Learning Image and User Features for Recommendation in Social Networks

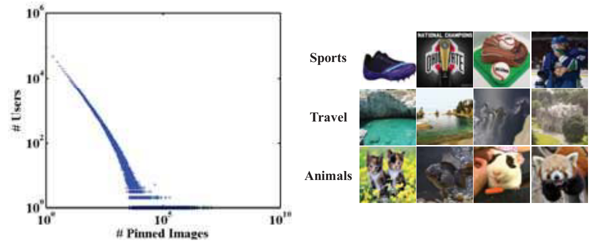
Xue Geng Hanwang Zhang Jingwen Bian Tat-Seng Chua
School of Computing, National University of Singapore
snownus@gmail.com hanwangzhang@gmail.com {bian_jingwen, chuats}@comp.nus.edu.sg

Abstract

Good representations of data do help in many machine learning tasks such as recommendation. It is often a great challenge for traditional recommender systems to learn representative features of both users and images in large social networks, in particular, social curation networks, which are characterized as the extremely sparse links between users and images, and the extremely diverse visual contents of images. To address the challenges, we propose a novel deep model which learns the unified feature representations for both users and images. This is done by transforming the heterogeneous user-image networks into homogeneous low-dimensional representations, which facilitate a recommender to trivially recommend images to users by feature similarity. We also develop a fast online algorithm that can be easily scaled up to large networks in an asynchronously parallel way. We conduct extensive experiments on a representative subset of Pinterest, containing 1,456,540 images and 1,000,000 users. Results of image recommendation experiments demonstrate that our feature learning approach significantly outperforms other state-of-the-art recommendation methods.

1. Introduction

The nature of social networks has gradually been shifted from the conventional user-centric networks such as Facebook (*i.e.*, friendship based) and Twitter (*i.e.*, follower-followee based), to content-centric social curation networks such as Pinterest and Delicious. As a new concept different from user-centric social networks, the development of social curation networks originates from users' emerging need —a pure interest-based social service to explore and collect interesting contents (such as images, videos and topics), through which to communicate with people (mostly strangers) who share similar interest. In addition to creating contents, social curation encourages users to involve in an information filtering process to identify, collect and aggregate images into their own stories [33]. Today, over 73% of social media marketing efforts exploit social curation as sources of marketing contents [27]. However, the traditional recommender systems are not designed to function effectively in this new era of social curation marketing due to the following challenges: 1) The *extreme sparsity* of network structure (cf. Figure 1(a)). For instance, in Pinterest, an ordinary user often curates around 100 images which is only *one in a million* as compared to the whole Pinterest image collection. That is to say, it is hardly possible to infer the similarity between users based on the shared images. Clearly, this will render collaborative filtering ineffective. 2) The *extreme diversity* of the multimedia contents (cf. Figure 1(b)). Different from products that can be easily categorized (such as those in Amazon), the categories of multimedia contents are usually hard to be identified automatically, causing difficulties for content-based recommender systems to infer accurate user interest from the curated contents, with the problem of over-specification [1]. In this paper, we introduce a novel feature learning approach for recommendation that aims to tackle the above two extreme challenges in social curation. Different from conventional recommenders that indirectly rank images for users, we directly measure the similarity between users and images through a compact, low-dimensional vector space, spanned by "interest", which is the core motive of any social curation network. Our algorithm takes a social curation



(a) (b)

Figure 1. Illustrations of the two extremes in a typical social curation network, *e.g.*, Pinterest. (a) The power-law distribution of the number of images pinned by users. It means that the user-image connections are long-tailed and very sparse. (b) Some exemplar images of three interest categories. The contents of images in the same category are very diverse.

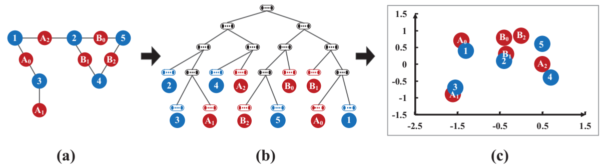


Figure 2. Our goal is to transform the users and images in a social curation network into a compact, low-dimensional feature space. Our approach takes user-image pairs in the social curation network (a) as input to the proposed deep architecture (b) which is built based on frequencies of user-image interactions. It then learns the user and image feature representations (c) as the output. Here is a simple illustration of our proposed method on a toy image-centric network: blue ones are users and red ones are images. We can see that the learned features can capture the pairwise user-image similarity.

network with user-image links as input and produces latent representations of users and images as output. As illustrated by a toy network with 5 users and 6 images in Figure 2, we expect the vectors of linked users and images to be closer than other non-linked ones. The closer the pair of vectors, the higher the possibility that the user-image pair belongs to the same interest, and hence the rank of the image with respect to the user is higher.

Our model is a novel deep learning framework that breaks down a large and sparse network topology into a tree-structured deep hierarchy, where the leafs are users and images (Figure 2). Each non-leaf feature encodes the information about the social interactions (*i.e.*, user-image, user-user, and image-image) and each resultant leaf embeds the "interest" of a user or an image into a vector. Note that our deep model is used as an "end-to-end" fashion, that is, we start from the most basic curation behavior "a user likes or dislikes an image" as the "low-level end", and the latent features forwardly propagate the curation belief into the resultant user-image features as the "high-level end". Different from shallow methods that attempts to learn user and image features directly [12, 15], our deep model can compactly [3] and efficiently learned representative features to reveal the weak correlations between images and users at the scene of the extreme sparse connections and extreme diverse images due to its deep structure.

In our proposed deep model, the input of user-image pairs could be over billions. Thus, how to efficiently optimize such a deep model becomes a big challenge. Fortunately, we observe that the user-image connections are long-tailed and very sparse, and hence there should be very limited shared parameters for different user-image pairs in the proposed deep tree structure. Therefore, we proposed a fast optimization algorithm that deploys an asynchronously parallel stochastic gradient descent method that can significantly reduce the time for the training of different user-image pairs.

We conduct extensive experiments on a representative subset of Pinterest, which is the most popular social curation network. In particular, the subset covers 468 popular interests on Pinterest with 1,456,540 images and 1,000,000 users who have interactions with these images. Through image recommendations, we demonstrate that the proposed deep model significantly outperforms the other state-of-the-art recommender systems. Our contributions are summarized as follows:

- We propose a deep learning framework for learning compact user and image features in a unified space from large, sparse and diverse social curation networks. The learnt user and image features support effective recommendation by directly computing the similarity between the user vector and image vector.

- We develop a fast on-line algorithm to train the proposed deep learning framework.

- To our best knowledge, this is the first work on developing deep learning methods on content-centric networks. Extensive experimental results have demonstrated the proposed deep model significantly outperforms other benchmark recommendation methods.

2. Related Work

Feature learning plays an important role in computer vision. From low-level hand-crafted features (such as HOG [6] and SIFT [17]), to current high-level features (such as DCNN [16]), image contents have been sufficiently analyzed to some degree from shallow architectures to deep architectures. In addition to image contents, human engagement such as data collection [8], annotation and knowledge extraction is also curial for advancing computer vision. Specifically, the collective intelligence of social media can competitively enhance computer vision [4]. For example, [11] use image relationship discovered from user behavioral data to guide image feature learning.

Recommendation is a classic problem in Artificial Intelligence. It is beyond the scope of this paper to do a comprehensive review [1, 28]. In general, recommendation methods can be classified into content-based filtering (CBF) methods and collaborative filtering (CF) methods. CBF recommends images based on a comparison between the contents of the images and a user profile [2, 26]. User profiles can be identified by the users themselves, or learned from the content of the images that users have rated. The main problem of content-based filtering is the over specification [1]. That is, when a user only rates a limited number of images, the limited content information cannot be generalized to discover the user's broader interest. CF [28] recommends images to users based on the images shared by other users with similar interest. The similarity between users are often computed based on the overlap of shared images. However, CF cannot achieve satisfactory performance when the network is very sparse. To alleviate the sparsity problem, matrix factorization based CF models have been proposed, such as the Singular Value Decomposition (SVD) [25], Weighted Matrix Factorization (WMF) [13], and the combination of probabilistic matrix factorization (PMF) [19] and topic models [31]. These models assume that the user-image matrix has a low-rank reconstruction by low-dimensional user and image features. We argue that such methods are essentially "shallow" models since they directly seek the resultant high-level features from user-image matrix. When the matrix is very sparse, these methods will fail to find meaningful latent factors.

Our work is related to the recent music recommendation work [30]. We share similar ideas in injecting contents (music and images) using deep models (*i.e.*, DCNN) into social latent vectors. But, they used traditional matrix factorization which is not as powerful as our deep model for social networks. Besides, our deep model is related to DeepWalk proposed by Perozzi *et al.* [23], in terms of similar formulations. They learned latent representations for network vertices by modeling a stream of short random walks. The underlying reasoning is that they empirically observed that the short random walks are similar with word distributions and thus their formulation is identical to the one used in language modeling [18]. However, our formulation is derived from the nature of social network: modularity [21], which is theoretically sound. Moreover, our model can handle heterogenous network of images and users.

3. Problem Definition

3.1. Recommendation by Similarity

We consider the problem of recommending images denoted as $\mathcal{I}$ or users denoted as $\mathcal{U}$ to users in a social curation network denoted as $\mathcal{G} = \{\mathcal{U}, \mathcal{I}, \mathcal{E}\}$, where $\mathcal{E}$ is the set of edges that connect users and images. Although real-world social curation networks allow users to connect to other users[1], without loss of generality, we only assume that connections exist between users and images, *i.e.*, $\mathcal{E} \subseteq \mathcal{U} \times \mathcal{I}$. We are interested in the following user-image similarity measure:

$$s_{ui} = \mathbf{x}_u^T \mathbf{x}_i \tag{1}$$

where $s_{ui} \in \mathbb{R}$ is the rating score of image i being recommended to user u, $\mathbf{x}_u \in \mathbb{R}^d$ and $\mathbf{x}_i \in \mathbb{R}^d$ are the latent feature representations for users and images. In order to make a valid recommendation score by Eq. (1), we require $\mathbf{x}_u$ and $\mathbf{x}_i$ to encode interests. For example, if user u likes traveling and image i is about traveling, we expect the values of $\mathbf{x}_u$ and $\mathbf{x}_i$ to be consistently small.

In general, we seek a transformation $g : \mathcal{G} \mapsto \mathbb{R}^d$, where $\mathbb{R}^d$ is the unified space for users and images and thus facilitates direct user-image similarity measure in Eq. (1). Note that the transformation is generic since content-based filtering and collaborative filtering can be viewed in this form. For content-based filtering, it considers $\mathbf{x}_u$ as a content feature generated from the user's favored images. On the other hand, collaborative filtering treats $\mathbf{x}_i$ as the vector consisting of ratings $r_{u'i}$, where u' is a friend of u, and $\mathbf{x}_u$ is a vector of the similarities between the friends of u. As discussed in Section 1, the extreme connection sparsity and content diversity will make these traditional methods ineffective. For example, in content-based filtering, even if a user only likes a single interest "travel", it is difficult to generate $\mathbf{x}_u$ that is consistently similar to diverse images about traveling; in collaborative filtering, as the user-image connections are very sparse, it is impossible to infer accurate user similarities based on the shared images between users.

3.2. Modularity

Due to the sparsity of social networks, we wish to seek low-dimensional features for items (*i.e.*, images) and users, through an objective that represents the interest communities of social networks. Modularity is a widely-used community partition measure that the larger the value, the better the partition of the network [5]. The underlying principle of using modularity is that the power-law distribution of connections between users and items is very significant in social curation network[2]. Consider the partitioning network $\mathcal{G}$ of n vertices (*e.g.*, $n = |\mathcal{U}| + |\mathcal{I}|$) and m edges into k non-overlapping interest communities. Let d_i represents the degree of vertex i. Modularity penalizes the situations when the number of within-group connections is smaller than the number of uniformly random connections, whose expected number is $d_i d_j / 2m$. Therefore, the modularity is formulated as:

$$J = \frac{1}{2m} \sum_{ij} \left(\mathbf{G}_{ij} - \frac{d_i d_j}{2m} \right) \delta\left(i, j\right), \tag{2}$$

where $G_{ij} = 1$ if i and j is connected and 0 otherwise, $\delta(i,j) = 1$ if i and j belong to the same membership and 0 otherwise. Note that $0 \leq d_i d_j / 2m \leq 1$, so the penalty comes in if $\left(\mathbf{G}_{ij} - \frac{d_i d_j}{2m} \right) < 0$. One aims to find a community partition over the network $\mathcal{G}$ when J is maximized. Note that we make no difference between users and items since our goal is to learn a unified interest space.

Although maximizing the modularity J over hard partition (*i.e.*, $\sigma(i,j) = 1$ or 0) is NP-hard [5], a relaxed approximation of the problem can be solved efficiently [29] when we relax the membership indicator $\sigma(i,j) = p(i|j) =$

[1] This rarely happens because most users only enjoy the curation function and ignore the social function.

[2] The fraction of nodes in the network have k connections to other nodes is proportional to $k^{-\gamma}$.

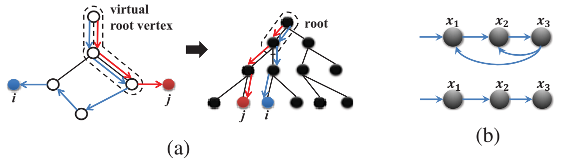


Figure 3. Illustrations of the proposed deep architecture for social network. (a) The node parameters of two paths in the deep hierarchy encode the topology information of a random walk from a virtual root to vertices. For example, the shared parameters correspond to the overlaps of the two routes (in dashed region). (b) Traditional deep architecture (bottom)s feed a fixed input into a forward network, while the proposed model (top) feeds both the output image and user features as input to every forwarding layer.

$exp\left(\mathbf{x}_i^T\mathbf{x}_j\right)/\sum_{i'} exp\left(\mathbf{x}_{i'}^T\mathbf{x}_j\right)$ as a valid probability: where $\mathbf{x}_i \in \mathbb{R}^d$ is a latent membership feature vector and the probability function is known as the softmax function. One can easily derive that this relaxed formulation is strongly related to the formulation of matrix factorization for recommendation [12, 15], which usually fails in sparse social network as we argued in Section 1.

4. Deep Learning Features for Social Networks

In general, the latent interests encoded in the topology is difficult to be revealed by using these shallow methods when we directly solving Eq. (2). This is analogous to the situation in image classification, which suffers from the gap between noisy visual cues and the target labels. For this task, it is well-known that DCNN performs the best because they learn hierarchical features which are beneficial for the ultimate classification [3, 16]. Inherited from this core spirit of deep learning, we propose to solve Eq. (2) by a hierarchical deep model, which can learn useful intermediate features.

4.1. Architecture

We start from introducing an approximation of $p(i|j)$ called "Hierarchical Softmax", which is widely used in neural computation [20]. It approximates $p\,(i|j)$ by a series of binomial distributions along a tree-structured hierarchy. Specifically, we assign the vertices to the leafs of a binary tree (see Figure 2). For computation efficiency, the tree is a Huffman tree [18] according to the frequency of user-image interactions. Let $n_i(m)$ be the m-th node on the path from root to i, and let L_i be the length of this path. In particular, we have $n_i(1)$ as root and $n_i(L_i)$ as i. In addition, we denote $lc(n_i(m))$ as the left child of node $n_i(m)$ and let $I(n_i(m))$ be an indicator function such that it is 1 if $n_i(m+1) = lc(n_i(m))$, and -1 otherwise. Then, the hierarchical softmax version of $p\,(i|j)$ is defined as

$$p\,(i|j) = \prod_{m=1}^{L_i-1} \sigma\left(I[n_i(m)] \cdot \mathbf{x}_{n_i(m)}^T \mathbf{x}_j\right) \tag{3}$$

where $\sigma(x) = 1/(1+exp(-x))$ is the sigmoid function, which is widely-used to model the binary-valued binomial probability and $\mathbf{x}_{n_i(m)}$ is the representations of inner node $n_i(m)$. In terms of computation complexity, Eq. (3) can reduce the computation complexity of n sums (where $\mathcal{O}(n)$ can be millions in our case) with normalization in Eq. (1) to $\mathcal{O}(\log_2 n)$, which is significant.

Here, we show that the hierarchical softmax as formulated in Eq. (3) can be viewed as a deep architecture that represents the network topology. First, we can view the binary tree as a coding structure for each vertex in the network because each vertex i is assigned to a path from root to leaf. Then, the series of binary decisions from root to bottom mimic the route in the network from a common virtual root to vertex i. As shown in Figure 3(a), the route to the vertex i is by way of vertex j. The shared nodes along the path of j to i encode this routing information. So, we can view the nodes in the hierarchy encodeing the topology of the entire network. Finally, we illustrate that Eq. (3) is in fact a forward propagation in the deep model. As illustrated in Figure 3(b), the difference between a traditional deep neural network and our network is that the proposed deep model is forwarded by using both the output features (*i.e.*, the leaf vectors) and the hidden units, while traditional neural network is forwarded by using only the hidden units. Detailed information can be seen in Equation 5.

4.2. Formulation

We are interested in recommending image i to user u (or user u to image i). Intuitively, our learning objective seeks for feature representations $\mathbf{x}_u$ and $\mathbf{x}_i$ such that

$$\begin{cases} \max\limits_{\mathbf{x}_u,\mathbf{x}_i} \;\; p(u|i) \;\text{ or }\; p(i|u), \;\text{ if } \; u \; likes \; i, \\ \min\limits_{\mathbf{x}_u,\mathbf{x}_i} \;\; p(u|i) \;\text{ or }\; p(i|u), \;\text{ if } \; u \; dislikes \; i. \end{cases} \tag{4}$$

Note that the above objective is consistent with the modularity maximization in Eq. (2). Moreover, we deploy a DCNN to transform images into the desired feature space: $\mathbf{x}_i = \text{CNN}(i)$, in order to generalize for new images. In this paper, we adopt the AlexNet [16] where the softmax layer is removed but an additional fully-connected layer is added (*i.e.*, from 4,096 to d neurons).

By incorporating the $p(u|i)$ formulated as in Eq. (3) into Eq. (2), the overall objective function becomes

$$\begin{aligned} &\max_{\mathbf{x}_{n_u(m)},\mathbf{x}_{n_i(m)},\mathbf{x}_u,\text{CNN}(\cdot)} \\ J = &\sum_{ui} A_{ui} \sum_{m=1}^{L_u-1} \log\sigma\left(I[n_u(m)]\cdot \mathbf{x}_{n_u(m)}^T \text{CNN}(i)\right) \\ &+ \sum_{iu} A_{iu} \sum_{m=1}^{L_i-1} \log\sigma\left(I[n_i(m)]\cdot \mathbf{x}_{n_i(m)}^T \mathbf{x}_u\right) \end{aligned} \tag{5}$$

where $A_{ui} = (G_{ui} - d_u d_i/2m)$. Note that the above formulation allows us to encourage $p(u|i)$ to be larger if $A_{ui} \geq 0$ and smaller if $A_{ui} < 0$. Recall that $0 \leq d_i d_j/2m \leq 1$, so $A_{ui} \geq 0$ indicates user u likes image i and vice versa. Also, A_{ui} assigns a weight to encourage the connection G_{ui} if the expected connection $d_u d_i/2m$ is small. For example, if user u is only linked to few images (*i.e.*, small d_u) and image i is only linked to few users (*i.e.*, small d_i), then an observation of u linked to i is informative. Therefore, the likelihood for $p(u|i)$ or $p(i|u)$ should be emphasized in optimization. For $A_{ui} < 0$, we only compute the pairs with the smallest 20 values for efficiency. Note that one can try more advanced negative sampling tricks [30], however, we found that there is no significant improvement.

4.3. Algorithm

For a typical social curation network, the number of user-image pair could be over billions. Therefore, it is impractical to optimize Eq. (5) even if we use the popular online stochastic gradient descent method for deep learning [3]. Here, we design a fast algorithm for tackling the large-scale networks. The main idea of our algorithm is that we deploy an asynchronously paralleled stochastic gradient descent method that can significantly reduce the time of scanning the user-image pairs.

The parallelization is made possible by the two observations from the structure of the topology parameters $\mathbf{x}_{n_u(m)}$ and $\mathbf{x}_{n_i(m)}$. First, as shown in Figure 1(a), the frequency distributions of users and images follow the power-law distribution. This observation is generally true in most social networks [21]. It means that we have a very long tail of infrequent pairs and thus the chance of two computing threads conflict when scanning the same pair is rare. Second, thanks to the binary tree structure of the parameters, the number of shared parameters between two leafs are limited. To see this, suppose that u and i correspond to sibling leafs, which is the worst case. The number of shared parameters is only $\log_2 n - 1$, where n is the total number of users and images. When $n = 10^7$, the fraction of affected parameters is only around 0.00002%, which is negligible.

However, the parameters of CNN is shared by all the pairs. Therefore, jointly optimizing all the parameters in Eq. (5) will harm parallelization. To tackle this, we propose an alternative updating algorithm as shown in Algorithm (1). Specifically, we first fix the features of users $\mathcal{X}$ and CNN, and only update the topology parameters (*i.e.*, the inner node features) $\mathcal{T}$ as in Algorithm 2. Note that Steps 2-11 can be run asynchronously with multiple threads. In general, Algorithm 2 requires about 100 iterations for convergence. Next, as shown in Algorithm 3, we solve for $\mathcal{X}$ and CNN with fixed $\mathcal{T}$. It should be noted that $\mathcal{X}$ and CNN in Eq. (5) can be updated independently. In particular, they can be trained by asynchronous stochastic gradient descent on a distributed computing platform as described in [7]. We employ the momentum-based gradient descent as Steps 6-7

Algorithm 1: Deep Feature Learning for Images and Users

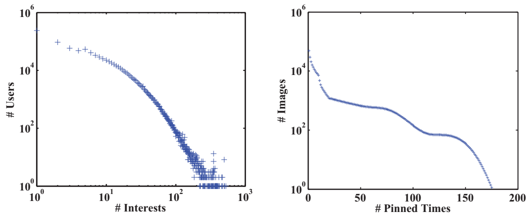
```
   Input: Social curation network $\mathcal{G}$, feature dimension $d$
   Output: User features $\mathbf{x}_u$ and image visual feature
           transformation CNN
1  Initialization: Build a binary tree for the users and images
   in $\mathcal{G}$; randomly set topology parameters $\mathbf{x}_{n_u(m)}$ or
   $\mathbf{x}_{n_i(m)} \in \mathcal{T}^{(0)}$, and user feature $\mathbf{x}_u \in \mathcal{X}^{(0)}$, initialize CNN
   with ImageNet pretrained model; randomly initialize the last
   layer of CNN, $t \leftarrow 0$
2  repeat
3      $\mathcal{T}^{(t+1)} \leftarrow \text{UpdateTopology}\left(\mathcal{T}^{(t)}, \mathcal{X}^{(t)}, \mathbf{W}^{(t)}\right)$
4      $\mathcal{X}^{(t+1)}, \text{CNN}^{(t+1)} \leftarrow \text{UpdateFeature}\left(\mathcal{T}^{(t+1)}\right)$
5      $t \leftarrow t + 1$
6  until converges;
```

Algorithm 2: UpdateTopology $\left(\mathcal{T}^{(0)}, \mathcal{X}, \text{CNN}\right)$

```
1  Initialization: $t \leftarrow 0$, momentum $\Delta^{(0)} \leftarrow 0$, weight-decay
   factor $\alpha$, learning rate $\eta$
2  repeat
3      Online gradient descent:
4      foreach pair of $u$ and $i$ do
5          foreach $\mathbf{x} \in \mathcal{T}$ do
6              $\Delta^{(t+1)} = 0.9\Delta^{(t)} - \alpha \cdot \eta \cdot \mathbf{x}^{(t)} - \eta \nabla_{\mathbf{x}} J(\mathcal{T}^{(t)})$,
7              $\mathbf{x}^{(t+1)} = \Delta^{(t+1)} + \mathbf{x}^{(t)}$,
8          end
9      end
10     $t \leftarrow t + 1$
11 until converges;
12 return $\mathcal{T}^{(t)}$
```

Algorithm 3: UpdateFeature $(\mathcal{T})$

```
1  Initialization: $t \leftarrow 0$, momentum $\Delta^{(0)} \leftarrow 0$, weight-decay
   factor $\alpha$, learning rate $\eta$
2  repeat
3      Stochastic Gradient descent:
4      foreach randomly selected mini-batch of user-image
       pairs do
5          $\Delta^{(t+1)} = 0.9\Delta^{(t)} - \alpha \cdot \eta \cdot \left(\mathcal{X}^{(t)}, \text{CNN}^{(t)}\right) -$
           $\eta \nabla_{(\mathcal{X},\text{CNN})} J\left(\mathcal{X}^{(t)}, \text{CNN}^{(t)}\right)$,
6          $\left(\mathcal{X}^{(t+1)}, \text{CNN}^{(t+1)}\right) = \Delta^{(t+1)} + \left(\mathcal{X}^{(t)}, \text{CNN}^{(t)}\right)$,
7      end
8      $t \leftarrow t + 1$
9  until converges;
10 return $\left(\mathcal{X}^{(t)}, CNN^{(t)}\right)$
```



(a) (b)

Figure 4. Dataset statistics. (a) This shows the number of users' interests; and (b) this shows the distribution of the times an image has been pinned.

in Algorithm 2 and Steps 5-6 in Algorithm 3. This method has been shown to result in faster learning paces [24].

5. Experiment

In this section, we conduct extensive recommendation experiments to evaluate the effectiveness of the learnt user and image features from the proposed deep model.

5.1. Dataset

We used *Pinterest*, which is one of the largest social curation networks, as the source of the content-centric network for evaluating our proposed methods. To our best knowledge, there is no publicly available social media dataset that is large scale and image-centric with ground-truths of images. In Pinterest, users "pin" images to their own boards, showing their preferences of these images. In this research, we only crawled images with additional information indicating their categories[3] (*e.g.*, Fishing, Travel, and Hockey). We used the image categories as the groundtruth of user interests. In particular, given a user and his/her pinned images, we first found the category labels of these images and used these labels as the interests of this user. We crawled the profiles of 1 million users together with their pinned images from Pinterest. The users were randomly sampled from the users communities found in the 468 categories we analyzed. For the pinned images, we removed images without category labels, resulting in 686,457 images. We named this set of images I_u, those that actually pinned by users. In order to test the ability of recommending new images not pinned by users, we also crawled additional 770,083 images which belong to the 468 interest categories but not pinned by any of the crawled users. The new image set is named as I_{new}. In the process, we also removed duplicated images which may impact the final evaluation results. These images were used to evaluate the performance of new image recommendation.

Figure 4 and Figure 1(a) show three distributions of our dataset: the distribution of the number of users' interests, the distribution of the times an image has been pinned, and the distribution of the number of users' pinned images. These distributions are power-law, where most users pin only a small number of images and have only a few interests; similarly, the images are only pinned by a very small number of users as compared to the total number of users. These distributions showed the sparsity and diversity of a typical social curation network. In order to demonstrate that our method can perform consistently well on different network topology, we randomly divided our dataset into 10 groups, each of which contains $100,000$ users and around $1,000,000$ images. The set of images includes those images pinned by the users in the group, with remaining randomly sampled from I_{new} set. The experiments were conducted on all the 10 groups. We reported averaged results with significance tests (applying t-test) and published the dataset [1].

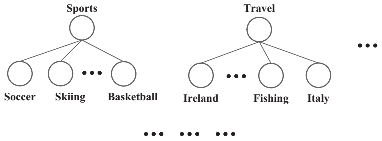


Figure 5. Interest categories in Pinterest are organized as a forest.

5.2. Implementation Details

For deep CNN, we depoyed Caffe framework [14] for CNN implementation on a NVIDIA Titan Z GPU. In particular, we used the well-known AlexNet architecture [16], which consists of 5 convolutional layers with max-pooling and 2 fully connected layers before the loss layer. Our CNN added an additional fully connected layer for the resultant d-dimensional feature space. We used the author provided ImageNet pretrained model (in Caffe format) as initializations. The initial learning rate was set to $1e^{-4}$ with dynamic momentum. The size of the batch was 128 and it took 20 epochs to converge using Algorithm 3. Each epoch took about 40 mins. For Algorithm 2, we randomly initialized all the parameters, and the starting learning rate was set to $1e^{-5}$ with dynamic momentum. We used 8 computing threads on a 8-core machine. It took around 100 epochs to converge with each epoch taking about 10 mins. For the above algorithms, we used ℓ_2-norm weight decay with $5e^{-5}$ coefficient. For Algorithm 1, we found that 2 iterations were sufficient for a good solution. The choice of feature dimension is crucial. We tuned the values within $\{100, 200, ..., 1,000\}$ and found that 300 was the best choice.

5.3. Evaluation Metrics

We evaluated our method and other compared ones on image recommendation. We adopted the widely-used Normalized Discounted Cumulative Gain (NDCG) as the eval-

[3] http://www.pinterest.com/categories/

[1] https://sites.google.com/site/xueatalphabeta/academic-projects

uation metric for both tasks. NDCG is defined as:

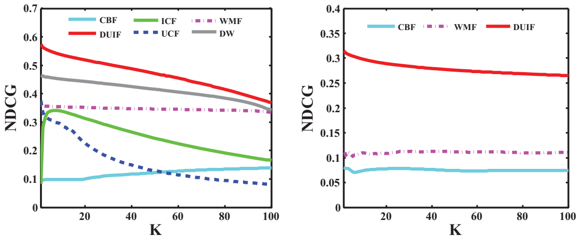
$$NDCG_k = \frac{1}{IDCG_k} \times \sum_{i=1}^{k} \frac{2^{r_i - 1}}{log_2(i+1)} \qquad (6)$$

where $IDCG_k$ is the maximum $NDCG_k$ that corresponds to the optimal ranking list so that the perfect $NDCG_k$ is 1, and r_i is the degree of relevance of the image in position i. We adopted a 3-scale $r \in \{0, 1, 2\}$ relevance score, representing *irrelevant*, *relevant*, and *highly relevant*, respectively. For image recommendation, we defined a recommended image to be: (a) highly relevant if the interest category of the image falls within the groundtruth interests of users; (b) relevant if the interest category of the image maps to sibling interests of users' groundtruth interests (Figure 5 illustrates a part of the interest category forest collected from Pinterest); and (c) irrelevant if none of the above.

In addition to NDCG which measures the relevance of the recommended images, users may also prefer the recommended images to be more diverse, *i.e.*, if a user has many interests, results that cover more interests are preferred. Therefore, we used entropy $H_k = -\sum_{i=1}^{R} p_i \ln p_i$ to measure the diversity of the recommendation results, where S_k is the set of successfully recommended (highly relevant and relevant) images up to position k, R is the total number of types of interests in S_k, and p_i is the proportion of images belonging to the ith type of interest in S_k. Here, a larger H_k represents more diverse results.

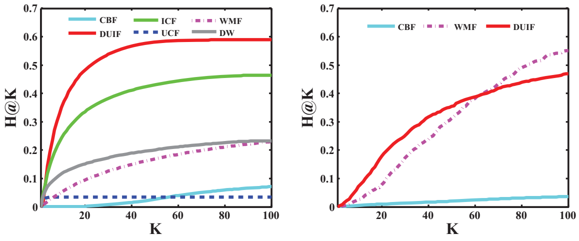
5.4. Comparing Methods

We compared the performance of our proposed Deep User-Image Feature (**DUIF**) with the following five baseline methods: a) Content-based filtering (**CBF**) [22, 28]: It generates a user feature vector by averaging all the image features (we used the state-of-the-art 4,096-d DeCAF [10] feature) pinned by the user and then recommend images based on the similarity between the image features and the user features. b) User-based collaborative filtering (**UCF**) [32]: It analyzes the user-image matrix to compute the similarities between users and then recommends images to people with similar tastes and preference. c) Item-based collaborative filtering (**ICF**) [9]: This technique first analyzes the user-image matrix to identify relationships between different images, and uses these relationships to indirectly compute recommendations for users. d) Weighted Matrix Factorization (**WMF**) [30]: It decomposes the user-image matrix into latent user and image features by the weighted matrix factorization [13] and uses CNN to regress images to the image vectors. e) Deep Walk (**DW**) [23]: It learns the user and image latent representations of vertices in a social network by applying a language model. Then, images are recommended by the similarity between the user features and the image features. We empirically tested different configurations of baseline methods and employed the best ones as baselines.



(a) (b)

Figure 6. Performances ($NDCG_k$) of various methods on recommending new images to users based on (a) existing pinned set (I_u) and (b) new image set I_{new}.



(a) (b)

Figure 7. Performances of diversity (H_k) of various methods on recommending new images to users based on (a) existing pinned set (I_u) and (b) new image set (I_{new}).

5.5. Results and Analysis

For our evaluation, we want to test the effectiveness of the recommendation methods to recommend new images based on those pinned by existing user community I_u and those unseen images I_{new} not pinned by existing user community. We note that among the five baseline methods, CBF is based on the contents of the images, UCF and ICF are traditional collaborative filtering methods, while WMF and DW are based on latent factors. We note that UCF, ICF and DW cannot be used to recommend new images, which are unseen in existing networks. Hence for testing recommending new unpinned images from set I_{new}, we only compare our proposed method with CBF and WMF.

Figure 6 and 7 compare the performance of recommendation methods to recommend relevant images to users based on existing pinned set I_u and new image set I_{new}. Figure 6 presents the performance in terms of relevance based on $NDCG@K$; while Figure 7 presents the performance in terms of diversity based on $H@K$. In addition, Table 1 and Table 2 separately lists the respective results with significant test on image recommendation at the top 5, 10, 20, 50 and 100 positions. Some illustrative examples are shown in Figure 8.

As can be seen from the results, the proposed DUIF

Table 1. Detailed recommendation performance ($NDCG_k$) on recommending new images to users based on existing pinned set (I_u) and new image set (I_{new}) with significance test. Results labeled with ‡ are highly significant (p<0.01), and † indicates significant (p<0.05), against the best comparing method.

	Existing Image Recommendation				
	$NDCG_5$	$NDCG_{10}$	$NDCG_{20}$	$NDCG_{50}$	$NDCG_{100}$
CBF	0.098	0.099	0.100	0.122	0.139
UCF	0.308	0.290	0.226	0.129	0.081
ICF	0.338	0.338	0.314	0.244	0.165
WMF	0.356	0.354	0.352	0.346	0.334
DW	0.457	0.451	0.443	0.416	0.342
DUIF	**0.550**‡	**0.537**‡	**0.519**‡	**0.472**‡	**0.368**†
	New Image Recommendation				
	$NDCG_5$	$NDCG_{10}$	$NDCG_{20}$	$NDCG_{50}$	$NDCG_{100}$
CBF	0.079	0.080	0.081	0.080	0.081
WMF	0.103	0.110	0.108	0.111	0.110
DUIF	**0.304**‡	**0.298**‡	**0.289**‡	**0.276**‡	**0.265**‡

Table 2. Detailed recommendation performance (H_k) on recommending new images to users based on existing pinned set (I_u) and new image set (I_{new}) with significance test. Results labeled with ‡ are highly significant (p<0.01), and † indicates significant (p<0.05), against the best comparing method.

	Existing Image Recommendation				
	H_5	H_{10}	H_{20}	H_{50}	H_{100}
CBF	0.000	0.000	0.002	0.027	0.071
UCF	0.034	0.035	0.035	0.035	0.035
ICF	0.147	0.243	0.335	0.430	0.465
WMF	0.025	0.052	0.095	0.169	0.230
DW	0.082	0.117	0.152	0.201	0.233
DUIF	**0.194**‡	**0.350**‡	**0.481**‡	**0.581**‡	**0.589**†
	New Image Recommendation				
	H_5	H_{10}	H_{20}	H_{50}	H_{100}
CBF	0.002	0.005	0.010	0.020	0.037
WMF	0.005	0.025	0.078	0.312	0.551
DUIF	**0.022**‡	**0.071**‡	**0.180**‡	**0.354**‡	**0.470**

significantly outperforms the other methods for image recommendation. The comparatively good performance of DUIF mainly comes from the following aspects. As previously introduced, the multimedia contents are very diverse, even for the same interest topic, hence methods (*e.g.*, CBF) that only consider image contents would have poor performance. Moreover, each user often has many different interests. Such diverse images and varying users would result in a more sparse and complex user-item matrix, which renders those matrix decomposition based methods such as UCF and WMF ineffective in revealing the underlying user interests. Further, we observe that the latent factor based models such as WMF often outperforms the traditional collaborative filtering methods such as UCF and ICF. The findings verified that methods that attempt to discover compact latent vectors for users and images tend to perform better than those that directly apply the user-image matrix. Finally, although DW which is similar to DUIF, it does not consider the contents of images and the intrinsic property of social curation network, namely modularity. Hence it performs worse than DUIF on the recommendation task. Overall, DUIF differs from the baseline methods in that it jointly considers image content analysis and social curation network topology. Experimental results have shown that it can effectively map images and users into a unified space for effective image recommendation.

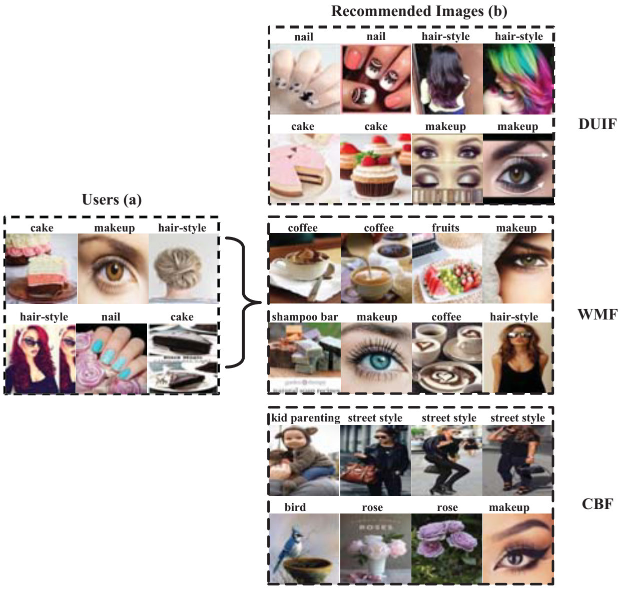


Figure 8. Illustrative examples of recommending new images to users using different methods (b) based on users' pinning profiles (a).

6. Conclusion

We proposed a novel deep learning framework for learning the representations for topological user nodes and visual images in large, very sparse and diverse social curation network and applied the resulting model to recommender system. Experimental results on a representative subset of Pinterest with about 1.4 million images and 1 million users have demonstrated that the proposed approach can significantly outperform other methods. Exploiting social media data to generate features could be a promising research direction in computer vision community.

References

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *TKDE*, 2005. 1, 2

[2] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 1997. 2

[3] Y. Bengio. Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2009. 2, 4, 5

[4] E. Chatzilari, S. Nikolopoulos, I. Patras, and I. Kompatsiaris. Enhancing computer vision using the collective intelligence

of social media. In *New Directions in Web Data Management 1*. Springer, 2011. 2

[5] A. Clauset, M. E. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 2004. 3

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[7] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, et al. Large scale distributed deep networks. In *NIPS*, 2012. 5

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[9] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *TOIS*, 2004. 7

[10] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014. 7

[11] C. Fang, H. Jin, J. Yang, and Z. Lin. Collaborative feature learning from social media. In *CVPR*, 2015. 2

[12] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 2001. 2, 4

[13] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, 2008. 3, 7

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 2014. 6

[15] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009. 2, 4

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2, 4, 6

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 2004. 2

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 2013. 3, 4

[19] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, 2007. 3

[20] F. Morin and Y. Bengio. Hierarchical probabilistic neural network language model. In *AISTATS*, 2005. 4

[21] M. E. Newman. Modularity and community structure in networks. *PNAS*, 2006. 3, 5

[22] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The adaptive web*. Springer, 2007. 7

[23] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *SIGKDD*, 2014. 3, 7

[24] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 1999. 6

[25] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system-a case study. Technical report, DTIC Document, 2000. 3

[26] M. Saveski and A. Mantrach. Item cold-start recommendations: learning local collective embeddings. In *RecSys*, 2014. 2

[27] M. A. Stelzner. *2014 Social Media Marketing Industry Report: How Marketers are Using Social Media to Grow Their Businesses*. Social Media Examiner, 2014. 1

[28] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009. 2, 7

[29] L. Tang and H. Liu. Relational learning via latent social dimensions. In *SIGKDD*, 2009. 3

[30] A. Van den Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In *NIPS*, 2013. 3, 5, 7

[31] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *SIGKDD*, 2011. 3

[32] Z.-D. Zhao and M.-S. Shang. User-based collaborative-filtering recommendation algorithms on hadoop. In *WKDD'10*, 2010. 7

[33] C. Zhong, S. Shah, K. Sundaravadivelan, and N. Sastry. Sharing the loves: Understanding the how and why of online content curation. In *ICWSM*, 2013. 1