (2015) is a sparse, high-dimensional latent factor model that learns multiple global similarity functions for estimating the similarity between items based on their content representations. These similarities are combined together with user-specific weights to provide highly personalized recommendations. The model is trained either minimizing a squared error loss function or adopting a Bayesian Personalized Ranking (BPR) optimization procedure. Collaborative boosted Feature Weighting (CFW) in Dacrema et al. (2018) proposes a feature weighting technique equivalent to Least-squares Feature Weighting presented in Cella et al. (2017). The main characteristic of the model is that it learns weights from item-based collaborative similarity matrices, instead of ratings, a technique that has several advantages: flexibility in the selection of the collaborative algorithm; similarity values are less noisy than user interactions; faster convergence. Similar to CFW, HP^3 in Bernardis et al. (2018) is a feature weighting approach that exploits the advantages of learning from similarity matrices in a graph-based model.

Feature weighting models are usually simple and fast to train. However, they do not take into account nonlinear combinations of features, which contribution might be very important to model user preferences correctly. To overcome this drawback, FBSM (Sharma et al. 2015) uses a bilinear model to capture pairwise dependencies between the features. Proposed as an evolution of UFSM, it learns a global, bilinear similarity function between the content representations of couples of items. Moreover, in order to reduce the high number of parameters introduced by bilinearity, it models pairwise relations as a combination of a linear component and a low-rank component.

Another family of approaches for item cold-start recommendation is based on matrix factorization techniques. These methods learn to map items' or users' features into a low-dimensional embedding space. DropoutNet (Volkovs et al. 2017) proposes a neural network-based latent approach explicitly trained for cold-start that can be applied on top of any existing latent model. The authors observe that the cold-start scenario is equivalent to the missing data problem where preference

information is missing. By applying dropout during training, they condition the neural network to generalize for missing input. Gantner et al. (2010) learn how to map attributes of users and items to latent features of a matrix factorization model optimized for BPR. Exploiting the learned mappings, the model can provide latent representations also for new users and new items, retaining a competitive predictive accuracy and recommendation celerity. Local Collective Embeddings in Saveski and Mantrach (2014) is a matrix factorization method that exploits items' properties and past user preferences while enforcing the manifold structure exhibited by the collective embeddings. Given the content representation of a new item, the model projects it in a hybrid low-dimensional space and infers the users that are most likely to be interested in it. Decoupled completion and transduction in Barjasteh et al. (2015) is a matrix factorization-based approach that simultaneously exploits the similarity information among users and items to alleviate the cold-start problem. It decouples the completion of a rating sub-matrix, generated by excluding cold-start items, and the transduction of knowledge to cold-start items using side information. CB2CF (Barkan et al. 2019) is a deep neural model designed ad hoc for the Microsoft Store that learns a multiview mapping of content-based item representations to collaborative factorized ones, produced by a BPR approach. It is capable of encoding different types of content information, including categorical, numerical, continuous, or unstructured textual data.

Alongside all the techniques specifically designed for cold-start recommendation, there are also other more generic approaches that have been successfully employed for the same task (Pan et al. 2019; Lian et al. 2017). The factorization machine in Rendle (2010) is a general predictor that models all nested variable interactions, like a polynomial kernel in Support Vector Machine. Using a factorized parametrization, the factorization machine can be computed in linear time and employs a linear number of parameters. The Wide and Deep approach in Cheng et al. (2016) exploits the potential of deep learning and jointly trains wide linear models and deep neural networks. The deep part helps to generalize to previously unseen user-item feature couples, learning low-dimensional dense embeddings. On the contrary, the linear part, with the introduction of cross-product feature transformations, prevents the model from over-generalizing, due to the high sparsity of user-item interactions.

The NFC model that we propose merges and exploits several strengths of the state-of-the-art recommendation algorithms:

- CFW and HP^3 are two feature weighting techniques that learn the model parameters using collaborative item similarities as targets during training. NFC adopts the same strategy, but it is a more powerful technique compared to a feature weighting method since it is able to model also nonlinearities and combinations of features.
- FBSM and UFSM learn similarity models for item cold-start recommendation, taking into account also bilinear relations between features. Similarly, NFC learns a similarity model, but it is able to exploit multiple combinations and nonlinear relations among item content features. Moreover, it is trained using collaborative similarity values instead of user-item interactions directly.

- CB2CF takes advantage of the expressive power of deep learning to enable the embedding of completely cold items, learning to map item content features to a factorized, collaborative item representation. Likewise, NFC exploits the potential of deep learning to map content features into low-dimensional embedded representations of the items. However, differently from CB2CF, it generates a similarity model, and it is trained using collaborative item similarities.

Most important, to the best of our knowledge, NFC is the first algorithm that directly maps a vector of features – of any item, including cold-start items – with the similarity to any other warm item. On the contrary, all other methods in the literature learn feature weights, either explicit feature weights, such as CFW and HP^3, or embedded feature weights, such as FBSM and UFSM. Feature weights are later used to compute the similarity between cold and warm items.

3 Model

3.1 Notation

Let $\mathcal{U}, \mathcal{I}, \mathcal{F}$ represent, respectively, the set of users, items, and item features. The set $\mathcal{I}$ is composed of two disjoint subsets: the set of cold items $\mathcal{I}_c$, and the set of warm items $\mathcal{I}_w$.

Matrix $\mathbf{R}$ of size $|\mathcal{U}| \times |\mathcal{I}|$ is the User Rating Matrix (URM). If the URM contains explicit feedback, each entry r_{ui} represents the rating value given by user u to item i, while it contains 0 if the user did not provide any feedback on the item. If the URM contains implicit feedback, r_{ui} assumes a binary value (either 0 or 1), where value 1 is assigned when the user expressed positive feedback, while 0 is assigned otherwise.

Matrix $\mathbf{A}$ of size $|\mathcal{I}| \times |\mathcal{F}|$ represents the Item Content Matrix (ICM), where each entry a_{if} assumes value 1 if a certain item i has the feature f, 0 otherwise.

Matrix $\mathbf{S}$ of size $|\mathcal{I}| \times |\mathcal{I}|$ represents a generic item-based similarity matrix, where each entry s_{ij} in $\mathbf{S}$ assesses how similar items i and j are.

3.2 Neural feature combiner

In this paper, we propose Neural Feature Combiner (NFC), an item-based approach that employs a deep learning architecture to generate the item similarity matrix. The NFC model structure can be divided into two main components. The first takes as input the content-based representation $\mathbf{a}_i$ of a given item $i \in \mathcal{I}$ and generates an embedded representation $\mathbf{z}_i$ in a low-dimensional space. We refer to this component as the *embedding network*. The second component of the model takes as input the embedded representation $\mathbf{z}_i$ of an item and combines the embedded features that compose it in order to generate the similarity scores s_{ji} between all the warm items

$j \in \mathcal{I}_w$ and item i. We refer to this part of the architecture as the *combination network*. Finally, the preferences are predicted as a common item-based model:

$$\tilde{r}_{ui} = \sum_{j}^{\mathcal{I}_w} r_{uj} \cdot s_{ji} \tag{1}$$

Note that if user u did not provide any feedback for item j, then $r_{uj} = 0$ according to the definition in Sect. 4.1. Consequently, the summation in Eq. (1) is performed exploiting only the similarities related to the warm items for which u provided feedback. The complete structure of NFC is graphically shown in Fig. 1. The two components of the network are discussed more in detail in the following sections.

The model is trained end-to-end using a collaborative similarity matrix as target, minimizing the sum of squared errors loss L with L2 regularization defined as:

$$L = \sum_{i}^{\mathcal{I}_w} \sum_{j}^{\mathcal{I}_w} (s_{ij}^{CF} - s_{ij})^2 + \lambda ||\theta||^2 \tag{2}$$

where θ represents the parameters of the model. During this procedure, the content features of warm items are provided as input to the model, and the respective collaborative similarities are used as learning targets to reproduce. NFC learns to encode the content representation of an item into a hybrid embedding, as it is generated using content information as source and collaborative information as target, and to combine the latent features that compose it in order to predict the similarity values. Note that, as shown in (1), in an item-based model only warm items contribute to the user preference prediction, since if j is a cold item, then $r_{uj} = 0$ by definition.

During the recommendation phase, the model is able to estimate also the similarity values between all the warm items and the new cold items. Indeed, in the inference process, NFC requires as input only the content features of an item, which is the only type of information available about cold-start items. This allows providing personalized recommendations also in an item cold-start scenario.

3.2.1 Embedding network

The embedding network is a multi-layer, fully connected neural network that transforms the content representation of an item $\mathbf{a}_i$, under the form of content features, in a low-dimensional, hybrid embedding $\mathbf{z}_i$. It can be defined as:

$$\begin{cases} \mathbf{o}_i^1 = g_1(\mathbf{W}_1 \mathbf{a}_i + \mathbf{b}_1) \\ \mathbf{o}_i^2 = g_2(\mathbf{W}_2 \mathbf{o}_i^1 + \mathbf{b}_2) \\ \vdots \\ \mathbf{z}_i = \mathbf{o}_i^h = g_h(\mathbf{W}_h \mathbf{o}_i^{h-1} + \mathbf{b}_h) \end{cases} \tag{3}$$

where h is the number of layers that compose the network, and $\mathbf{W}_x$, $\mathbf{b}_x$, and g_x are weight matrix, bias vector, and activation function of the x-th layer, respectively. The expressive power of neural networks allows capturing nonlinear relationships among multiple features that are of great importance to represent high-level concepts. To

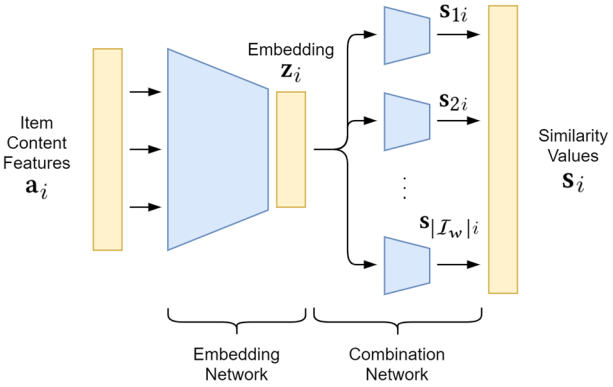

Fig. 1 Architecture of NFC. The model takes as input the feature vector of an item i and predicts the similarities between each warm item and i

better understand the relevance of feature combinations, consider, as an example, the movie domain. *Harrison Ford* and *Mark Hamill* are famous actors that appeared in several movies, often very different from each other. While the information that each of them acted in a movie does not provide many insights, the simultaneous presence of both is very informative. Indeed, they interpreted two main characters in the widely known *Star Wars* saga. It follows that if a model is able to merge the two pieces of information, it can recognize movies that belong to that saga, an important detail that single features could not depict individually.

3.2.2 Combination network

Given the embedded representation $\mathbf{z}_i$ of an item i, the combination network estimates the similarity values between all the warm items $j \in \mathcal{I}_w$ and item i. As shown in Fig. 1, it is composed of a small, independent neural network with h' fully connected hidden layers for every single warm target item, that can be defined as:

$$\begin{cases} \mathbf{y}_{ji}^1 = g'_{j1}(\mathbf{W'}_{j1}\mathbf{z}_i + \mathbf{b'}_{j1}) \\ \mathbf{y}_{ji}^2 = g'_{j2}(\mathbf{W'}_{j2}\mathbf{y}_{ji}^1 + \mathbf{b'}_{j2}) \\ \quad \vdots \\ \mathbf{s}_{ji} = \mathbf{y}_{ji}^{h'} = g'_{jh'}(\mathbf{W'}_{jh'}\mathbf{y}_{ji}^{h'-1} + \mathbf{b'}_{jh'}) \end{cases} \tag{4}$$

where $\mathbf{W'}_{jx}$, $\mathbf{b'}_{jx}$, and g'_{jx} are weight matrix, bias vector, and activation function of the x-th layer of the neural network related to target item j, respectively.

The aim of this structure is to exploit nonlinear combinations of the latent features that compose the embedding, taking into account multiple and different aspects, depending on the warm target item involved in the computation of the similarity. Indeed, different embeddings might lead to close similarity values, because different users look at different aspects of the same item, and each of them can be relevant in a specific scenario. As another example related to the movie domain that

can help in understanding this intuition, consider a film belonging to the aforementioned *Star Wars* saga. Clearly, it is similar to all the other movies belonging to the saga. However, it can also be considered similar to a wide variety of other movies for several different reasons: similar to all the other science fiction movies with Harrison Ford, because he has unique acting skills; similar to philosophical movies, because of the story it tells; similar to movies with cutting edge special effects of the same decade; and so on. In general, there are many relevant high-level concepts that could conduce to high values of similarity that nonlinear relations between features allow to represent.

3.2.3 Comparison with feature weighting

CFW-D and HP^3 are feature weighting techniques based on the intuition that learning from collaborative similarity matrices can be beneficial for the accuracy in the item cold-start recommendation task. NFC is based on the same insight, but, compared to feature weighting techniques, it is a widely more powerful model for two main reasons. The first is that it is able to capture nonlinearities and multiple combinations of the content features of the input item and to weigh them differently, according to the warm target item. The second is that it does not require that content and collaborative information *overlap*. As an example, consider two items that have very similar interaction profiles, but their content representations are completely different. In this scenario, a feature weighting approach cannot merge collaborative and content information, because while the collaborative similarity value between the items is high, the content similarity cannot be different from 0. NFC, instead, removes the overlap constraint, combining directly the two types of information.

3.2.4 Implementation details

In this section, we provide some details about the implementation of NFC. The final architecture of the model includes batch normalization (Ioffe and Szegedy 2015) after every hidden layer, which allows mitigating the issues introduced by very sparse content representations used as input. The batch size during training is set to 24. We also adopt the dropout technique (Srivastava et al. 2014), to reduce the risk of overfitting and increase the generalization capabilities of the model. We employ the widely known Rectified Linear Unit (ReLU) as activation function for all the hidden layers and a linear activation for the output layer, as this configuration provided the highest accuracy in our experiments.

We perform a sub-sampling of zero values used in the training process when the sparsity of the collaborative similarity matrix to learn is very high (i.e., in Nearest Neighbors algorithms). In these scenarios, the number of zeros in the similarity matrix largely outclasses the number of non-zero values, generating an imbalance problem that could negatively affect the training procedure. Formally, given a discrete random variable X with a Bernoulli distribution with success probability ρ, we approximate the loss function L in (2) as:

$$L \approx \sum_{(i,j)\notin\mathcal{Z}} (s_{ij}^{CF} - s_{ij})^2 + \sum_{(i,j)\in\mathcal{Z}} \mathrm{X}(s_{ij}^{CF} - s_{ij})^2 \tag{5}$$

where $\mathcal{Z} = \{(i,j) \in \mathcal{I}_w \times \mathcal{I}_w | s_{ij}^{CF} = 0\}$ is the set of the item couples for which the collaborative similarity is null. This corresponds to select only a fraction equal to ρ of the zero-valued similarities to use as target samples during the model training.

Finally, we apply a Nearest Neighbors technique to the similarity matrix generated by NFC, selecting, for each item, only the k most similar items, while all the other similarities are set to zero. This common technique allows to largely reduce the time and memory requirements of the recommendation process (Desrosiers and Karypis 2011).

Note that we release the repository with the code and the data used in the experiments.[1] It contains the implementation of all the algorithms, including NFC and all the other baselines. We include the code used for the generation of the datasets and the splits used for train, test, and validation. We provide the implementation of the hyper-parameter optimization, and the code used to perform all the experiments that we report in this paper.

4 Learning from ratings

The main goal of NFC is to generate a hybrid similarity matrix to provide more accurate recommendations in an item cold-start scenario. One of the main characteristics of the model is that it is trained using collaborative item similarity values as targets instead of user preferences directly. This technique has been recently proposed in some works that specifically treat the cold-start item recommendation problem (Cella et al. 2017; Dacrema et al. 2018; Bernardis et al. 2018). According to these works, training on similarity values brings a number of advantages:

- Similarity matrices are denser than user rating matrices, alleviating the sparsity problem that is known to afflict the field of recommender systems;
- The collaborative item-based model used to generate the similarity matrix to learn can be chosen properly, depending on its characteristics;
- The model requires a lower amount of time to converge.

One of the objectives of this paper is to assess the validity of these statements, in particular for the NFC model. For this reason, in this section we propose three different variants of NFC that learn from user feedback employing different optimization procedures, that will be thoroughly compared with NFC under different aspects in the experimental part of this work.

To let the model learn from user preferences, we simply modify the training procedure, while the architecture of the model remains unchanged. In Eq. (1), we defined how an item-based model, including NFC, predicts a preference for a

[1] https://github.com/cesarebernardis/NeuralFeatureCombiner

user-item couple. Given a user profile $\mathbf{r}_u$ and the i-th column of the similarity matrix $\mathbf{s}_i$, the prediction is computed with the dot product between them. Note that considering the structure of NFC, $\mathbf{s}_i$ is the output of the deep learning model when the feature vector $\mathbf{a}_i$ of item i is provided as input. It follows that the prediction for a user-item couple requires only the user preferences on warm items and the content features of the item.

4.1 NFC-MSE

NFC-MSE is a variant of NFC that adopts the mean squared error as pointwise loss, in order to reproduce the correct feedback value. Due to the high sparsity that characterizes recommender systems datasets, for the majority of user-item couples we have no information about the preference score that the user would give to the item. To solve this issue, one of the most common approaches, that we also adopt in this variant of our model, is to consider the absence of a preference as negative feedback, according to the *Missing-as-Negative* assumption. However, since the amount of missing feedback is way higher than the amount of the available ones, we subsample their number. Similarly to Sect. 4.2.4, in order to sample a fraction ω of all the negative feedback available, we define a discrete random variable X with a Bernoulli distribution with probability ω, and the set $\mathcal{N}$ of user-item couples with negative preference, i.e., $\mathcal{N} = \{(u,i) \in \mathcal{U} \times \mathcal{I} | r_{ui} = 0\}$.[2] Moreover, to prevent the model from learning the identity matrix as output similarity, during training we exclude the rating of an item when computing a preference that involves it, defining a new prediction rule: $\hat{r}_{ui} = \sum_j^{\mathcal{I}_w \setminus i} r_{uj} \cdot s_{ji}$. The new regularized loss function becomes:

$$L = \sum_{(u,i)}^{\mathcal{U} \times \mathcal{I} \setminus \mathcal{N}} (r_{ui} - \hat{r}_{ui})^2 + \sum_{(u,i)}^{\mathcal{N}} \mathrm{X}(r_{ui} - \hat{r}_{ui})^2 + \lambda ||\theta||^2 \tag{6}$$

Note that if $r_{ui} = 0$, i.e., if user u provided negative feedback for item i, then $\tilde{r}_{ui} = \hat{r}_{ui}$ and both notations can be used equivalently.

4.2 NFC-CE

The second variant of NFC that we propose interprets the preference prediction as a binary classification problem and minimizes the binary cross-entropy loss. In this scenario, we assume that all positive preferences have value 1, while negative preferences have value 0. We use the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ to transform the rating predictions provided by the model into probabilities. Similarly to NFC-MSE, we adopt the Missing-as-Negative assumption, and we subsample the negative feedback. The loss function is:

[2] ω is a hyper-parameter of the model, and it is optimized during the tuning procedure.

$$L = -\sum_{(u,i)}^{\mathcal{U}\times\mathcal{I}\setminus\mathcal{N}} \ln \sigma(\hat{r}_{ui}) - \sum_{(u,i)}^{\mathcal{N}} \mathrm{X} \ln (1 - \sigma(\hat{r}_{ui})) + \lambda||\theta||^2 \tag{7}$$

where $\mathcal{N}$ is the set of user-item couples with negative preference defined in Sect. 5.1.

4.3 NFC-BPR

This variant of NFC is trained using the Bayesian Personalized Ranking optimization criterion presented in Rendle et al. (2009). This approach uses item pairs as training data and learns to rank them correctly for a given user, instead of scoring them individually. Each pair of items is composed of an item for which the user provided positive feedback and an item for which the user provided negative feedback. Also in this case, according to the Missing-as-Negative assumption, we consider missing preferences as negative. We define the set of training triplets $\mathcal{D} = \{(u,i,j)|u \in \mathcal{U} \wedge i \in \mathcal{I}_u^+ \wedge j \in \mathcal{I}\setminus\mathcal{I}_u^+\}$, where $\mathcal{I}_u^+$ is the set of items for which a given user u provided a positive feedback. The new loss function to minimize is:

$$L = -\sum_{(u,i,j)}^{\mathcal{D}} \ln \sigma(\hat{r}_{ui} - \hat{r}_{uj}) + \lambda||\theta||^2 \tag{8}$$

where σ is the sigmoid function.

To select the negative item sample j in each triplet (u, i, j), we employ a multinomial probability distribution, where the probability of selecting an item is proportional to its popularity $c_j = |\{v \in \mathcal{U}|r_{vj} \neq 0\}|$, as proposed in several works in literature (Rendle and Freudenthaler 2014; Hidasi et al. 2016a, b). In particular, the probability $p(j|u)$ of selecting an item $j \in \mathcal{I}\setminus\mathcal{I}_u^+$ as negative sample for user u is computed as:

$$p(j|u) = \left(\frac{c_j}{\sum_k^{\mathcal{I}\setminus\mathcal{I}_u^+} c_k}\right)^{\alpha} \tag{9}$$

where the exponent $\alpha \in [0, 1]$ is a hyper-parameter, tuned during the hyper-parameter optimization procedure on the validation set, that attenuates the bias toward the choice of popular items as negative samples.

5 Experimental settings

In this section, we report the information concerning the experimental setup, including the datasets used, the collaborative algorithms employed to generate the similarity matrices, and the baselines that we compare NFC with.

5.1 Datasets

We used four well-known datasets that also provided non-user-based tags as item features:

Movielens Hetrec:[3] this dataset is an extension of MovieLens10M dataset and links the movies of MovieLens dataset with their corresponding web pages at Internet Movie Database (IMDb) and Rotten Tomatoes movie review systems. As item features, we kept only movie years, genres, actors, directors, countries, and locations. The preferences have been binarized considering all the ratings greater or equal to 3.5 as positive preferences.
BookCrossing (Ziegler et al. 2005): it is composed of user ratings over books, collected from the BookCrossing website. The dataset preferences have been binarized considering implicit ratings and explicit ratings greater or equal to 7 as positive feedbacks.
Yahoo! Movies:[4] it contains user preferences for various movies collected from the Yahoo! Movies community. It also includes descriptive information about movies like their cast, the crew, the genre, the awards, and much more. The preferences have been binarized as described for Movielens.
Amazon Video Games (Ni et al. 2019): it is a collection of reviews gathered from the Amazon website in a 22 years period between 1996 and 2018. Reviews contain ratings on a 5-star scale, and products are accompanied with detailed metadata. Also in this case the preferences have been binarized as described for Movielens.

In every dataset, we kept only users with at least 5 interactions, items with at least 5 interactions and 5 features, and features belonging to at least 5 items. These filters have been applied iteratively, in order to assert that all the constraints were satisfied after the processing. Note that these filters are commonly applied in literature (Elbadrawy and Karypis 2015; Sharma et al. 2015; Liang et al. 2018). Some statistics about the final datasets are summarized in Table 1.

Table 1 Details of datasets used

Dataset	Users (K)	Items (K)	Features (K)	Ratings (K)	Density (%)
Movielens Hetrec (MH)	2.1	6.2	8.2	511	3.92
BookCrossing (BX)	8.7	19.7	4.5	299	0.17
Yahoo! Movies (YM)	7.4	3.2	8.4	161	0.68
Amazon Video Games (AV)	33.9	12.5	13.0	295	0.07

[3] https://grouplens.org/datasets/hetrec-2011/.
[4] Yahoo! Webscope dataset, Movies User Ratings and Descriptive Content Information, v.1.0. http://research.yahoo.com/Academic_Relations.

5.2 Collaborative algorithms

To generate the collaborative similarity matrices used as learning targets in our new model, we employed four different item-based algorithms:

ItemKNN-CF: it is a Nearest-Neighbors item-based Collaborative Filtering approach. This algorithm defines the similarity between two items as the Cosine similarity between the vectors of the respective preference profiles. For each item, only the k most similar items and their respective values are considered, all the other similarities are set to 0 (Desrosiers and Karypis 2011).
RP^3_β: it is a graph-based approach where the similarity between two items i and j is defined as the probability to reach j from i after 2 steps random walks. This probability is also penalized by the popularity of the item to reach (Paudel et al. 2016).
SLIM: it is the well-known item-based approach called Sparse Linear Method. The sparse similarity matrix is obtained solving a linear regression problem, minimizing the root mean square error between all the predictions for each item and the corresponding actual rating profile (Ning and Karypis 2011).
EASE-R: it is a linear model called Embarrassingly Shallow AutoEncoder, recently proposed by Harald Steck. It can be considered an atypical autoencoder without hidden layers. Despite its simplicity, it achieves state-of-the-art performance in terms of top-N recommendation accuracy (Steck 2019).

We tuned all the algorithms to find the best configurations, evaluating on the warm split described in Sect. 7.2. In Table 2, we show the performance of each collaborative approach in the warm-start scenario. As expected, we can see that all collaborative algorithms largely outperform the content-based approach. In particular, SLIM proves to be the best performing algorithm on all the datasets. Note that we do not test these algorithms in a cold-start scenario as they would not provide personalized recommendations, since they have no collaborative information about cold items.

Table 2 Comparison of NDCG@10 of item-based algorithms in the warm-start scenario

Algorithm	AV	BX	MH	YM
ItemKNN-CF	0.1035	0.0849	0.1694	0.2941
RP^3_β	0.1027	0.0823	0.1738	0.2964
SLIM	**0.1051**	**0.0878**	**0.2010**	**0.3249**
EASE-R	0.1039	0.0873	0.1921	0.3200
ItemKNN-CBF	0.0298	0.0499	0.0202	0.0963

The best performing algorithm is in bold

5.3 Baselines

We compared our new model with both baselines and state-of-the-art models for cold items recommendation:

ItemKNN-CBF: it is an item content-based k Nearest-Neighbors approach based on cosine similarity (Desrosiers and Karypis 2011; Lops et al. 2011). We tested this algorithm also applying TF-IDF and BM25 weighting techniques to item features before calculating the content similarity matrix. The best performing weighting approach on the validation set is used. Despite their simplicity, recent works have shown that these algorithms still have competitive accuracy in the top-N recommendation task (Dacrema et al. 2019).
FBSM: it is a factorized model called Feature-based factorized Bilinear Similarity Model that learns a factorized similarity model, based on user preferences and item features (Sharma et al. 2015).
BPRMF-AFM: it is the linear-mapping version of BPR Matrix Factorization with Attribute-to-Feature Mapping, that uses the mapping concept to construct a feature-aware matrix factorization model (Gantner et al. 2010).
LCE: it is a matrix factorization technique called Local Collective Embeddings, that collectively decomposes the ICM and the URM in a common low-dimensional space (Saveski and Mantrach 2014).
DCT: it is a matrix factorization method called Decoupled Completion and Transduction, that decouples the completion of rating matrix from the transduction of knowledge from existing ratings to cold-start items (Barjasteh et al. 2015). The auxiliary items' similarity matrix is obtained through ItemKNN-CBF with the best performing configuration on each dataset, and no auxiliary users' similarity matrix is employed.
CFW-D: it is Collaborative Feature Weighting, a feature weighting method that learns the relevance of each content feature from a collaborative item similarity. Dacrema et al. (2018) propose two variants of this model, with and without latent factors. In this work, we adopt the variant without latent factors, called CFW-D, that the authors prove to be the best performing one.
HP^3: it is a graph-based feature weighting technique that, similarly to CFW-D, learns the importance of features from a collaborative item similarity (Bernardis et al. 2018).

We also include two models that have not been explicitly designed for the cold-start item recommendation scenario, but they have been successfully employed for this task in literature (Pan et al. 2019; Lian et al. 2017):

FM: it is a model class known as Factorization Machines that combines the advantages of Support Vector Machines (SVM) with factorization models for

high sparsity problems (Rendle 2010). We used the implementation provided in the xLearn library.[5]
WideAndDeep: it is a neural approach called that jointly trains wide linear models and deep neural networks to combine the benefits of memorization and generalization (Cheng et al. 2016). We used the implementation provided in the Tensorflow library.[6]

Note that for FM, WideAndDeep, and BPRMF-AFM we used one-hot encoding of ids for users' representations and content features as items' representations. The implementation of all the algorithms is available in the repository.

6 Cold-start recommendation

In this section, we report the results of the analyses that we performed in item cold-start scenarios.

6.1 Evaluation procedure

In order to reproduce a cold-start item scenario, we adopted the same strategy proposed in Sharma et al. (2015). We performed the split selecting three random subsets of items and all their respective interactions. We kept 60% of items for the training set, 20% for the validation set, and 20% for the test set. We trained the models on the training set, and we selected the hyper-parameter configuration that provided the best performance on the validation set. We obtained the final results reported in this paper training the models on the union of train and validation sets, using the best configuration found, and we evaluated them on the test set.[7] For a more thorough analysis, we repeated the evaluation on the test sets of 10 different splits, using the same optimal configuration. During the evaluation, we forced all the models to recommend only items belonging to the test (or validation) set used.

6.2 Hyper-parameter optimization

We optimized the hyper-parameters of all the algorithms maximizing the NDCG@10 on the validation set described in Sect. 7.1. We used the same validation also to select the number of training epochs for each algorithm, through an early stopping technique. In particular, we evaluated each model every 5 epochs and terminated the training after 2 consecutive evaluations in which the best result did not improve. To select the best hyper-parameters of collaborative algorithms, we further split the training set in order to obtain a warm holdout, selecting randomly the 80%

[5] https://xlearn-doc.readthedocs.io/en/latest/.
[6] https://www.tensorflow.org/.
[7] This is equivalent to an 80–20% split for train and test, respectively.

Table 3 Best NFC configurations and hyper-parameter values found on the validation set for each dataset

Hyper-parameter	Range	AV	BX	MH	YM
k-NN	[50, 500]	50	50	160	80
Learning rate	$[1e^{-6}, 1e^{-3}]$	$6.9e^{-6}$	$1.0e^{-5}$	$3.6e^{-4}$	$4.2e^{-5}$
L2 reg.	$[1e^{-6}, 1e^{-2}]$	$4.7e^{-6}$	$1.0e^{-6}$	$2.0e^{-4}$	$5.9e^{-4}$
Zeros quota (ρ)	$[1e^{-3}, 0.5]$	0.26	0.50	0.02	0.04
Dropout rate	[0.00, 0.80]	0.80	0.50	0.05	0.80
Epochs	[5, 200]	105	105	10	55
Collaborative Similarity	(EASE-R, ItemKNN-CF, RP^3_β, SLIM)	RP^3_β	RP^3_β	SLIM	ItemKNN-CF
Embedding network					
Number of layers	[2, 3]	2	2	2	3
Sizes of layers	[128, 1024]	(1024, 482)	(1024, 371)	(205, 488)	(1024, 823, 512)
Combination network					
Number of layers	[1][a]	1	1	1	1
Sizes of layers	[1, 32]	1	32	2	1

The combination network is assumed to be composed of networks with the same architecture for each warm item. We report the characteristics of one of them. Square brackets in explored ranges define an interval

[a]Multiple layers in the Combination Network were high memory demanding and did not improve the accuracy of the model

of the interactions for the warm training set and the remaining 20% for the warm validation set. As optimization algorithm, we employed the Bayesian Optimization technique implemented in the *scikit-optimize* Python library,[8] testing 50 configurations for each approach. In Table 3, we report the explored ranges for all the hyper-parameters and, for each dataset, the best configurations of NFC that we used in all the experiments. The optimization procedures of NFC and all the other baselines are available in the repository.

6.3 Comparing learning targets

In Sect. 5, we discussed the theoretical advantages of learning from similarities against learning from user-item interactions directly. We now want to assess the difference between the two techniques under different aspects, in order to prove empirically the validity of those claims.

In Tables 4 and 5, we show NDCG and Recall, respectively, in the item cold-start scenario at cutoffs 10 and 25 of NFC and all its variants that learn from ratings presented in Sect. 5. We report the average metric scores of the evaluations performed over 10 different splits and the p values of the Wilcoxon signed-rank tests between

[8] https://scikit-optimize.github.io/stable/index.html.

Table 4 Comparison of NDCG of the NFC variants in the cold-start scenario

Algorithm	AV		BX		MH		YM	
	@10	@25	@10	@25	@10	@25	@10	@25
NFC	**0.0953**	**0.1212**	**0.1207**	**0.1448**	**0.1212**	**0.1687**	**0.2680**	**0.3317**
NFC-MSE	0.0783*	0.1038*	0.0926*	0.1118*	0.0804*	0.1177*	0.1834*	0.2329*
NFC-CE	0.0461*	0.0611*	0.0602*	0.0735*	0.0715*	0.1146*	0.1724*	0.2114*
NFC-BPR	0.0173*	0.0233*	0.0175*	0.0209*	0.0623*	0.1033*	0.1882*	0.2501*

We report the average of the results obtained on 10 different folds. The best performing algorithm for each dataset and cutoff is highlighted in bold

*Indicates a p value of the significance test below 0.05

Table 5 Comparison of recall of the NFC variants in the cold-start scenario

Algorithm	AV		BX		MH		YM	
	@10	@25	@10	@25	@10	@25	@10	@25
NFC	**0.1486**	**0.2350**	**0.1448**	**0.2040**	**0.0754**	**0.1350**	**0.3479**	**0.5256**
NFC-MSE	0.1229*	0.2094*	0.1120*	0.1595*	0.0525*	0.1000*	0.2250*	0.3625*
NFC-CE	0.0727*	0.1238*	0.0754*	0.1113*	0.0494*	0.1063*	0.2219*	0.3299*
NFC-BPR	0.0254*	0.0457*	0.0178*	0.0261*	0.0416*	0.0950*	0.2419*	0.4221*

We report the average of the results obtained on 10 different folds. The best performing algorithm for each dataset and cutoff is highlighted in bold

*Indicates a p value of the significance test below 0.05

NFC and its variants that learn from ratings. The results show that NFC largely outperforms all the variants that learn from ratings by a wide margin on all the datasets, metrics, and cutoffs. Moreover, the p value of the significance test is below 0.05 in every configuration. NFC-MSE is the second-best performing algorithm in most scenarios, but its accuracy is about 20% lower than NFC. NFC-CE has a performance comparable with NFC-MSE on Movielens and Yahoo!, the two densest datasets, while the distance widens on the other, more sparse datasets. The same behavior is shown more evidently by NFC-BPR, which strongly struggles on Amazon and Bookcrossing, with accuracy about 5 times lower than NFC-MSE.

Another important aspect that we want to analyze, is the training time required by the different techniques. In Table 6, we report the average time in seconds taken by the algorithms to complete the training procedure over the 10 different splits that we used for the accuracy evaluation. Note that the values reported for NFC include the time required by the collaborative algorithm to train and generate the similarity matrix used as learning target. We trained all the models on an NVIDIA RTX 2080Ti. Even in this case, NFC outperforms all the variants that learn from ratings in almost every scenario. It is twice faster than the second-best approach on 3 datasets over 4, demonstrating that learning from similarities leads to a quicker

Table 6 Comparison of training time in seconds in the cold-start scenario

Algorithm	AV	BX	MH	YM
NFC	**833**	**2805**	**203**	**49**
NFC-MSE	8080	10,085	2674	1257
NFC-CE	1987	6394	408	129
NFC-BPR	1629	6829	1647	**47**

We report the average of the results obtained on 10 different folds. Lowest training times for each dataset are in bold

convergence and allows training the model in a much shorter amount of time. NFC-MSE, even if it is usually more accurate than the other NFC variants, is the slowest on every dataset. NFC-CE and NFC-BPR have similar training times on Amazon and BookCrossing. The latter, in particular, has a convergence time on par with NFC on the Yahoo! dataset.

The results obtained in these experiments empirically confirm the advantages of learning from similarities. This technique allows reaching a largely higher recommendation accuracy in the cold-start scenario, requiring a widely smaller amount of time for the model training.

6.4 Comparison with baselines

In Tables 7 and 8, we summarize the accuracy of Top-N recommendations for all the algorithms in the cold-start item scenario, expressed, respectively, as NDCG and Recall at cutoffs 10 and 25. We report the average metric scores obtained on 10

Table 7 Comparison of NDCG in the cold-start scenario

Algorithm	AV		BX		MH		YM	
	@10	@25	@10	@25	@10	@25	@10	@25
ItemKNN-CBF	0.0782*	0.0979*	0.1135*	0.1355*	0.1043*	0.1413*	0.2160**	0.2616*
FM	0.0080*	0.0129*	0.0106*	0.0175*	0.0772*	0.1175*	0.1723*	0.2231*
WideAndDeep	0.0057*	0.0085*	0.0207*	0.0310*	0.0787*	0.1088*	0.1793*	0.2358*
FBSM	0.0630*	0.0884*	0.0819*	0.1007*	0.0488*	0.0770*	0.2269	0.2962
BPRMF-AFM	0.0611*	0.0836*	0.0914*	0.1133*	0.0828*	0.1292*	0.1819*	0.2437*
LCE	0.0407*	0.0560*	0.0665*	0.0833*	0.0880*	0.1202*	0.1256*	0.1942*
DCT	0.0697*	0.0885*	0.0941*	0.1152*	0.1133**	0.1593**	0.1854*	0.2237*
HP^3	0.0713*	0.0896*	0.1112*	0.1322*	0.1102**	0.1533*	0.2230	0.2830
CFW-D	0.0589*	0.0732*	0.1074*	0.1295*	0.0850*	0.1240*	0.2405	0.3041
NFC	**0.0953**	**0.1212**	**0.1219**	**0.1461**	**0.1212**	**0.1687**	**0.2680**	**0.3317**

We report the average of the results obtained on 10 different folds. The best performing algorithm for each dataset and cutoff is in bold

*Indicates a p value of the significance test below 0.05

**Indicates a p value below 0.10

Table 8 Comparison of Recall in the cold-start scenario

Algorithm	AV		BX		MH		YM	
	@10	@25	@10	@25	@10	@25	@10	@25
ItemKNN-CBF	0.1181*	0.1842*	0.1378*	0.1910*	0.0650*	0.1115*	0.2455*	0.3710*
FM	0.0134*	0.0302*	0.0142*	0.0317*	0.0489*	0.1017*	0.2531*	0.4024*
WideAndDeep	0.0100*	0.0191*	0.0317*	0.0584*	0.0485*	0.0866*	0.2574*	0.4194*
FBSM	0.1039*	0.1904*	0.0988*	0.1462*	0.0314*	0.0666*	0.2997	0.5016**
BPRMF-AFM	0.0984*	0.1746*	0.1165*	0.1726*	0.0567*	0.1155*	0.2610*	0.4416*
LCE	0.0672*	0.1187*	0.0847*	0.1282*	0.0556*	0.0963*	0.1772*	0.3763*
DCT	0.1084*	0.1709*	0.1143*	0.1661*	0.0713**	0.1281*	0.2082*	0.3118*
HP^3	0.1062*	0.1677*	0.1319*	0.1824*	0.0722	0.1265	0.2703**	0.4444*
CFW-D	0.0876*	0.1358*	0.1294*	0.1849*	0.0549*	0.1047*	0.3122	0.4917*
NFC	**0.1486**	**0.2350**	**0.1455**	**0.2049**	**0.0754**	**0.1350**	**0.3479**	**0.5256**

We report the average of the results obtained on 10 different folds. The best performing algorithm for each dataset and cutoff is in bold

*Indicates a p value of the significance test below 0.05

**Indicates a p value below 0.10

different splits and the p values of the Wilcoxon signed-rank tests between NFC and all the other approaches.

The first evident result is that NFC largely outperforms all the other techniques on all the datasets, metrics, and cutoffs. Its accuracy is between 10 and 20% higher than the second-best performing algorithm. However, it is interesting to notice that on Yahoo!, despite the quite large difference in the mean value of the metric, due to the high variance of the NDCG scores and the low number of samples, the p value of the significance test is above 0.1 in 6 comparisons out of 18. This effect is largely mitigated when looking at the Recall on the same dataset, since the p value is above 0.1 in only 2 cases out of 18. Another interesting result is represented by the performance of ItemKNN-CBF, the purely content-based approach. It is the second most accurate approach on the two most sparse datasets, namely Amazon and BookCrossing, and it provides competitive results also on Movielens and Yahoo!. The other hybrid baselines, instead, are particularly influenced by the characteristics of the dataset, and it is hard to delineate a ranking among them. FBSM obtains good Recall scores on Amazon and Yahoo! at high cutoffs, but it does not perform competitively on Movielens. LCE is on par with most techniques on Movielens, but its accuracy is generally lower on the other datasets. BPRMF-AFM and DCT obtain consistent results across the datasets, but their accuracy is never the best among the hybrid baselines. CFW-D and HP^3 are always among the best performing hybrid baselines. FM and WideAndDeep have very similar results in every scenario. In particular, they obtain quite good results on the densest datasets, Movielens and Yahoo!, while they hardly fail on the most sparse ones, where their accuracy is several times lower than the other hybrid alternatives.

Despite NFC is based on the same intuition of CFW-D and HP^3, it brings to an important improvement in terms of accuracy. This difference provides empirical

evidence of the advantages of the new NFC model over simple feature weighting techniques discussed in Sect. 4.2.3. Nevertheless, the competitive results obtained by CFW-D and HP^3 confirm that learning from collaborative similarity matrices is an effective strategy for the item cold-start recommendation task.

6.5 Extreme cold-start scenario

In this section, we want to perform a sensitivity analysis to study the accuracy of NFC in scenarios of extreme item cold-start. This experiment simulates the scarcity of collaborative information that can be extracted from the available data, reducing the number of interactions available in the train set. The results of these experiments can also be used to assess the performance of the algorithms in a ramp-up scenario, i.e., when the system is new and the quantity of feedback available is initially very low, but it increases over time since users continuously interact with the system. We adopted the same evaluation procedure and split strategy described in Sect. 7.1, gradually increasing the percentage of the items selected for the test set from 20 (i.e., the scenario described in Sect. 7.1) to 80%. We performed the experiments on the two datasets with the highest number of items, namely Amazon and BookCrossing, comparing NFC with a selection of the best performing algorithms, on average, in the cold-start scenario, according to the results in Table 7. In particular, we selected ItemKNN-CBF, HP^3, DCT and FBSM. For each approach, we adopted the best performing hyper-parameter configuration found as described in Sect. 7.2.

In Fig. 2, we show average and confidence interval at 95% of the accuracy performance obtained on 10 different folds, expressed as the NDCG@10. Overall, we observe a common decreasing trend in the accuracy of all the algorithms on both datasets, which means that they all benefit from a higher amount of information about user profiles. At low percentages of items in the test set, NFC is clearly the best performing algorithm, confirming the results reported in Sect. 7.4. However, when the size of the test set reaches 60% or more, the absence of collaborative information negatively influences the performance of NFC. Indeed, in these scenarios,

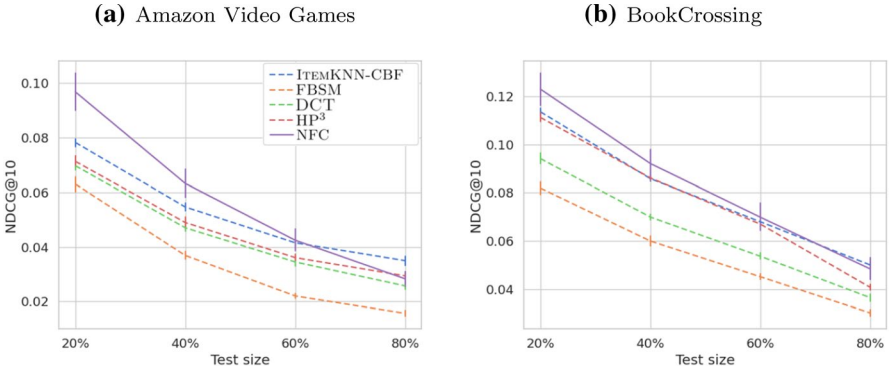

Fig. 2 Comparison of NDCG@10 of the best techniques in an extreme cold-start scenario, at different sizes of the test set. We report the average and confidence interval at 95% of 10 evaluation

it has comparable or worse accuracy than the purely content-based ItemKNN-CBF, but still in line or better than the hybrid competitors. HP^3 and ItemKNN-CBF have very similar trends on both datasets, with the second having a slightly better performance in most of the experimental configurations. DCT has a performance comparable to HP^3 on Amazon, while on BookCrossing it has a very similar trend with respect to FBSM, but with higher accuracy. FBSM, instead, is clearly the worse algorithm in the comparison.

Interesting to notice that all the baselines have analogous behaviors on both the datasets, while NFC is more impacted by the decreasing amount of interactions in the training set. This suggests that NFC is able to exploit more efficiently the collaborative information with respect to the competitors, but it is more negatively influenced by the scarcity of user feedback.

7 Hybrid scenario

In a real environment, new items, that have no interactions in the system and are consequently considered as cold items, are introduced into the catalog alongside the existing ones, for which user feedbacks are available (i.e., warm items). In order to achieve the highest accuracy, the algorithms have to find a good balance between the exploitation of the content information available for cold items and the collaborative information that they have for warm items. In this set of experiments, we wish to analyze this scenario at different cold-warm balancings, gradually increasing the weight of the cold items from 0% (all the interactions in the test set belong to warm items) to 75 (75% of the interactions are performed on cold items, the other 25% are performed on warm items).[9] Note that in this case also collaborative algorithms can provide personalized recommendations for warm items.

We compare NFC with the techniques that showed the best performance in the cold-start scenario, reported in Sect. 7.4. In particular, we select ItemKNN-CBF, FBSM, DCT, and HP^3. We also include SLIM as representative of collaborative approaches in the comparison, since it is the best performing collaborative technique in a warm-start scenario, as shown in Sect. 6.2. We perform the experiments on the two datasets used for the sensitivity analysis in Sect. 7.5, namely Amazon Video Games and BookCrossing.

7.1 Evaluation procedure

To form the test set, we first defined the fraction $\gamma \in \{0\%, 25\%, 50\%, 75\%\}$ of the test interactions that belong to cold items. Then we performed the same type of cold split described in Sect. 7.1, selecting a number of items equal to $20\% \cdot \gamma \cdot |\mathcal{I}|$ and all the respective interactions for the test set.[10] Finally, we selected the remaining

[9] We skip the case related to 100% since it is the cold-start scenario studied in Sect. 7.

[10] Note that this cold portion accounts for about $20\% \cdot \gamma$ of all the interactions in the dataset.

$20\% \cdot (100\% - \gamma)$ of the interactions uniformly at random from the items that were not selected by the cold split. Consequently, the test set contains 20% of all the interactions in the dataset, while the other 80% composes the training set. Note that $\gamma = 0\%$ corresponds to a common warm holdout, while $\gamma = 100\%$ is equivalent to the cold-start holdout presented in Sect. 7.1.

For all the recommenders, we used the best hyper-parameter configurations found during the optimization procedure described in Sect. 7.2. During the evaluation, we forced the algorithms to recommend the items that had at least one interaction in the test set. Note that, since also warm items have interactions in the test set, the number of items that an algorithm can recommend is way higher compared to the cold-start scenario described in Sect. 7.1. This difference makes the metrics scores of the two experiments directly incomparable.

7.2 Results

In Fig. 3, we show average and confidence interval at 95% of the accuracy performance obtained on 10 different folds, expressed as the NDCG@10.

NFC shows to be a strong competitor in every situation, especially if compared to all the other cold-start techniques, as it is by far the best performing hybrid algorithm at all cold percentages. At $\gamma = 75\%$, it achieves the best performance overall on both datasets. At lower values of γ, it reaches the highest accuracy on BookCrossing, while it fills the gap between SLIM and all the other baselines on Amazon. DCT, HP^3, and ItemKNN-CBF have very similar and stable trends, independently from γ. FBSM benefits from the presence of interactions and reaches higher accuracy in warm scenarios on Amazon, but it is the worse performing algorithm in most situations. Interesting to notice that the difference between NFC and all the other hybrid baselines is higher at low values of γ (i.e., a low amount of test interactions related to cold items). This confirms that our new NFC model is able to exploit the available collaborative information more efficiently than its competitors.

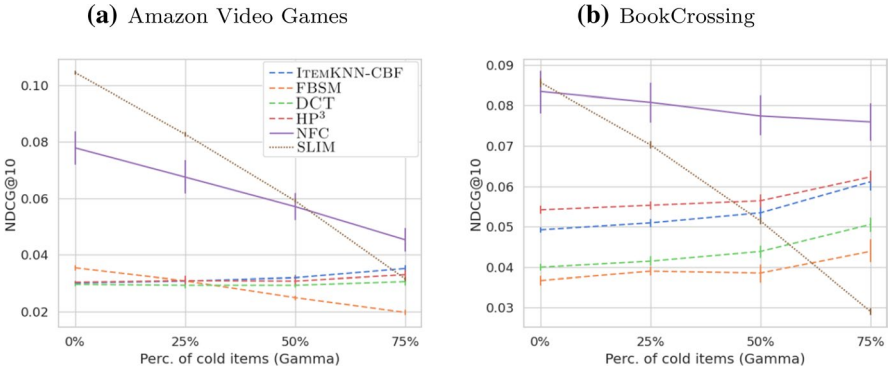

Fig. 3 Comparison of NDCG@10 of the best techniques in a hybrid scenario, at different percentages of interactions related to cold items. We report the average and confidence interval at 95% of 10 evaluations. We included SLIM as representative of the best pure collaborative algorithms across all the datasets

Note that at values of γ equal to 0%, 25%, and 50%, SLIM proves to be the most accurate algorithm on Amazon. This is an expected result, as collaborative approaches, and SLIM in particular, are known to perform very well in warm scenarios (Dacrema et al. 2019), while the other approaches we compare it with are specifically designed for cold-start recommendation. Moreover, recommending warm items is an easier task, since the recommenders have valuable collaborative information about them. However, with the increasing percentage of interactions related to cold-start items, the gap between SLIM and the other models decreases. Indeed, at $\gamma = 75\%$, SLIM performs on par or worse than the other algorithms on both datasets.

8 Illuminating the black box

Neural networks are often considered as *black boxes*, but interpreting their output and understanding what they learn are key factors for assessing the quality of a model. Indeed, these tasks allow ensuring that the model is providing a proper representation of a problem, and exclude undesired outcomes as the exploitation of artifacts in the data, biases, overfitting, and much more. For this reason, examining the *black box* has attracted attention in many different fields of application (Montavon et al. 2018; Gevrey et al. 2003; Guidotti et al. 2018). The aim of this section is to provide a qualitative analysis of the NFC model, by using two different approaches. In the first approach, we study the behavior of NFC with respect to other techniques by studying the neighborhood of episodes from the same series. In the second approach, we examine the importance that the network gives to the different input features, for a better understanding of how the model is exploiting the input data.

8.1 Neighborhood analysis

In this section, we qualitatively analyze the neighborhoods in the embedding space generated by our new NFC model, in order to gain explainable insights on why it provides more accurate recommendations. We perform the experiments on the Yahoo! movies dataset, since it has a good amount of quality editorial features and, due to the popularity of the movie domain, the results are easier to interpret and explain. We compare the hybrid embeddings of our new NFC model and two models that provide embedded representations for items, LCE, and BPRMF-AFM. In the comparison, we also include a hybrid and a content-based representation in the original features space with features weights applied, formerly CFW-D and ItemKNN-CBF.

This experiment is performed simulating a cold-start scenario. We replicate the cold split described in Sect. 7.1, keeping the 80% of the items for the training of the models and the other 20% for the test. We train the algorithms using the best hyper-parameter configurations found for the cold-start evaluation experiments in Sect. 7.2. We analyze the 5 nearest neighbors in the test set, according to different

models, of two popular movies: *Indiana Jones and the Temple of Doom* and *The World is Not Enough*. They are movies belonging to two well-known sagas, *Indiana Jones,* and *James Bond,* respectively. We ensure that the movie subject of the analysis, in turn, is in the training set, while all the other movies belonging to the same saga are included in the test set. Similarities between movies are computed with the cosine of the embeddings or the cosine of the weighted features, depending on the type of model. We expect to find films belonging to the corresponding sagas among the closest neighbors. However, note that this is not a trivial task because, due to the experimental setting we adopt, there is no information about the saga in the training set.

In the left column of Table 9, we show the 5 neighbors of *Indiana Jones and the Temple of Doom*. For each algorithm, neighbors are ordered from the closest (top) to the farthest (bottom). This movie is the second episode of the Indiana Jones trilogy, a very well-known character created by George Lucas, interpreted by Harrison Ford, and directed by Steven Spielberg. Since it is part of a trilogy, we expect the other two movies (namely *Raiders of the Lost Ark* and *Indiana Jones and the Last*

Table 9 Five nearest neighbors in a cold-start item scenario of the *Indiana Jones and the Temple of Doom* (on the left) and *The World Is Not Enough* (on the right) movies in the embedding or weighted feature spaces for the different algorithms

Algorithm	Nearest Neighbors	
	Indiana Jones and the Temple of Doom	The World is Not Enough
ItemKNN-CBF	Indiana Jones and the Last Crusade	Goldeneye
	Raiders of the Lost Ark	License to Kill
	Homeward Bound: The Incredible Journey	For Your Eyes Only
	Star Trek III: The Search For Spock	A View to a Kill
	The Color Purple	The Spy Who Loved Me
CFW-D	Indiana Jones and the Last Crusade	Goldeneye
	Raiders of the Lost Ark	License to Kill
	Sabrina	Octopussy
	Minority Report	A View to a Kill
	Empire of the Sun	Dr. No
LCE	The Pagemaster	White Squall
	Raiders of the Lost Ark	American Movie
	101 Dalmatians	Ronin
	Star Wars: Episode I	Shriek If You Know .
	Indiana Jones and the Last Crusade	Black and White
BPRMF-AFM	Indiana Jones and the Last Crusade	Jackie Chan's First Strike
	X-Men	The Legend of Drunken Master
	Bad Company	No Man's Land
	Frailty	A View to a Kill
	John Carpenter's Ghosts of Mars	Octopussy
NFC	Indiana Jones and the Last Crusade	Goldeneye
	Raiders of the Lost Ark	Dr. No
	Back to the Future Part III	A View to a Kill
	Ransom	Octopussy
	Jurassic Park III	License to Kill

For each algorithm, neighbors are ordered from the closest (top) to the farthest (bottom)

Crusade) to be among the nearest neighbors. Indeed, all the neighborhoods of the different models, with the exception of BPRMF-AFM, which is not able to recognize the first as one of the nearest neighbors, include both the additional movies of the saga. LCE and BPRMF-AFM struggle more to recognize the saga, since they are not able to identify the other two movies of Indiana Jones as the closest neighbors. Moreover, their neighborhoods are hardly interpretable, since they contain movies aimed to other types of audience. The neighbors generated by BPRMF-AFM are far in time from the Indiana Jones saga, since they all have been released in the new millennium, while Indiana Jones was released in 1984. Moreover, even though they are mainly action movies, they are less popular and quite different from the adventure genre of Indiana Jones, as they are closer to the thriller and horror genres. The same holds also looking at the neighbors generated by LCE. *The Pagemaster* is a live-action/animated fantasy adventure film, and *101 Dalmatians* is a live action adaptation of the homonymous animated movie. Both of them are family-friendly movies that are also suitable for children, and they are different from the movies of the Indiana Jones saga.

Concerning the two feature weighting models, CFW-D and ItemKNN-CBF, they both recognize the movies belonging to the saga as the most similar ones, but they struggle to fill the rest of the neighborhood with suitable movies for the same audience. In particular, they mainly focus on one or a small set of features that they consider highly relevant, but they include in the neighborhood movies that hardly fit with the saga. ItemKNN-CBF includes the coming-of-age period drama film *The Color Purple*, which has Steven Spielberg as director like *Indiana Jones and the Temple of Doom*, but it is a completely different type of movie. *Homeward Bound - The Incredible Journey* has the same adventure genre of Indiana Jones, according to the metadata of the Yahoo! dataset, but it is a children's movie. Looking at the neighbors generated by CFW-D, we can notice a similar behavior. *Sabrina* is a comedy with Harrison Ford (the interpreter of the Indiana Jones character) as protagonist, *Minority Report* and *Empire of the Sun* are, respectively, a thriller and a war drama both directed by Steven Spielberg. As highlighted, all these movies have relevant features in common with the considered Indiana Jones movie, but they are very different from it. The neighborhood of NFC instead, beside the two movies of the saga, includes action and adventure movies aimed at the same target audience. In particular, it inserts two movies belonging to, respectively, the Jurassic Park and the Back to the Future sagas. These movies are part of two of the most successful and popular sagas of the late '80s, like the Indiana Jones one. In this scenario, collaborative information plays a fundamental role, allowing NFC to map these movies in the same neighborhood in the embedding space learned.

In the right column of Table 9, we show the nearest neighbors of the movie *The World Is Not Enough*, one of the movies belonging to the James Bond saga, also known as *agent 007*. Also in this case, for each algorithm, neighbors are ordered from the closest (top) to the farthest (bottom). The recognition of the James Bond movies in the cold-start scenario of this experiment is a quite challenging task for two main reasons. First, the James Bond saga counts many movies in the Yahoo! dataset (14), covers dozens of years, and the casts and the crews of the movies are often very different from each other. Second, the models have no information about

the saga in the training set, as previously mentioned. Despite these difficulties, the two feature weighting methods, ItemKNN-CBF and CFW-D, are able to put the James Bond saga movies all close together. The same holds also for NFC that inserts only James Bond movies in the neighborhood. Interesting to notice that the two hybrid models that exploit collaborative information from similarities, CFW-D and NFC, identify the same 5 closest neighbors. The other approaches, namely LCE and BPRMF-AFM, strongly struggle to recognize the movies of the saga as the closest neighbors. LCE, especially, is not able to include even a single James Bond movie among the 5 nearest neighbors. BPRMF-AFM, instead, inserts two 007 movies as the fourth and fifth closest neighbors, respectively. Note that these two movies are also among the nearest neighbors identified by the other hybrid approaches, CFW-D and NFC, indicating that collaborative information likely influenced similarly the training of these models.

8.2 Feature weights

In this section, we want to compare the importance, or weight, that the different algorithms assign to the content features. Our goal is twofold: assess the difference between content-based and hybrid models, and analyze what distinguishes NFC from the other approaches. Moreover, we want to test the robustness of the different techniques, repeating the experiments without some crucial information present in the original dataset. Indeed, the unexpectedly good results obtained by the content-based algorithms in Sect. 9.1, can be explained by exploring the features available in the Yahoo! dataset. One of the richest fields available as content information is represented by the *synopsis*. This field contains some hand-crafted, meaningful keywords that describe each movie and, among them, there are very explicit ones, that allow even content-based approaches to recognize very different movies that belong to the same sagas. For this reason, to test the robustness we replicated the experiments also removing the synopsis features from the content information.

We compare NFC with the best performing algorithms in the qualitative experiment presented in Sect. 9.1, namely ItemKNN-CBF[11] and CFW-D. These last two approaches exploit feature weighting techniques, and assessing the feature importance is an integral part of the model. Shedding light on the black box represented by deep neural networks, instead, is not an easy task, but it has increasingly drawn the attention of researchers in the recent past, due to the importance that these techniques have across so many tasks and fields of application. In our experiments, we adopt DeepLIFT, proposed by Shrikumar et al. (2017), in order to assess the importance that the NFC model assigns to each feature when a specific input sample is provided. Following the categorization of the black box explanation problems proposed by Guidotti et al. (2018),

[11] The best configuration of ItemKNN-CBF on Yahoo! includes the BM25 feature weighting technique.

DeepLIFT is considered to solve the *outcome explanation problem*, which consists of providing an explanation for the outcome of the model against a given input instance. This approach backpropagates the contributions of all neurons in the network to every feature of the input, assigning scores according to the difference between the activation of each neuron and its *reference activation*. In our experiments, we used a zero-valued array as reference. We employed the implementation of DeepLIFT proposed by Ancona et al. (2018) in the DeepExplain python framework.[12]

In Fig. 4, we show the weights that the different techniques assign to the content features. We focus on the James Bond saga because it is the one that counts the highest number of movies (14) with a high diversity in terms of features among the movies that compose it, as mentioned in Sect. 9.1. For each recommender, we selected the 10 features with the highest average weight on all the movies that belong to the saga. In Fig. 4a, we show the weights of the models trained on the dataset including synopsis features, while in Fig. 4b these features are excluded. For a clearer representation, the average weights have been max-normalized.

Figure 4a shows that explicit synopsis features like *007* and *bond* have a largely higher importance compared to all the others. The importance that ItemKNN-CBF and CFW-D give to the two aforementioned features is so high to make the presence of all the other features almost irrelevant, with weights five or more times lower than the highest ones. NFC, instead, is able to take into account also less specific features, assigning them higher weights. Moreover, NFC is able to consider the genre *action/adventure* as a very important feature, which is a quite difficult task, as it appears in 13 movies of the saga, but also in other 455 movies. Clearly, ItemKNN-CBF and CFW-D are completely focused on the features that uniquely identify the James Bond saga, but they end up with a very narrow perspective. Valuing more generic features, like the genre, allows recognizing similar movies that do not necessarily belong to the same saga and help to provide more diverse and explainable recommendations. Thanks to the expressive power of neural networks, NFC is able to effectively combine specific and generic features and to assess their importance more carefully, according to the collaborative information exploited. Consequently, we also expect our approach to smooth the *over-specialization* problem (Lops et al. 2011). Thanks to the collaborative information enclosed in collaborative similarities, NFC can pay attention only to those features that are relevant to the users, reducing the risk of overweighting less popular features, as it happens when using information-based normalization schemes (such as TF-IDF or BM25).

The same result is reflected also in Fig. 4b, where the explicit and impactful synopsis features are missing. All the algorithms assess *desmond llewelyn*[13] and *albert broccoli*[14] as the two most important features. Looking at all the non-synopsis

[12] https://github.com/marcoancona/DeepExplain.

[13] Desmond Llewelyn is the actor that interpreted *Q*, the head of the research and development division of MI6 in James Bond movies between 1963 and 1999.

[14] Albert Broccoli is the producer of most of the James Bond movies until 1995.

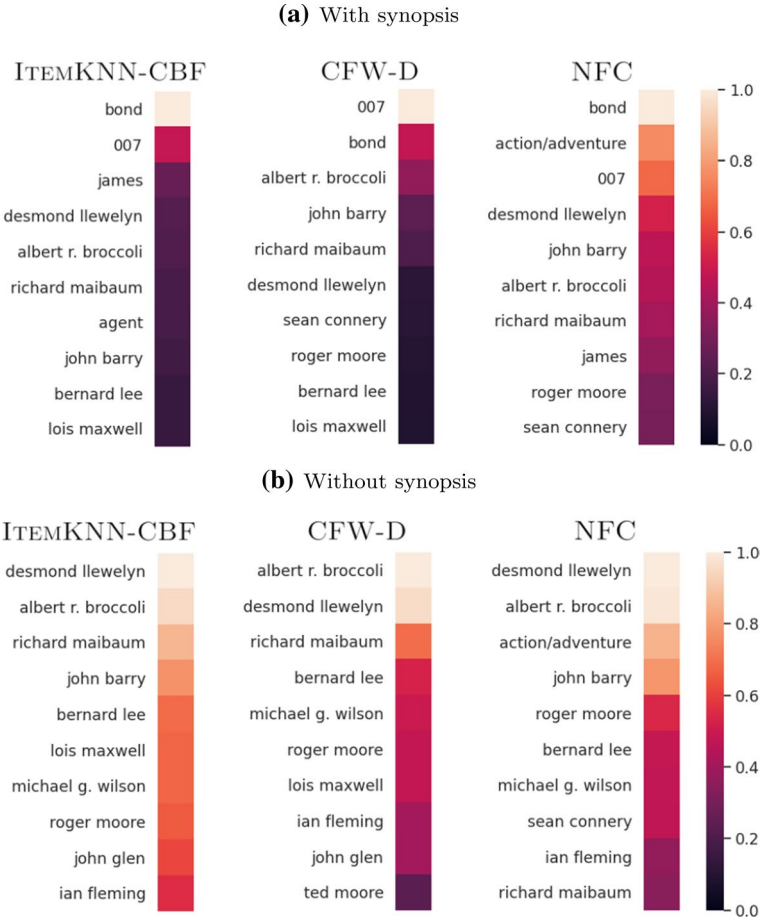

Fig. 4 Comparison of the average importance assigned by three recommenders to the features of the movies belonging to the James Bond saga. The experiment is performed with and without the synopsis features

features of James Bond movies, these two are the most characterizing ones, since they both appear in 9 movies of the saga and in almost no other movie in the dataset. However, users most likely do not watch James Bond movies due to their contribution. On the contrary, Roger Moore and Sean Connery are very famous actors that interpreted the James Bond character in a number of movies. Due to their popularity, it is not uncommon that people watch films where they appear mainly because of their acting skills. However, ItemKNN-CBF and CFW-D can only recognize Roger Moore as a relevant feature, while Sean Connery is not among the 10 most important ones. This happens because Roger Moore appears in only 5 movies that do not belong to the James Bond saga, while Sean Connery acts in 34, excluding the saga's ones. Even in this case, ItemKNN-CBF and CFW-D are strongly biased

toward characterizing singular features, while NFC confirms its ability to exploit also important, but more popular features.

9 Limitations and future directions

In the previous sections, we have shown the potential of our new NFC model, demonstrating that it represents an interesting advancement for cold-start item recommendation. However, this work also presents some limitations that can become the subject of future studies.

The first limitation of our study is represented by the offline setting of the experiments performed. It is widely known in literature that offline evaluation is often affected by several biases (Shani and Gunawardana 2011; Bellogín et al. 2017). Moreover, the offline qualitative study presented in Sect. 9 cannot be reliably generalized to every scenario. An online evaluation of the accuracy and the quality of the recommendations generated by NFC would confirm or deny its perceived effectiveness.

The second limitation of our study is the absence of an analysis about the consequences of using different collaborative similarities. The NFC model learns to reproduce a given collaborative item similarity matrix. In this work, the algorithm used to generate the similarities has been treated as a hyper-parameter of NFC. However, the qualitative and quantitative differences produced by the usage of different collaborative algorithms in the generation of the similarity matrix to learn have not been thoroughly investigated. This type of analysis represents an interesting direction for future works.

A third limitation of this work is represented by the absence of a study on the impact of item popularity or, in other words, the amount of collaborative information available for an item. Collaborative algorithms usually benefit from rich collaborative information, resulting in higher accuracy of the recommendations generated. It could be valuable to assess whether this statement is true also for our NFC model.

Finally, in the implementation proposed in this paper, NFC adopts a unique and collective concept of similarity, which is the most common approach in item-based similarity models. As a future direction, it could be beneficial to enhance NFC in order to take into account multiple user-specific views of the similarities exploiting individual user preferences.

10 Conclusions

In this paper, we proposed NFC, a novel deep learning model for cold-start item recommendation. The model learns to map the content representation of an item into a hybrid embedding space and to combine the features that compose the embedding in order to reproduce collaborative similarity values.

We presented three additional variants of NFC that learn from user ratings, using different optimization procedures. With extensive experiments on four datasets, we

showed that learning from similarity values has several advantages over these alternatives, including higher accuracy and lower convergence times.

We compared NFC with a number of state-of-the-art approaches for item cold-start recommendation. The results proved that the new model outperforms the accuracy of all its competitors by a large margin. The experiments in extremely cold scenarios demonstrated that NFC effectively exploits collaborative information, but struggles more than other techniques when the amount of ratings available during training is very low. The tests in a hybrid scenario confirmed that NFC outperforms its counterparts in cold scenarios, while it fills the gap between cold-start recommendation techniques and the best collaborative approaches in warm ones.

Finally, we conducted a qualitative analysis of the embeddings generated by NFC. We showed its capability to recognize similar movies in the embedding space learned, comparing it with other state-of-the-art techniques. Moreover, we assessed the relevance of collaborative information with a feature importance analysis, showing the ability of NFC to combine specific and more generic, but significant, features together.

Besides the interesting and promising results obtained, our work also presents some limitations. The generality of the results obtained in our experiments is bounded by the offline setting. The consequences of using different collaborative similarity matrices and the impact of item popularity on the recommendation quality and accuracy have not been thoroughly investigated. Future works should aim to address these limitations.

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement. Not applicable.

Availability of data and material Three out of four datasets used in the experiments are publicly available online. The last (Yahoo! Movies) is provided for the review process.

Code availability We provide the source code used to perform all the experiments, including data splitting, hyper-parameter optimization, and top-n recommendation evaluation. The link is the following: https://github.com/cesarebernardis/NeuralFeatureCombiner.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Ancona, M., Ceolini, E., Öztireli, C., Gross, M.: Towards better understanding of gradient-based attribution methods for deep neural networks. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30–May 3, 2018, Conference Track Proceedings, OpenReview.net (2018). https://openreview.net/forum?id=Sy21R9JAW

Barjasteh, I., Forsati, R., Masrour, F., Esfahanian, A.H., Radha, H.: Cold-start item and user recommendation with decoupled completion and transduction. In: Proceedings of the 9th ACM Conference on Recommender Systems, ACM, New York, NY, USA, RecSys '15, pp. 91–98 (2015). https://doi.org/10.1145/2792838.2800196

Barkan, O., Koenigstein, N., Yogev, E., Katz, O.: CB2CF: a neural multiview content-to-collaborative filtering model for completely cold item recommendations. In: Bogers T., Said A., Brusilovsky P., Tikk D. (eds.) Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16–20, 2019, ACM, pp 228–236 (2019). https://doi.org/10.1145/3298689.3347038

Bellogín, A., Castells, P., Cantador, I.: Statistical biases in information retrieval metrics for recommender systems. Inf. Retr. J. **20**(6), 606–634 (2017). https://doi.org/10.1007/s10791-017-9312-z

Bernardis, C., Dacrema, M.F., Cremonesi, P.: A novel graph-based model for hybrid recommendations in cold-start scenarios. CoRR (2018). arXiv:abs/1808.10664

Cella, L., Cereda, S., Quadrana, M., Cremonesi, P.: Deriving item features relevance from past user interactions. In: Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, Association for Computing Machinery, New York, NY, USA, UMAP '17, pp. 275–279 (2017). https://doi.org/10.1145/3079628.3079695

Cheng, H., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., Shah, H.: Wide and deep learning for recommender systems. In: Karatzoglou A., Hidasi B., Tikk D., Shalom OS., Roitman H., Shapira B., Rokach L. (eds.) Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS@RecSys 2016, Boston, MA, USA, September 15, 2016, ACM, pp. 7–10 (2016). https://doi.org/10.1145/2988450.2988454

Dacrema, M.F., Cremonesi, P., Jannach, D.: Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In: Proceedings of the 13th ACM Conference on Recommender Systems, Association for Computing Machinery, New York, NY, USA, RecSys '19, pp. 101–109 (2019). https://doi.org/10.1145/3298689.3347058

Dacrema, M.F., Gasparin, A., Cremonesi, P.: Deriving item features relevance from collaborative domain knowledge. In: Anelli VW., Noia TD., Lops P., Musto C., Zanker M., Basile P., Bridge DG., Narducci F. (eds.) Proceedings of the Workshop on Knowledge-aware and Conversational Recommender Systems 2018 co-located with 12th ACM Conference on Recommender Systems, KaRS@RecSys 2018, Vancouver, Canada, October 7, 2018, CEUR-WS.org, CEUR Workshop Proceedings, vol. 2290, pp. 1–4 (2018). http://ceur-ws.org/Vol-2290/kars2018_paper1.pdf

Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 107–144. Springer, Berlin (2011). https://doi.org/10.1007/978-0-387-85820-3_4

Elbadrawy, A., Karypis, G.: User-specific feature-based similarity models for top-n recommendation of new items. ACM Trans. Intell. Syst. Technol. **6**(3), 33:1-33:20 (2015). https://doi.org/10.1145/2700495

Gantner, Z., Drumond, L., Freudenthaler, C., Rendle, S., Schmidt-Thieme, L.: Learning attribute-to-feature mappings for cold-start recommendations. In: Webb GI., Liu B., Zhang C., Gunopulos D., Wu X. (eds.) ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14–17 December 2010, IEEE Computer Society, pp. 176–185 (2010). https://doi.org/10.1109/ICDM.2010.129

Gevrey, M., Dimopoulos, I., Lek, S.: Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecol. Modell. **160**(3), 249–264 (2003). https://doi.org/10.1016/S0304-3800(02)00257-0

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. (2018). https://doi.org/10.1145/3236009

Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. In: Bengio Y., LeCun Y. (eds.) 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2–4, 2016, Conference Track Proceedings (2016a). http://arxiv.org/abs/1511.06939

Hidasi, B., Quadrana M, Karatzoglou, A., Tikk, D.: Parallel recurrent neural network architectures for feature-rich session-based recommendations. In: Sen S., Geyer W., Freyne J., Castells P. (eds.) Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15–19, 2016, ACM, pp. 241–248 (2016b). https://doi.org/10.1145/2959100.2959167

Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Bach FR., Blei DM. (eds.) Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015, JMLR.org, JMLR Workshop and Conference Proceedings, vol. 37, pp. 448–456 (2015). http://proceedings.mlr.press/v37/ioffe15.html

Lian, J., Zhang, F., Hou, M., Wang, H., Xie, X., Sun, G.: Practical lessons for job recommendations in the cold-start scenario. In: Proceedings of the Recommender Systems Challenge 2017, Association for Computing Machinery, New York, NY, USA, RecSys Challenge '17 (2017). https://doi.org/10.1145/3124791.3124794

Liang, D., Krishnan, R.G., Hoffman, M.D., Jebara, T.: Variational autoencoders for collaborative filtering. In: Champin P., Gandon F., Lalmas M., Ipeirotis PG. (eds.) Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018, ACM, pp. 689–698 (2018). https://doi.org/10.1145/3178876.3186150

Lops, P., de Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In: Ricci F., Rokach L., Shapira B., Kantor PB. (eds.) Recommender Systems Handbook, Springer, pp. 73–105 (2011). https://doi.org/10.1007/978-0-387-85820-3_3

Luhn, H.P.: A statistical approach to mechanized encoding and searching of literary information. IBM J. Res. Dev. **1**(4), 309–317 (1957). https://doi.org/10.1147/rd.14.0309

Montavon, G., Samek, W., Müller, K.: Methods for interpreting and understanding deep neural networks. Digit. Signal Process. **73**, 1–15 (2018). https://doi.org/10.1016/j.dsp.2017.10.011

Ni, J., Li, J., McAuley, J.J.: Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In: Inui K., Jiang J., Ng V., Wan X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019, Association for Computational Linguistics, pp. 188–197 (2019). https://doi.org/10.18653/v1/D19-1018

Ning, X., Karypis, G.: SLIM: sparse linear methods for top-n recommender systems. In: 11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11–14, 2011, pp. 497–506 (2011). https://doi.org/10.1109/ICDM.2011.134

Pan, F., Li, S., Ao, X., Tang, P., He, Q.: Warm up cold-start advertisements: Improving CTR predictions via learning to learn ID embeddings. In: Piwowarski B., Chevalier M., Gaussier É., Maarek Y., Nie J., Scholer F. (eds.) Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21–25, 2019, ACM, pp. 695–704 (2019). https://doi.org/10.1145/3331184.3331268

Paudel, B., Christoffel, F., Newell, C., Bernstein, A.: Updatable, accurate, diverse, and scalable recommendations for interactive applications. ACM Trans. Interact. Intell. Syst. **7**(1), 1:1-1:34 (2016). https://doi.org/10.1145/2955101

Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. In: Bilmes JA., Ng AY. (eds.) UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18–21, 2009, AUAI Press, pp. 452–461 (2009). https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1630&proceeding_id=25

Rendle, S., Freudenthaler, C.: Improving pairwise learning for item recommendation from implicit feedback. In: Carterette B., Diaz F., Castillo C., Metzler D. (eds.) Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24–28, 2014, ACM, pp. 273–282 (2014). https://doi.org/10.1145/2556195.2556248

Rendle, S.: Factorization machines. In: Webb GI., Liu B., Zhang C., Gunopulos D., Wu X. (eds.) ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14–17 December 2010, IEEE Computer Society, pp. 995–1000 (2010). https://doi.org/10.1109/ICDM.2010.127

Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-3. In: Harman DK. (ed.) Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2–4, 1994, National Institute of Standards and Technology (NIST), NIST Special Publication, vol. 500-225, pp. 109–126 (1994). http://trec.nist.gov/pubs/trec3/papers/city.ps.gz

Saveski, M., Mantrach, A.: Item cold-start recommendations: Learning local collective embeddings. In: Proceedings of the 8th ACM Conference on Recommender Systems, ACM, New York, NY, USA, RecSys '14, pp. 89–96 (2014). https://doi.org/10.1145/2645710.2645751

Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Ricci F., Rokach L., Shapira B., Kantor PB. (eds.) Recommender Systems Handbook, Springer, pp. 257–297 (2011). https://doi.org/10.1007/978-0-387-85820-3_8

Sharma, M., Zhou, J., Hu, J., Karypis, G.: Feature-based factorized bilinear similarity model for cold-start top-*n* item recommendation. In: Venkatasubramanian S., Ye J. (eds.) Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, April 30–May 2, 2015, SIAM, pp. 190–198 (2015). https://doi.org/10.1137/1.9781611974010.22

Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Precup D., Teh YW. (eds.) Proceedings of the 34th International Conference on Machine Learning, PMLR, International Convention Centre, Sydney, Australia, Proceedings of Machine Learning Research, vol. 70, pp. 3145–3153 (2017). http://proceedings.mlr.press/v70/shrikumar17a.html

Sparck Jones, K.: Document retrieval systems. Taylor Graham Publishing, London, UK, UK, chap A Statistical Interpretation of Term Specificity and Its Application in Retrieval, pp. 132–142 (1988). http://dl.acm.org/citation.cfm?id=106765.106782

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)

Steck, H.: Embarrassingly shallow autoencoders for sparse data. In: Liu L., White RW., Mantrach A., Silvestri F., McAuley JJ., Baeza-Yates R., Zia L. (eds.) The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13–17, 2019, ACM, pp. 3251–3257 (2019). https://doi.org/10.1145/3308558.3313710

Volkovs, M., Yu, G., Poutanen, T.: Dropoutnet: Addressing cold start in recommender systems. In: Guyon I., Luxburg UV., Bengio S., Wallach H., Fergus R., Vishwanathan S., Garnett R. (eds.) Advances in Neural Information Processing Systems 30, Curran Associates, Inc., pp. 4957–4966 (2017). http://papers.nips.cc/paper/7081-dropoutnet-addressing-cold-start-in-recommender-systems.pdf

Wei, J., He, J., Chen, K., Zhou, Y., Tang, Z.: Collaborative filtering and deep learning based recommendation system for cold start items. Expert Syst. Appl. **69**, 29–39 (2017). https://doi.org/10.1016/j.eswa.2016.09.040

Ziegler, C., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Ellis A., Hagino T. (eds.) Proceedings of the 14th International Conference on World Wide Web, WWW 2005, Chiba, Japan, May 10–14, 2005, ACM, pp. 22–32 (2005). https://doi.org/10.1145/1060745.1060754

Cesare Bernardis is a Ph.D. student at Politecnico di Milano (Polimi), Italy. He obtained his M.Sc. at Politecnico di Milano with a thesis on feature weighting for item cold-start recommendation. His research interests mainly include machine learning applications, recommender systems in particular. His actual research focuses on the evaluation and the implementation of new techniques.

Paolo Cremonesi is a professor of Computer Science Engineering at Politecnico di Milano (Polimi), Italy, and co-founder of Moviri, one of the first and most-successful start-ups from the Politecnico di Milano accelerator. His current research interests focus on the design and evaluation of machine learning algorithms, with a focus on recommender systems. He is author of more than 200 papers and co-inventor of 5 granted patents.