CuratorNet: Visually-aware Recommendation of Art Images

Pablo Messina[*]
Pontificia Universidad Católica
Santiago, Chile
pamessina@uc.cl

Manuel Cartagena
Pontificia Universidad Católica
Santiago, Chile
micartagena@uc.cl

Patricio Cerda
Pontificia Universidad Católica
Santiago, Chile
pcerdam@uc.cl

Felipe del Rio[†]
Pontificia Universidad Católica
Santiago, Chile
fidelrio@uc.cl

Denis Parra[‡]
Pontificia Universidad Católica
Santiago, Chile
dparra@ing.puc.cl

ABSTRACT

Although there are several visually-aware recommendation models in domains like fashion or even movies, the art domain lacks the same level of research attention, despite the recent growth of the online artwork market. To reduce this gap, in this article we introduce CuratorNet, a neural network architecture for visually-aware recommendation of art images. CuratorNet is designed at the core with the goal of maximizing generalization: the network has a fixed set of parameters that only need to be trained once, and thereafter the model is able to generalize to new users or items never seen before, without further training. This is achieved by leveraging visual content: items are mapped to item vectors through visual embeddings, and users are mapped to user vectors by aggregating the visual content of items they have consumed. Besides the model architecture, we also introduce novel triplet sampling strategies to build a training set for rank learning in the art domain, resulting in more effective learning than naive random sampling. With an evaluation over a real-world dataset of physical paintings, we show that CuratorNet achieves the best performance among several baselines, including the state-of-the-art model VBPR. CuratorNet is motivated and evaluated in the art domain, but its architecture and training scheme could be adapted to recommend images in other areas.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Machine learning approaches**; • **Applied computing** → *Media arts*.

KEYWORDS

recommender systems, neural networks, visual art

1 INTRODUCTION

The big revolution of deep convolutional neural networks (CNN) in the area of computer vision for tasks such as image classification [18, 27, 41], object recognition [1], image segmentation [3] or scene identification [40] has reached the area of image recommender systems in recent years [19, 20, 22, 29, 30, 32]. These works use neural visual embeddings to improve the recommendation performance compared to previous approaches for image recommendation based on ratings and text [2], social tags [39], context [5] and manually crafted visual features [43]. Regarding application domains of recent image recommendation methods using neural visual embeddings, to the best of our knowledge most of them focus on fashion recommendation [20, 22, 30], a few on art recommendation [19, 32] and photo recommendation [29]. He et al. [19] proposed Vista, a model combining neural visual embeddings, collaborative filtering as well as temporal and social signals for digital art recommendation.

However, digital art projects can differ significantly from physical art (paintings and photographs). Messina et al. [32] study recommendation of paintings in an online art store using a simple k-NN model based on neural visual features and metadata. Although memory-based models perform fairly well, model-based methods using neural visual features report better performance [19, 20] in the fashion domain, indicating room for improvement in this area, considering the growing sales in the global online artwork market[1].

The most popular model-based method for image recommendation using neural visual embeddings is VBPR [20], a state-of-the-art model that integrates implicit feedback collaborative filtering with neural visual embeddings into a Bayesian Personalized Ranking (BPR) learning framework [33]. VBPR performs well, but it has some drawbacks. VBPR learns a latent embedding for each user and for each item, so new users cannot receive suggestions and new items cannot be recommended until re-training is carried out. An alternative is training a model such as Youtube's Deep Neural Recommender [8] which allows to recommend to new users with little preference feedback and without additional model training. However, Youtube's model was trained on millions of user transactions and with large amounts of profile and contextual data, so it does not easily fit to datasets that are small, with little user feedback or with little contextual and profile data.

In this work, we introduce a neural network for visually-aware recommendation of images focused on visual art named *CuratorNet*, whose general structure can be seen in Figure 1. CuratorNet leverages neural image embeddings as those obtained from CNNs [18, 27, 41] pre-trained on the Imagenet dataset (ILSVRC [36]). We train CuratorNet for ranking with triplets (P_u, i_+, j_-), where P_u is the history of image preferences of a user u, whereas i_+ and j_- are a pair of images with higher and lower preference respectively. CuratorNet draws inspiration from VBPR [20] and Youtube's Recommender System [8]. VBPR [20] inspired us to leverage pre-trained image embeddings as well as optimizing the model for ranking as in

[*] Also with Millennium Institute Foundational Research on Data, IMFD.
[†] Also with Millennium Institute Foundational Research on Data, IMFD.
[‡] Also with Millennium Institute Foundational Research on Data, IMFD.

[1] https://www.artsy.net/article/artsy-editorial-global-art-market-reached-674-billion-2018-6

BPR [33]. From the work of Convington et al. [8] we took the idea of designing a deep neural network that can generalize to new users without introducing new parameters or further training (unlike VBPR which needs to learn a latent user vector for each new user). As a result, CuratorNet can recommend to new users with very little feedback, and without additional training *CuratorNet*'s deep neural network is trained for personalized ranking using triplets and the architecture contains a set of layers with shared weights, inspired by models using triplet loss for non-personalized image ranking [38, 44]. In these works, a single image represents the input query, but in our case, the input query is a set images representing a user preference history, P_u. In summary, compared to previous works, our main contributions are:

- a novel neural-based visually-aware architecture for image recommendation,
- a set of sampling guidelines for the creation of the training dataset (triplets), which improve the performance of *CuratorNet* as well as *VBPR* with respect to random negative sampling, and
- presenting a thorough evaluation of the method against competitive state-of-the-art methods (VisRank [22, 32] and VBPR[20]) on a dataset of purchases of physical art (paintings and photographs).

We also share the dataset[2] of user transactions (with hashed user and item IDs due to privacy requirements) as well as visual embeddings of the paintings image files. One aspect to highlight about this research, is that although the triplets' sampling guidelines to build the BPR training set apply specifically to visual art, the architecture of *CuratorNet* can be used in other visual domains for image recommendation.

2 RELATED WORK

In this section we provide an overview of relevant related work, considering: *Artwork Recommender Systems* (2.1), *Visually-aware Recommender Systems* (2.2), as well as highlights of what differentiates our work to the existing literature.

2.1 Artwork Recommender Systems

With respect to artwork recommender systems, one of the first contributions was the CHIP Project [2]. The aim of the project was to build a recommendation system for the Rijksmuseum. The project used traditional techniques such as content-based filtering based on metadata provided by experts, as well as collaborative filtering based on users' ratings. Another similar system but non-personalized was *m4art* by Van den Broek et al. [43], who used color histograms to retrieve similar art images given a painting as input query.

Another important contribution is the work by Semeraro et al. [39], who introduced an artwork recommender system called FIRSt (Folksonomy-based Item Recommender syStem) which utilizes social tags given by experts and non-experts of over 65 paintings of the Vatican picture gallery. They did not employ visual features among their methods. Benouaret et al. [5] improved the state-of-the-art in artwork recommender systems using context obtained through a mobile application, with the aim of making museum tour recommendations more useful. Their content-based approach used ratings given by the users during the tour and metadata from the artworks rated, e.g. title or artist names.

Finally, the most recent works use neural image embeddings [19, 32]. He et al. [19] propose the system Vista, which addresses digital artwork recommendations based on pre-trained deep neural visual features, as well as temporal and social data. On the other hand, Messina et al. [32] address the recommendation of one-of-a-kind physical paintings, comparing the performance of metadata, manually-curated visual features, and neural visual embeddings. Messina et al. [32] recommend to users by computing a simple K-NN based similarity score among users' purchased paintings and the paintings in the dataset, a method that Kang et al. [22] call *VisRank*.

2.2 Visually-aware Image Recommender Systems

In this section we survey works using visual features to recommend images. We also cite a few works using visual information to recommend non-image items, but these are not too relevant for the present research.

Manually-engineered visual features extracted from images (texture, sharpness, brightness, etc.) have been used in several tasks for information filtering, such as retrieval [28, 35, 43] and ranking [37]. More recently, interesting results have been shown for the use of low-level handcrafted stylistic visual features automatically extracted from video frames for content-based video recommendation [11]. Even better results are obtained when both stylistic visual features and annotated metadata are combined in a hybrid recommender, as shown in the work of Elahi et al. [13]. In a visually-aware setting not related to recommending images, Elsweiller et al. [14] used manually-crafted attractiveness visual features [37], in order to recommend healthy food recipes to users.

Another branch of visually-aware image recommender systems focuses on using neural embeddings to represent images [19, 20, 22, 29, 32]. The computer vision community has a large track of successful systems based on neural networks for several tasks [1, 3, 18, 27, 40, 41]. This trend started from the outstanding performance of the AlexNet [27] in the Imagenet Large Scale Visual Recognition challenge (ILSVRC [36]), but the most notable implication is that the neural image embeddings have shown impressive performance for transfer learning, i.e., for tasks different from the original one [10, 26]. Usually these neural image embeddings are obtained from CNN models such as AlexNet [27], VGG [41] and ResNet [18], among others. Motivated by these results, McAuley et al. [30] introduced an image-based recommendation system based on styles and substitutes for clothing using visual embeddings pre-trained on a large-scale dataset obtained from Amazon.com. Later, He et al. [20] went further in this line of research and introduced a visually-aware matrix factorization approach that incorporates visual signals (from a pre-trained CNN) into predictors of people's opinions, called VBPR. Their training model is based on Bayesian Personalized Ranking (BPR), a model previously introduced by Rendle et al. [33].

[2] https://drive.google.com/drive/folders/1Dk7_BRNtN_IL8r64xAo6GdOYEycivtLy

The next work by He et al. [19] deals with visually-aware digital art recommendation, building a model called Vista which combines ratings, temporal and social signals and visual features.

Another relevant work was the research by Lei et al. [29] who introduced comparative deep learning for hybrid image recommendation. In this work, they use a siamese neural network architecture for making recommendations of images using user information (such as demographics and social tags) as well as images in pairs (one liked, one disliked) in order to build a ranking model. The approach is interesting, but they work with Flickr photos, not artwork images, and use social tags, not present in our problem setting. The work by Kang et al. [22] expands VBPR but they focus on generating images using Generative adversarial networks [17] rather than recommending, with an application in the fashion domain. Finally, Messina et al. [32] was already mentioned, but we can add that their neural image embeddings outperformed other visual (manually-extracted) and metadata features for ranking, with the exception of the metadata given by user's favorite artist, which predicted even better than neural embeddings for top@k recommendation.

2.3 Differences to Previous Research

Almost all the surveyed articles on artwork recommendation have in common that they used standard techniques such as collaborative filtering and content-based filtering, as well as manually-curated visual image features, but only the most recent works have exploited visual features extracted from CNNs [19, 32]. In comparison to these works, we introduce a model-based approach (unlike the memory-based VisRank method by Messina et al. [32]) and which recommends to cold-start items and users without additional model training (unlike [19]). With regards to more general work on visually-aware image recommender systems, almost all of the surveyed articles have focused on tasks different from art recommendation, such as fashion recommendation [20, 22, 30], photo [29] and video recommendation [13]. Only Vista, the work by He et al. [19], resembles ours in terms of the topic (art recommendation) and the use of visual features. Unlike them, we evaluate our proposed method, CuratorNet, in a dataset of physical paintings and photographs, not only digital art. Moreover, Vista uses social and temporal metadata which we do not have and many other datasets might not have either. Compared to all these previous research, and to the best of our knowledge, CuratorNet is the first architecture for image recommendation that takes advantage of shared weights in a triplet loss setting, an idea inspired by the results of Wang et al. [44] and Schroff et al. [38], but here adapted to the personalized image recommendation domain.

3 CURATORNET

3.1 Problem Formulation

We approach the problem of recommending art images from user positive-only feedback (e.g., purchase history, likes, etc.) upon visual items (paintings, photographs, etc.). Let U and I be the set of users and items in a dataset, respectively. We assume only one image per each single item $i \in I$. Considering either user purchases or likes, the set of items for which a user u has expressed positive preference is defined as I_u^+. In this work, we considered purchases to be positive feedback from the user. Our goal is to generate for each user $u \in U$ a personalized ranked list of the items for which the user still have not expressed preference, i.e., for $I \setminus I_u^+$.

Table 1: Notation for CuratorNet.

Symbol	Description
U, I	user set, item set
u	a specific user
i, j	a specific item (resp.)
i_+, j_-	a positive item and negative item (resp.)
I_u^+ or P_u	set of all items which the user u has expressed a positive preference (full history)
$I_{u,k}^+$	set of all items which the user u has expressed a positive preference up to his k-th purchase basket (inclusive)
$P_{u,k}$	set of all items which the user u has expressed a positive preference in his k-th purchase basket

3.2 Preference Predictor

The preference predictor in CuratorNet is inspired by VBPR [20], a state-of-the-art visual recommender model.

However, **CuratorNet has some important differences**. First, we do not use non-visual latent factors, so we remove the traditional user and item non-visual latent embeddings. Second, we do not learn a specific embedding per user such as VBPR, but we learn a joint model that, given a user's purchase/like history, it outputs a single embedding which can be used to rank unobserved artworks in the dataset, similar to YouTube's Deep Learning network [8]. Another important difference of VBPR with CuratorNet is that the former has a single matrix $\mathbf{E}$ to project a visual item embedding f_i into the user latent space. In CuratorNet, we rather learn a neural network $\Phi(\cdot)$ to perform that projection, which receives as input either a single image embedding $\mathbf{f}_i$ or a set of image embeddings representing users' purchase/like history $P_u = \{\mathbf{f}_1, ..., \mathbf{f}_N\}$. Given all these aspects, the preference predictor of CuratorNet is given by:

$$x_{u,i} = \alpha + \beta_u + \Phi(P_u)^T \Phi(\mathbf{f}_i) \tag{1}$$

where α is an offset, β_u represents a user bias, $\Phi(\cdot)$ represents CuratorNet neural network and P_u represents the set of visual embeddings of the images in user u history. After some experiments we found no differences between using or not a variable for item bias β_i so we dropped it in order to decrease the number of parameters (Occam's razor).

Finally, since we calculate the model parameters using BPR [33], the parameters α, β_u cancel out (details in the coming subsection) and our final preference predictor is simply

$$x_{u,i} = \Phi(P_u)^T \Phi(\mathbf{f}_i) \tag{2}$$

3.3 Model Learning via BPR

We use the Bayesian Personalized Ranking (BPR) framework [33] to learn the model parameters. Our goal is to optimize ranking by training a model which orders triples of the form $(u, i, j) \in \mathcal{D}_\mathcal{S}$, where u denotes a user, i an item with positive feedback from u,

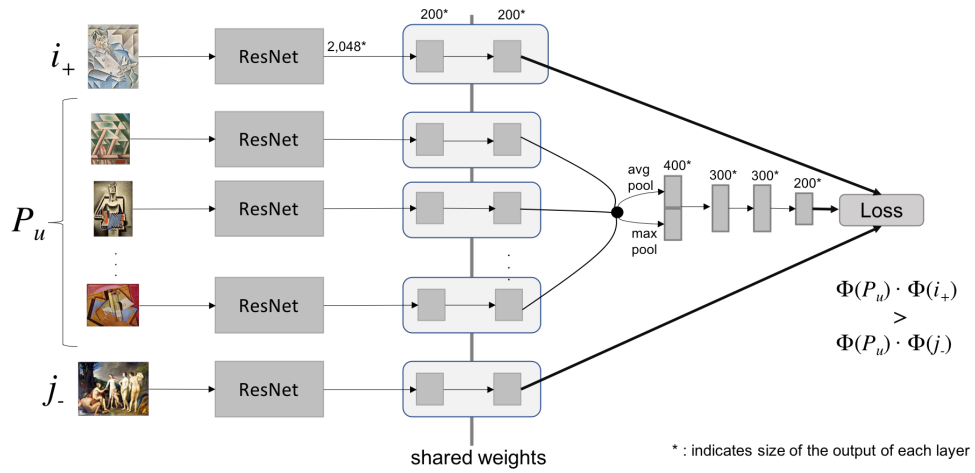

Figure 1: Architecture of CuratorNet showing in detail the layers with shared weights for training.

and j an item with non-observed feedback from u. The training set of triples $\mathcal{D}_{\mathcal{S}}$ is defined as:

$$\mathcal{D}_{\mathcal{S}} = \{(u, i, j) | u \in U \wedge i \in I_u^+ \wedge j \in I \setminus I_u^+\} \quad (3)$$

Table 1 shows that I_+^u denotes the set of all items with positive feedback from u while $I \setminus I_+^u$ shows those items without such positive feedback. Considering our previously defined preference predictor $x_{u,i}$, we would expect a larger preference score of u over i than over j, then BPR defines the difference between scores

$$x_{u,i,j} = x_{u,i} - x_{u,j} \quad (4)$$

an then BPR aims at finding the parameters Θ which optimize the objective function

$$\operatorname*{argmax}_{\Theta} \sum_{\mathcal{D}_{\mathcal{S}}} \ln \sigma(x_{u,i,j}) - \lambda_{\Theta} ||\Theta||^2 \quad (5)$$

where $\sigma(\cdot)$ is the sigmoid function, Θ includes all model parameters, and λ_{Θ} is a regularization hyperparameter.

In CuratorNet, unlike BPR-MF [33] and VBPR [20], we use a sigmoid cross entropy loss, considering that we can interpret the decision over triplets as a binary classification problem, where if $x_{u,i,j} > 0$ represents class $c = 1$ (triple well ranked, since $x_{u,i} > x_{u,j}$) and $x_{u,i,j} \leq 0$ signifies class $c = 0$ (triplet wrongly ranked, since $x_{u,i} \leq x_{u,j}$). Then, CuratorNet loss can be expressed as:

$$\mathcal{L} = -\sum_{\mathcal{D}_{\mathcal{S}}} c \ln(\sigma(x_{u,i,j})) + (1-c) \ln(1 - \sigma(x_{u,i,j})) + \lambda_{\Theta} ||\Theta||^2 \quad (6)$$

where $c \in \{0, 1\}$ is the class, Θ includes all model parameters, λ_{Θ} is a regularization hyperparameter, and $\sigma(x_{u,i,j})$ is the probability that a user u really prefers i over j, $P(i >_u j|\Theta)$ [33], calculated with the sigmoid function, i.e.,

$$P(i >_u j|\Theta) = \sigma(x_{u,i,j}) = \frac{1}{1 + e^{-(x_{u,i} - x_{u,j})}} \quad (7)$$

We perform the optimization to learn the parameters which reduce the loss function $\mathcal{L}$ by stochastic gradient descent with the Adam optimizer [23], using the implementation in Tensorflow[3]. During each iteration of stochastic gradient descent, we sample a user u, a positive item $i \in I_+^u$ (i.e., removed from P_u), a negative item $j \in I \setminus I_+^u$, and user u purchase/like history with item i removed, i.e., $P_u \setminus i$.

3.4 Model Architecture

The architecture of the CuratorNet neural network is summarized in Figure 1, but is presented with more details in Figure 1. For training, each imput instance is expected to be a triple (P_u,i,j), where P_u is the set of images in user u history (purchases, likes) with a single item i removed from the set, i is an item with positive preference, and j is an item with *assumed* negative user preference. The negative user preference is assumed since the item j is sampled from the list of images which u has not interacted with yet. Each image (i, j and all images $\in P_u$) goes through a ResNet [18] (pre-trained with ImageNet data), which outputs a visual image embedding in $\mathbb{R}^{2,048}$. ResNet weights are fixed during CuratorNet's training. Then, the network has two layers with scale exponential linear units (hereinafter, SELU [24]), with 200 neurons each, which reduce the dimensionality of each image. Notice that these two layers work similar to a siamese [7] or triplet loss architecture [38, 44], i.e., they have shared weights. Each image is represented at the output of this section of the network by a vector in $\mathbb{R}^{200}$. Then, for the case of the images in P_u, their embeddings are both averaged (average pooling [6]) as well as max-pooled per dimension (max pooling [6]) , and next concatenated to a resultant vector in $\mathbb{R}^{400}$. Finally, three SELU consecutive layers of 300, 200, and 200 neurons respectively end up with an output representation for P_u in $\mathbb{R}^{200}$. The final part of the network is a ranking layer which evaluates a loss such that $\Phi(P_u) \cdot \Phi(i) > \Phi(P_u) \cdot \Phi(j)$, where replacing in Equation (2), we have $x_{u,i} > x_{uj}$. There are several options of loss functions, but due to good results of the cross-entropy loss in similar architectures with shared weights [25] rather than, e.g. the hinge loss where we

[3] A reference CuratorNet implementation may be found at https://github.com/ialab-puc/CuratorNet.

need to optimize an additional margin parameter m, we chose the sigmoid cross-entropy for CuratorNet.

Notice that in this article we used a pre-trained ResNet [18] to obtain the image visual features, but the model could use other CNNs such as AlexNet [27], VGG [41], etc. We chose ResNet since it has performed the best in transfer learning tasks [10, 26].

3.5 Data Sampling for Training

The original BPR article [33] suggests the creation of training triples (u, i_+, j_-) simply by, given a user u, randomly sampling a positive element i_+ among those consumed, as well as sampling a negative feedback element j_- among those not consumed. However, eventual research has shown that there are more effective ways to create these training triples [12]. In our case, we define some guidelines to sample triples for the training set based on analyses from previous studies indicating features which provide signals of user preference. For instance, Messina et al. [32] showed that people are very likely to buy several artworks with similar visual themes, as well as from the same artist, then we used *visual clusters* and *user's favorite artist* to set some of these sampling guidelines.

Creating Visual Clusters. Some of the sampling guidelines are based on visual similarity of the items, and although we have some metadata for the images in the dataset, there is a significant number of missing values: only 45% of the images have information about subject (e.g., architecture, nature, travel) and 53% about style (e.g., abstract, surrealism, pop art). For this reason, we conduct a clustering of images based on their visual representation, in such a way that items with visual embeddings that are too similar will not be used to sample positive/negative pairs (i_+, j_-). To obtain these visual clusters, we followed the following procedure: (i) Conduct a Principal Component Analysis to reduce the dimensionality of images embedding vectors from $\mathbb{R}^{2,048}$ to $\mathbb{R}^{200}$, (ii) perform k-means clustering with 100 clusters. We conducted k-means clustering 20 times and for each time we calculated the Silhouette coefficient [34] (an intrinsic metric of clustering quality), so we kept the clustering resulting with the highest Silhouette value. Finally, (iii) we assign each image the label of its respective visual cluster. Samples of our clusters in a 2-dimensional projection map of images, built with the UMAP method [31], can be seen in Figure 2.

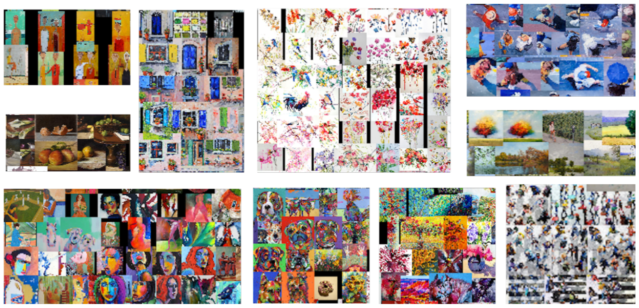

Figure 2: Examples of visual clusters automatically generated to sample triples for the training set.

Guidelines for sampling triples. We generate the training set $\mathcal{D}_\mathcal{S}$ as the union of multiple disjoint[4] training sets, each one generated with a different strategy in mind. These strategies and their corresponding training sets are:

(1) Removing item from purchase basket, and predicting this missing item.
(2) Sort items purchased sequentially, and then predict next purchase in basket.
(3) Recommending visually similar artworks from the favorite artists of a user.
(4) Recommending profile items from the same user profile.
(5) Create an artificial user profile of a single item purchased, and recommending profile items given this artificially created user profile.
(6) Create artificial profile with a single item, then recommend visually similar items from the same artist.

Finally, the training set $\mathcal{D}_\mathcal{S}$ is formally defined as:

$$\mathcal{D}_\mathcal{S} = \bigcup_{i=1}^{6} \mathcal{D}_\mathcal{S}{}^{i} \tag{8}$$

In practice, we uniformly sample about 10 million training triples, distributed uniformly among the six training sets $\mathcal{D}_\mathcal{S}{}^{i}$. Likewise, we sample about 300,000 validation triples. To avoid sampling identical triples, we hash them and compare the hashes to check for potential collisions. Before sampling the training and validation sets, we hide the last purchase basket of each user, using them later on for testing.

4 EXPERIMENTS

4.1 Datasets

For our experiments we used a dataset where the user preference is in the form of purchases over physical art (painting and pictures). This private dataset was collected and shared by an online art store. The dataset consists of 2, 378 users, 6, 040 items (paintings and photographs) and 5, 336 purchases. On average, each user bought 2-3 items. One important aspect of this dataset is that paintings are one-of-a-kind, i.e., there is a single instance of each item and once it is purchased, is removed from the inventory. Since most of the items in the dataset are one-of-a-kind paintings (78%) and most purchase transactions have been made over these items (81.7%) a method relying on collaborative filtering model might suffer in performance, since user co-purchases are only possible on photographs. Another notable aspect in the dataset is that each item has a single creator (artist). In this dataset there are 573 artists, who have uploaded 10.54 items in average to the online art store.

The dataset[5] with transaction tuples (user, item), as well as the tuples used for testing (the last purchase of each user with at least two purchases) are available for replicating our results as well as for training other models. Due to copyright restrictions we cannot share the original image files, but we share the embeddings of the images obtained with ResNet50 [18].

[4] Theoretically, these training sets are not perfectly disjoint, but in practice we hash all training triples and make sure no two training triples have the same hash. This prevents duplicates from being added to the final training set.
[5] https://drive.google.com/drive/folders/1Dk7_BRNtN_IL8r64xAo6GdOYEycivtLy

4.2 Evaluation Methodology

In order to build and test the models, we split the data into train, validation and test sets. To make sure that we could make recommendations for all cases in the test set, and thus make a fair comparison among recommendation methods, we check that every user considered in the test set was also present in the training set. All baseline methods were trained on the training set with hyperparameters tuned with the validation set.

Next, the trained models are used to report performance over different metrics on the test set. For the dataset, the test set consists of the last transaction from every user that purchased at least twice, the rest of previous purchases are used for train and validation.

Metrics. To measure the results we used several metrics: AUC (also used in [19, 20, 22]), normalized discounted cumulative gain (nDCG@k)[21], as well as Precision@k and Recall@k [9]. Although it might seem counter-intuitive, we calculate these metrics for a low (k=20) as well as high values of k (k = 100). Most research on top-k recommendation systems focuses on the very top of the recommendation list, (k=5,10,20). However, Valcarce et al. [42] showed that top-k ranking metrics measured at higher values of k (k=100, 200) are specially robust to biases such as sparsity and popularity biases. The sparsity bias refers to the lack of known relevance for all the user-items pairs, while the popularity bias is the tendency of popular items to receive more user feedback, then missing user-items are *not missing at random*. We are specially interested in preventing popularity bias since we want to recommend not only from the artists that each user is commonly purchasing from. We aim at promoting novelty as well as discovery of relevant art from newcomer artists.

4.3 Baselines

The methods used in the evaluation are the following:

(1) **CuratorNet**: The method described in this paper. We also test it with four regularization values for $\lambda = \{0, .01, .001, .0001\}$.
(2) **VBPR** [20]: The state-of-the-art. We used the same embedding size as in CuratorNet (200), we optimized it until converge in the training set and we also tested the four regularization values for $\lambda = \{0, .01, .001, .0001\}$.
(3) **VisRank** [22, 32]: This is a simple memory-based content filtering method that ranks a candidate painting i for a user u based on the maximum cosine similarity with some existing item in the user profile $j \in P_u$ i.e.

$$score(u, i) = max_{j \in P_u} cosine(i, j) \tag{9}$$

5 RESULTS AND DISCUSSION

In Table 2, we can see the results comparing all methods. As reference, at the top rows we present an oracle (perfect ranking), and in the bottom row a random recommender. Notice that AUC for a random recommender should be theoretically =0.5 (sorting pairs of items given a user), so the AUC= .4973 serves as a check. In terms of AUC, Recall@100, and Precision@100 CuratorNet with a small regularization (λ = .0001) is the top model among other methods. We highlight the following points from these results:

- CuratorNet, with a small regularization λ = .0001, outperforms the other methods in five metrics (AUC, Precision@20, Recall@100, Precision@100 and nDCG@100), while it stands second in Recall@20 and nDCG@20 against the non-regularized version of CuratorNet. This implies that CuratorNet overall ranks very well at top positions, and is specially robust against sparsity and popularity bias [42]. In addition, CuratorNet seems robust to changes in the regularization hyperparameter.
- Compared to VBPR, CuratorNet is better in all seven metrics (AUC, Precision@20, Recall@100, Precision@100 and nDCG@100). Notably, it is also more robust to the regularization hyperparameter λ than VBPR. We think that this is explained in part due to the characteristics of the dataset: VBPR exploits non-visual co-occurrance patterns, but in our dataset this signal provides a rather small preference information, since almost 80% are one-of-a-kind items and transactions.
- VisRank presents very competitive results, specially in terms of AUC, nDCG@20 and nDCG@100, performing better than VBPR in this high one-of-a-kind dataset. However, CuratorNet performs better than VisRank in all metrics. This provides evidence that the model-based approach of CuratorNet that aggregates user preferences into a single embedding is a better approach than the heuristic-based scoring of VisRank.

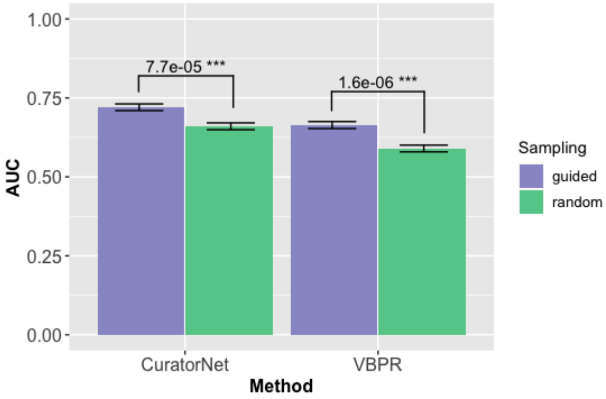

Figure 3: The sampling guidelines had a positive effect on AUC compared to random negative sampling for building the BPR training set.

5.1 Effect of Sampling Guidelines

We studied the effect of using our sampling guidelines for building the training set $\mathcal{D}_\mathcal{S}$ compared to the traditional BPR setting where negative samples j are sampled uniformly at random from the set of unobserved items by the user, i.e., $I \setminus I_u^+$. In the case of CuratorNet we use all six sampling guidelines (${\mathcal{D}_\mathcal{S}}^1 - {\mathcal{D}_\mathcal{S}}^6$), while in VBPR we only used two sampling guidelines (${\mathcal{D}_\mathcal{S}}^3$ and ${\mathcal{D}_\mathcal{S}}^4$), since VBPR has no notion of session or purchase baskets in its original formulation, and it has more parameters than CuratorNet to model collaborative non-visual latent preferences. We tested AUC in both CuratorNet and VBPR, under their best performance with regularization parameter λ, with and without our sampling guidelines. Notice that results in Table 2 all consider the use of our sampling guidelines. After conducting pairwise t-tests, we found a significant improvement in CuratorNet and VBPR, as shown in

Table 2: Results for all methods, sorted by AUC performance. The top five results are highlighted for each metric. For reference, the bottom row presents a random recommender, while the top row presents results of a perfect Oracle.

Method	λ (L2 Reg.)	AUC	R@20	P@20	nDCG@20	R@100	P@100	nDCG@100
Oracle	–	**1.0000**	**1.0000**	**.0655**	**1.0000**	**1.0000**	**.0131**	**1.0000**
CuratorNet	.0001	**.7204**	**.1683**	**.0106**	**.0966**	**.3200**	**.0040**	**.1246**
CuratorNet	.001	**.7177**	**.1566**	**.0094**	**.0895**	**.2937**	**.0037**	**.1160**
VisRank	–	**.7151**	**.1521**	**.0093**	**.0956**	**.2765**	**.0034**	**.1195**
CuratorNet	0	**.7131**	**.1689**	**.0100**	**.0977**	**.3048**	**.0038**	**.1239**
CuratorNet	.01	.7125	.1235	.0075	.0635	.2548	.0032	.0904
VBPR	.0001	.6641	.1368	.0081	.0728	.2399	.0030	.0923
VBPR	0	.6543	.1287	.0078	.0670	.2077	.0026	.0829
VBPR	.001	.6410	.0830	.0047	.0387	.1948	.0024	.0620
VBPR	.01	.5489	.0101	.0005	.0039	.0506	.0006	.0118
Random	–	.4973	.0103	.0006	.0041	.0322	.0005	.0098

Figure 3. CuratorNet with sampling guidelines (AUC=.7204) had a significant improvement over CuratorNet with random negative sampling (AUC=.6602), $p = 7.7 \cdot 10^{-5}$. Likewise, VBPR with guidelines (AUC=.6641) had a significant improvement compared with VBPR with random sampling (AUC=.5899), $p = 1.6 \cdot 10^{-6}$. With this result, we conclude that the proposed sampling guidelines help in selecting better triplets for more effective learning in our art image recommendation setting.

6 CONCLUSION

In this article we have introduced CuratorNet, an art image recommender system based on neural networks. The learning model of CuratorNet is inspired by VBPR [20], but it incorporates some additional aspects such as layers with shared weights and it works specially well in situations of one-of-a-kind items, i.e., items which disappear from the inventory once consumed, making difficult to user traditional collaborative filtering. Notice that an important contribution of this article are the data shared, since we could not find on the internet any other dataset of user transactions over physical paintings. We have anonymized the user and item IDs and we have provided ResNet visual embeddings to help other researchers building and validating models with these data.

Our model outperforms state-of-the-art VBPR as well as other simple but strong baselines such as VisRank [22, 32]. We also introduce a series of guidelines for sampling triples for the BPR training set, and we show significant improvements in performance of both CuratorNet and VBPR versus traditional random sampling for negative instances.

Future Work. Among our ideas for future work, we will test our neural architecture using end-to-end-learning, in a similar fashion than [22] who used a light model called CNN-F to replace the pre-trained AlexNet visual embeddings. Another idea we will test is to create explanations for our recommendations based on low-level (textures) and high level (objects) visual features which some recent research are able to identify from CNNs, such as the Network Dissection approach by Bau et al. [4]. Also, we will explore ideas from the research on image style transfer [15, 16], which might help us to identify styles and then use this information as context to produce style-aware recommendations. Another interesting idea for future work is integrating multitask learning in our framework, such as the recently published paper on the newest Youtube recommender [45]. Finally, from a methodological point-of-view, we will test other datasets with *likes* rather than *purchases*, since we aim at understanding how the model will behave under a different type of user relevance feedback.

ACKNOWLEDGMENTS

This work has been supported by the Millennium Institute for Foundational Research on Data (IMFD) and by the Chilean research agency ANID, FONDECYT grant 1191791.

REFERENCES

[1] S. Akçay, M. E. Kundegorski, M. Devereux, and T. P. Breckon. 2016. Transfer learning using convolutional neural networks for object classification within X-ray baggage security imagery. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. 1057–1061.

[2] LM Aroyo, Y Wang, R Brussee, Peter Gorgels, LW Rutledge, and N Stash. 2007. Personalized museum experience: The Rijksmuseum use case. In *Proceedings of Museums and the Web*.

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence* 39, 12 (2017), 2481–2495.

[4] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6541–6549.

[5] Idir Benouaret and Dominique Lenne. 2015. Personalizing the museum experience through context-aware recommendations. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 743–748.

[6] Y-Lan Boureau, Jean Ponce, and Yann LeCun. 2010. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 111–118.

[7] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. 2005. Learning a similarity metric discriminatively, with application to face verification. In *CVPR (1)*. 539–546.

[8] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 191–198.

[9] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 39–46.

[10] Felipe del Rio, Pablo Messina, Vicente Dominguez, and Denis Parra. 2018. Do Better ImageNet Models Transfer Better... for Image Recommendation?. In *2nd workshop on Intelligent Recommender Systems by Knowledge Transfer and Learning*. https://arxiv.org/abs/1807.09870

[11] Yashar Deldjoo, Mehdi Elahi, Paolo Cremonesi, Franca Garzotto, Pietro Piazzolla, and Massimo Quadrana. 2016. Content-based video recommendation system based on stylistic visual features. *Journal on Data Semantics* 5, 2 (2016), 99–113.

[12] Jingtao Ding, Fuli Feng, Xiangnan He, Guanghui Yu, Yong Li, and Depeng Jin. 2018. An improved sampler for bayesian personalized ranking by leveraging view data. In *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 13–14.

[13] Mehdi Elahi, Yashar Deldjoo, Farshad Bakhshandegan Moghaddam, Leonardo Cella, Stefano Cereda, and Paolo Cremonesi. 2017. Exploring the Semantic Gap for Movie Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (Como, Italy) *(RecSys '17)*. 326–330.

[14] David Elsweiler, Christoph Trattner, and Morgan Harvey. 2017. Exploiting food choice biases for healthier recipe recommendation. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval*. ACM, 575–584.

[15] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2414–2423.

[16] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. 2017. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv preprint arXiv:1705.06830* (2017).

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[19] Ruining He, Chen Fang, Zhaowen Wang, and Julian McAuley. 2016. Vista: A Visually, Socially, and Temporally-aware Model for Artistic Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems* (Boston, Massachusetts, USA) *(RecSys '16)*. 309–316.

[20] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from implicit feedback. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. 144–150.

[21] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[22] Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian McAuley. 2017. Visually-aware fashion recommendation and design with generative image models. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 207–216.

[23] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[24] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*. 971–980.

[25] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, Vol. 2.

[26] Simon Kornblith, Jonathon Shlens, and Quoc V Le. 2018. Do better imagenet models transfer better? *arXiv preprint arXiv:1805.08974* (2018).

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of Advances in neural information processing systems 25 (NIPS)*. 1097–1105.

[28] Marco La Cascia, Saratendu Sethi, and Stan Sclaroff. 1998. Combining textual and visual cues for content-based image retrieval on the world wide web. In *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries*. 24–28.

[29] Chenyi Lei, Dong Liu, Weiping Li, Zheng-Jun Zha, and Houqiang Li. 2016. Comparative Deep Learning of Hybrid Representations for Image Recommendations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2545–2553.

[30] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 43–52.

[31] L. McInnes, J. Healy, and J. Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints* (Feb. 2018). arXiv:1802.03426 [stat.ML]

[32] Pablo Messina, Vicente Dominguez, Denis Parra, Christoph Trattner, and Alvaro Soto. 2018. Content-based artwork recommendation: integrating painting metadata with neural and manually-engineered visual features. *User Modeling and User-Adapted Interaction* (2018), 1–40.

[33] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. 452–461.

[34] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.

[35] Yong Rui, Thomas S Huang, Michael Ortega, and Sharad Mehrotra. 1998. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on circuits and systems for video technology* 8, 5 (1998), 644–655.

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115, 3 (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[37] Jose San Pedro and Stefan Siersdorfer. 2009. Ranking and Classifying Attractiveness of Photos in Folksonomies. In *Proceedings of the 18th International Conference on World Wide Web* (Madrid, Spain) *(WWW '09)*. 771–780.

[38] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.

[39] Giovanni Semeraro, Pasquale Lops, Marco De Gemmis, Cataldo Musto, and Fedelucio Narducci. 2012. A folksonomy-based recommender system for personalized access to digital artworks. *Journal on Computing and Cultural Heritage (JOCCH)* 5, 3 (2012), 11.

[40] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 806–813.

[41] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

[42] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. 2018. On the robustness and discriminative power of information retrieval metrics for top-N recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 260–268.

[43] Egon L van den Broek, Thijs Kok, Theo E Schouten, and Eduard Hoenkamp. 2006. Multimedia for art retrieval (m4art). In *Multimedia Content Analysis, Management, and Retrieval 2006*, Vol. 6073. International Society for Optics and Photonics, 60730Z.

[44] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1386–1393.

[45] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending What Video to Watch Next: A Multitask Ranking System. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19)*. ACM, New York, NY, USA, 43–51. https://doi.org/10.1145/3298689.3346997