Personalized Tag Recommendation for Images Using Deep Transfer Learning

Hanh T. H. Nguyen(✉), Martin Wistuba, and Lars Schmidt-Thieme

Information Systems and Machine Learning Lab, University of Hildesheim,
Universitätsplatz 1, 31141 Hildesheim, Germany
{nthhanh,wistuba,schmidt-thieme}@ismll.de

Abstract. Image tag recommendation in social media systems provides the users with personalized tag suggestions which facilitate the users' tagging task and enable automatic organization and many image retrieval tasks. Factorization models are a widely used approach for personalized tag recommendation and achieve good results. These methods rely on the user's tagging preferences only and ignore the contents of the image. However, it is obvious that especially the contents of the image, such as the objects appearing in the image, colors, shapes or other visual aspects, strongly influence the user's tagging decisions.

We present a personalized content-aware image tag recommendation approach that combines both historical tagging information and image-based features in a factorization model. Employing transfer learning, we apply state of the art deep learning image classification and object detection techniques to extract powerful features from the images. Both, image information and tagging history, are fed to an adaptive factorization model to recommend tags. Empirically, we can demonstrate that the visual and object-based features can improve the performance up to 1.5% over the state of the art.

Keywords: Image tagging · Convolutional neural networks
Personalized tag recommendation · Factorization models

1 Introduction

A large number of digital resources are stored, shared and accessed by users around the world everyday. To assist the organization and retrieval of images, social media services allow users to annotate their resources with their own keywords, called tags. Even though tagging is a relatively simple task, it is tedious, time-consuming and thus discourages the users from tagging their images. A study by Sigurbjörnsson and Van Zwol revealed that most images uploaded to Flickr have only few or even no tags [17]. They analyzed photos uploaded between February 2004 and June 2007 and reported that around 64% of them have 1 to 3 tags and around 20% have no tags at all.

Tag recommendation is used to save the user's time by suggesting relevant tags for the uploaded content. These suggestions are preferably based on the

M. Ceci et al. (Eds.): ECML PKDD 2017, Part II, LNAI 10535, pp. 705–720, 2017.
https://doi.org/10.1007/978-3-319-71246-8_43

user's tag preferences and the contents of the uploaded resource. However, in practice the tag recommendation systems are often solely based on the user's tagging history, often ignoring the content of the uploaded items [2,13,16].

One disadvantage of the narrow folksonomy systems, which allow one or few people providing tags for a given resource, is the item cold-start problem. Most images uploaded to platforms such as Flickr are tagged only by few users, i.e. the owner of the image and other users with permissions granted by the owner. Hence, personalized tag recommendation models that are solely based on the user's preferences have tremendous problems providing useful predictions, especially for images that just have been uploaded. Thus, these recommendation models are often predicting the most popular tags.

According to Sigurbjörnsson and Van Zwol [17], people usually choose words related to the contents or contexts such as location or time to annotate images. Image features could be used to solve the cold-start problem. The low-level features such as color histograms have been often used in the personalized content-aware tag recommendation to overcome the problem.

In this paper, we propose a personalized tag recommendation which uses various deep learning methods and publicly available data sets for image classification and object recognition to extract powerful image features. These image features are combined with factorization models in order to boost the prediction performance. We propose to train a convolutional neural network on the famous ImageNet data set which is able to extract useful features from images on our image data set. Furthermore, we are training a convolutional neural network to detect 80 different objects on the MS COCO data set. Both these tasks (classification and object detection) are different to our task (tag recommendation) and use different data sets. However, we will show that we can use these networks to extract useful features from images that will help us recommending better tags. The extracted visual features are finally used by factorization machines (FM) [13] and pairwise interaction tensor factorization (PITF) [16] in order to give final recommendations. Our experiments are conducted on a real world data set, namely NUS-WIDE, and we can show that our proposed way of extracting image features improves the accuracy of the tag recommender by at least 1%.

The motivation for our approach can be explained easily and follows the way how human beings tag images. Lets have a look at Fig. 1. The user tagged this image with "urban", "motorcycle" and "downtown". While the COCO data set only allows us to distinguish 80 different objects which are completely unrelated to our task, we can nevertheless detect a person, a motorbike, a car and few further objects. This is also what a human being does and the appearance of the motorbike likely resulted into the tag "motorcycle". However, object detection can also help to recommend tags such as "urban". This tag is obviously no object which you see on the image but the recommender system can learn that whenever motorbikes, cars and people are detected on an image that an urban area or city is pictured. In a similar way the classification algorithm can extract image features such as specific shapes, colors and so on.

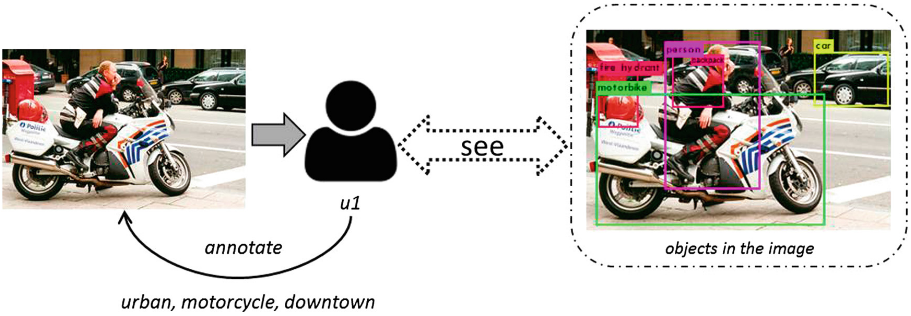


Fig. 1. When tagging an image, the user is highly influenced by what she sees on the image. In this example the user chooses “motorcycle” due to its occurrence. Furthermore, the tag “urban” is chosen since the image shows a street, people, cars and other things typical for cities. Our idea is to create an automatic system that uses object detection as a part of it to improve the tag recommendation performance.

2 Related Work

Tag recommendation can be based on different information such as the user’s tagging behavior, the image contents, the time and location when the image was taken. A large number of approaches have been proposed which target various of these information. Li and Wang [5] extracted color and texture features and learned a mapping between these features and semantic concepts described by several keywords. The recommended tags were obtained based on the profiling models constructed from the concepts and the visual features. Li [6] also focused on using visual features in a neighbor voting model. The relevant tags for a given image are retrieved by the votes of similar images. The users’ vocabularies approach also searches all neighbors of a given image according to location, time and visual features from the tagging history of the image’s owner. A tag list is generated from tags of these neighbors and the most frequent tags are recommended [9].

An other approach is based on collective knowledge [17]. Tags correlated with the user-provided tags having higher co-occurrence scores are recommended to the given user. The approach proposed by Garg and Weber [2] also depends on the co-occurrence metric to get global and personal candidate tags correlated to the initial tags.

The correlated scores of tags retrieved from different contexts such as the personal or social tagging history are aggregated to compute the final scores of tags [10]. The social features extracted from users’ social activities are combined with the textual features derived from tags, titles, contents and comments to represent tags. A predictor such as logistic regression or Naïve Bayes is employed to compute the scores of tags.

The relation between users, items and tags is mostly used in factorization models that provide a great performance for tag recommendation. Two of the state-of-the-art models are Pairwise Interaction Tensor Factorization (PITF) [16] and Factorization Machines (FM) [13]. While PITF models all pairwise interactions between users, items and tags with different latent features, FM takes advantage of feature engineering flexibility and powerful predicting capability of the factorization which share the latent features of tags between all pairwise interactions.

Deep learning approaches are applied to image annotation [3,19] that can be viewed as multi-label classification models. The models learn their parameters by optimizing different losses including pairwise and Weighted Approximate Ranking (WARP) or predict labels from arbitrary objects. However, these models provide unpersonalized tag recommendations. It means the recommended tags for similar images are the same for all users.

Factorization models, which do not use image features, cannot recommend tags which are related to the image's contents. They perform worse when recommending tags for new images and they merely recommend the most popular tags by users. In contrast, neural networks that are able to suggest the content-based tags to images will miss personal tags during the recommendation process. We propose a novel approach which combines the best of both worlds. Our model can catch both, tags which are related to the image itself and those which are user-specific, and is able to recommend the most relevant tags to the user.

3 Problem Formulation

The personalized tag recommender will suggest a ranked list of tags to a given user and image. The set of historical tagging assignments represented as $\mathcal{A}$ is a relation between the set of user U, images I and tags T. If user u assigns tag t to image i, the value of $a_{u,i,t} = 1$, or otherwise $a_{u,i,t} = 0$ [7].

The observed tagging set is defined as

$$S := \{(u, i, t) | a_{u,i,t} = 1\}$$

and all observed pairs (u, i), called posts, are grouped in a set [14] that are defined as

$$P_S := \{(u, i) \mid \exists t \in T : (u, i, t) \in S\}$$

Our content-aware recommendation extracts various image features from color images in the set $R := \{R_i \mid i \in I\}$. Visual features of an image $i \in I$ are extracted by the image classification network and denoted as $\mathrm{z}_i \in \mathbb{R}^m$. Object detection features are represented by a vector $\mathrm{o}_i \in \mathbb{R}^n$.

The scoring function $\hat{y}(u, z_i, o_i, t)$ of the image-based recommendation model computes the scores of tags for a given post $p_{u,i}$ which are used to rank tags. If the score $\hat{y}_{u,z_i,o_i,t_a}$ is larger than the score $\hat{y}_{u,z_i,o_i,t_b}$, the tag t_a is more relevant to the post $p_{u,i}$ than the tag t_b.

A content-aware tag recommendation model is expected to provide a top-K tag list $\hat{T}_{u,i}$ that is ranked in descending order of tags' scores for a post $p_{u,i}$.

$$\hat{T}_{u,i} := \underset{t \in T}{\overset{K}{\operatorname{argmax}}}\, \hat{y}(u, z_i, o_i, t) \tag{1}$$

4 The Proposed Architecture

Our personalized image-aware tag recommendation aims at taking benefit of deep learning methods to extract visual information and improve the recommendation capability of factorization models. Our proposed architecture is illustrated in Fig. 2.

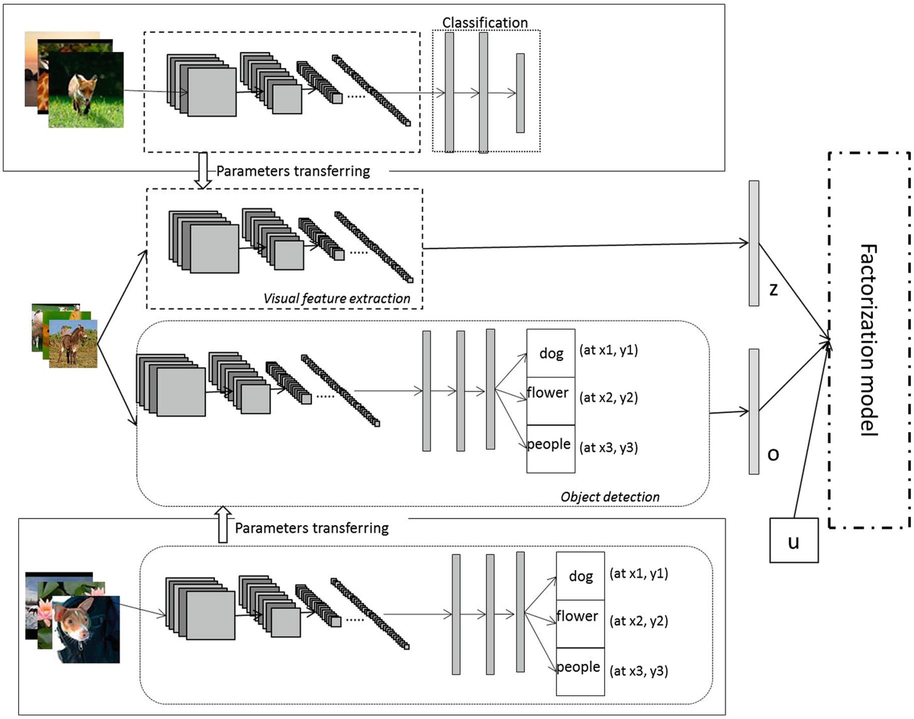


Fig. 2. The proposed architecture for personalized content-aware tag recommendation. We train one network for the task of image classification and one network for the task of object detection on two different data sets. These networks are finally used to extract image features or detect objects. These features and predictions are used as visual features in order to train a factorization model.

A deep neural network is trained on the ImageNet data set for the task of classification and another network is trained on the COCO data set for the task of object detection. The parameters of the networks are transferred to the tag recommender system and used to build the feature extractor.

The image features and the historical tagging assignments are fed to an adapted factorization model to compute the tag scores.

4.1 Visual Feature Extraction

Convolutional neural networks (CNNs) have recently achieved a great success in image classification. They can be used as strong extractors to achieve valuable visual features for images. In these networks, one or more convolutional layers are deployed to generate feature maps by sliding kernel windows across images. Several pooling layers can follow the convolutional layers.

Instead of training all network weights with back-propagation and spending the majority of the run time for learning these parameters, it is very common to use pretrained CNN on large data sets such as ImageNet. Later, the parameters of the convolutional layers are fixed and used as given feature extraction layers.

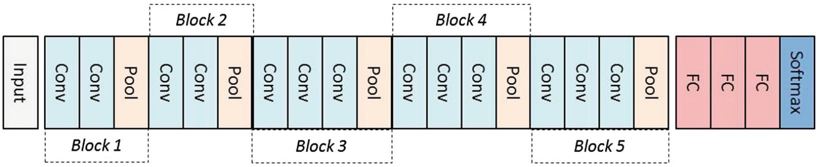


Fig. 3. The architecture of VGG-16 that having 16 weighted layers

One of the state-of-the-art CNN architectures in image classification is the VGG model [18]. The architecture contains multiple convolutional layers located in 5 sequential blocks and several max pooling layers are alternated between these blocks. The predictor block involves several fully-connected layers to predict the probabilities of different labels. The arrangement of the network's layers is illustrated in Fig. 3.

We train a network on the ImageNet data set to achieve a strong image feature extractor which we use for our tag recommender. Firstly, we train a deep neural network using the VGG-16 architecture for the image classification task on ImageNet. Later, all fully connected layers and the softmax layer are removed from the network and a global average pooling replaces these layers in the network. Finally, the network is used as the feature extractor in the tag recommender system and the output of the network is used as the new representation of the image. The extracting features process is formulated as:

$$z_i := f_{\text{vgg16}}(R_i) : \mathbb{R}^{224\times224\times3} \rightarrow \mathbb{R}^m$$

4.2 Object Detection

Deep learning does not only achieve state-of-the-art performance for image classification but is also applied successfully for the task of object detection. One of the state-of-the-art system that works fast and effective is YOLOv2 [12]. It is based on the DarkNet19 architecture that is described in Table 1. It is an improved version of YOLO (You Only Look Once) [11]. YOLO uses a single

Table 1. YOLOv2 is a fully convolutional network and is based on the Darknet-19 architecture sketched below.

Type	Filters	Size/stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

convolutional network in order to predict multiple bounding boxes and the label probabilities for these boxes.

The network comprises multiple convolutional layers mostly having 3×3 filters and the number of feature maps are doubled after each pooling step.

Our proposed architecture uses the probabilities of detected objects as features. If one object has been detected multiple times, we are using the maximum probability of this object. The information of bounding boxes is not used in the models and it is ignored during the extracting process. We train YOLOv2 on the COCO data set. Then, the network is used to extract the object representation for images in tag recommendation. The output of the network is a sparse vector representing for the detected probabilities of 80 categories (one for each object in the COCO data set) and it is denoted as:

$$o_i := f_{\text{YOLO}}(R_i) : \mathbb{R}^{448\times448\times3} \to \mathbb{R}^n$$

4.3 Factorization Models

Two state-of-the-art factorization models applied widely for tag recommendation are factorization machines (FMs) [13] and pairwise interaction tensor factorization (PITF) [16] that model the interaction between different elements of tag assignments. While PITF distinguishes latent features of tags for different pairs of interaction, FM shares these features between all pairs of interaction. In more detail, the input of these models is defined as the following,

$$x_{u,i,t} = \Big(\underbrace{0,\ldots,\overbrace{1}^{u},\ldots,0}_{|U|},\underbrace{0,\ldots,\overbrace{1}^{i},\ldots,0}_{|I|},\underbrace{0,\ldots,\overbrace{1}^{t},\ldots,0}_{|T|}\Big) \qquad (2)$$

The scoring function in FM models is denoted as

$$\hat{y}(u,i,t) = b + \sum_{j=1}^{p} x_j w_j + \sum_{j=1}^{p-1}\sum_{j'=j+1}^{p} x_j x_{j'} \langle \mathrm{v}_j, \mathrm{v}_{j'} \rangle \qquad (3)$$

where $p = |U| + |I| + |T|$ and $\mathrm{v}_j \in \mathbb{R}^k$ are the latent features of the j-th feature. Moreover, $\langle \mathrm{v}_j, \mathrm{v}_{j'} \rangle$ is computed as

$$\langle \mathrm{v}_j, \mathrm{v}_{j'} \rangle = \sum_k v_{j,k} \cdot v_{j',k}$$

Because exactly one x_u, x_i and x_t are one and all others are zero, and we are applying a pair-wise loss function, the prediction function of the FM can be simplified to

$$\hat{y}(u,i,t) = w_t + \sum_{j=1}^{k} (v^U_{u,j} + v^I_{i,j}) v^T_{t,j} \qquad (4)$$

where k is the number of latent features, $V^U \in \mathbb{R}^{|U|\times k}$, $V^I \in \mathbb{R}^{|I|\times k}$ and $V^T \in \mathbb{R}^{|T|\times k}$ are the latent features of users, images and tags.

Similarly, the PITF prediction model simplifies to

$$\hat{y}(u,i,t) = \sum_{j=1}^{k} v^U_{u,j} \cdot v^{T^U}_{t,j} + v^I_{i,j} \cdot v^{T^I}_{t,j} \qquad (5)$$

where model parameters are denoted as $V^U \in \mathbb{R}^{|U|\times k}$, $V^I \in \mathbb{R}^{|I|\times k}$, $V^{T^U} \in \mathbb{R}^{|T|\times k}$ and $V^{T^I} \in \mathbb{R}^{|T|\times k}$.

The models are plainly based on the relation between different elements and use the index of all elements as their input. We cannot directly apply these models to content-aware recommendation where the input contains information of images representing in feature vectors.

4.4 Factorization Models for Image-Aware Tag Recommendation

The aforementioned factorization models focus on using users' preferences, instead of using contents of images. To feed image-based features to the factorization models, the part representing the image in Eq. (2) is replaced by its image-based features. If the features are the combination of visual and object features, it is denoted as:

$$x_{u,z_i,o_i,t} = \Big(\underbrace{\ldots,\overbrace{1}^{u},\ldots}_{|U|},\underbrace{z_{i_1},\ldots,z_{i_m}}_{m},\underbrace{o_{i_1},\ldots,o_{i_n}}_{n},\underbrace{\ldots,\overbrace{1}^{t},\ldots}_{|T|}\Big)$$

Depending on the types of features used to predict the tags' scores, the part of unused features is removed from the input.

Based on the description of the input, we propose different factorization models based on FM and PITF to generate the scoring functions.

If both types of features are used to predict the relevant tags, the scoring functions are formulated as:

- The FM-based formula is:

$$\hat{y}(u, z_i, o_i, t) = w_t + \sum_{j=1}^{k}\left(v^U_{u,j} + \sum_{a=1}^{m} z_{i_a} \cdot v^Z_{a,j} + \sum_{a=1}^{n} o_{i_a} \cdot v^O_{a,j}\right) v^T_{t,j} \qquad (6)$$

- The PITF-based function is:

$$\hat{y}(u, z_i, o_i, t) = w_t + \sum_{j=1}^{k} v^U_{u,j} \cdot v^{T^U}_{t,j} + (\sum_{a=1}^{m} z_{i_a} \cdot v^Z_{a,j}) v^{T^Z}_{t,j} + (\sum_{a=1}^{n} o_{i_a} \cdot v^O_{a,j}) v^{T^O}_{t,j} \quad (7)$$

If the input contains one type of features, the parameters associated with the unused features are removed from the formula.

Depending on the types of image-based features and the scoring function, the models are named differently. In detail, **FM-OD** and **PITF-OD** use only the object detection features while **FM-IC** and **PITF-IC** use the feature extraction obtained by the image classification knowledge. **FM-IC-OD** and **PITF-IC-OD** use all image-based features.

4.5 Optimization

The criterion of the optimization used is Bayesian Personalized Ranking (BPR) optimization criterion [15]. The parameters found satisfy that the difference between the relevant and irrelevant tags are maximal.

The stochastic gradient descent applied to BPR is in respect of quadruples (u, i, t^+, t^-); i.e., for each $(u, i, t^+) \in S_{train}$ and an unobserved tag of $p_{u,i}$ drawn at random t^-, the loss is computed and is used to update the model's parameters.

$$\text{BPR}(u, z_i, o_i, t^+, t^-) := \ln \sigma(\hat{y}'(u, z_i, o_i, t^+, t^-)) \qquad (8)$$

where

$$\sigma(\psi) = \frac{1}{1 + e^{-\psi}}$$

The tag assigned by the user u for image i, called t^+ and the unobserved tag t^- of the pair (u, i) are denoted as

$$t^+ \in T^+_{u,i} := \{t \in T \mid (u, i, t) \in S_{train}\}; \quad t^- \in T^-_{u,i} := \{t \in T \mid (u, i, t) \notin S_{train}\}$$

Moreover, the difference between two types of tags is defined as

$$\hat{y}'(u, z_i, o_i, t^+, t^-) = \hat{y}'(u, z_i, o_i, t^+) - \hat{y}'(u, z_i, o_i, t^-)$$

The learning algorithm is described in Algorithm 1. For each random post, a relevant tag and an irrelevant tag are sampled and the scores of these tags are computed. The gradients of the cost function in Eq. (8) with respect to the model's parameters are obtained as follows:

$$\frac{\partial \text{BPR}}{\partial \Theta} = \frac{e^{-\hat{y}'(u,z_i,o_i,t^+,t^-)}}{1 + e^{-\hat{y}'(u,z_i,o_i,t^+,t^-)}} \times \left(\frac{\partial \hat{y}'(u, z_i, o_i, t^+)}{\partial \Theta} - \frac{\partial \hat{y}'(u, z_i, o_i, t^-)}{\partial \Theta} \right) \qquad (9)$$

Algorithm 1. Learning BPR

```
Input: $P_S$, $S$, $Z$, $O$, $\alpha$
Output: $\Theta$

Initialize $\Theta \leftarrow \mathcal{N}(0, 0.1)$
repeat
    Pick $(u, i) \in P_{S_{train}}$, $z_i \in Z$, $o_i \in O$
    Get $t^+_{u,i} \in T$ and $(u, i, t) \in S$
    Pick $t^-_{u,i} \in T$ randomly whereas $(u, i, t) \notin S$
    Compute $\hat{y}'(u, z_i, o_i, t^+)$ and $\hat{y}'(u, z_i, o_i, t^-)$
    Update $\Theta \leftarrow \Theta + \alpha\left(\frac{\partial \text{BPR}(u,z_i,o_i,t^+,t^-)}{\Theta}\right)$
until convergence
return $\Theta$
```

In order to learn the model, the gradients $\frac{\partial \hat{y}'(u,z_i,o_i,t^+)}{\partial \Theta}$ and $\frac{\partial \hat{y}'(u,z_i,o_i,t^-)}{\partial \Theta}$ have been computed. For examples, from Eq. (6), the derivatives with respect to parameters of tags are computed as:

$$\frac{\partial \hat{y}'(u, z_i, o_i, t)}{\partial v^T_{t,j}} = v^U_{u,j} + \sum_{a=1}^{m} z_{i_a} \cdot v^Z_{a,j} + \sum_{a=1}^{n} o_{i_a} \cdot v^O_{a,j}$$

5 Evaluation

5.1 Dataset

We obtained experiments on subsets of the publicly available multilabel data set NUS-WIDE [1] that contains 269,648 images. We preprocessed the first subset by keeping available images tagged by the 100 most popular tags, sampling 1.000 users, refining to get 10-core dataset referring to users and tags where each user or tag occurs at least in 10 posts [4]. Later we remove tags assigning more than 50% of images by one user to avoid the case that users tag all their images by the same words.

In a similar way, the second subset is obtained after several steps. First, tags are filtered by matching to WordNet [8] and only English tags are kept. Later, the data set is refined to get 20-core regarding to users, 100-core to tags and removing tags annotating more than 50% of images by one user.

Table 2. Dataset characteristics

Dataset	Users $\lvert U\rvert$	Images $\lvert I\rvert$	Tags $\lvert T\rvert$	Triples $\lvert S\rvert$	Posts $\lvert P_S\rvert$	Training posts $\lvert P_{S_{train}}\rvert$	Test posts $\lvert P_{S_{test}}\rvert$
NUS-WIDE-1	1000	27.662	100	81.263	27.858	25.858	2.000
NUS-WIDE-2	1.999	90.483	1.661	634.739	95.130	76.842	18.288

We created our train/test split using leave-one-post-out [7]: for each user in NUS-WIDE-1, 2 posts are randomly picked and put into its test set. Similarly, 20% of NUS-WIDE-2 posts for each user are sampled to put into the test set. These data sets are described with respect to users, images, tags, triples and posts as in Table 2. The color images used to extract features are crawled from Flickr and rescaled into 224 × 224 dimension. The distribution of posts per tag in NUS-WIDE-1 is more balanced than in NUS-WIDE-2 which has more than 50% of tags appearing less than 500 times.

5.2 Experimental Setup

The visual features extracted are combined in a 512-dimension vector while the object recognition probabilities of a given image are appended in a 80-dimension vector.

The factor dimension for both factorization architecture is fixed in 128. The evaluation metric used in this paper is the F1-measure in top K tag lists where K is in the range of 1 to 10.

$$\text{F1@K} = \frac{2 \cdot \text{Prec@K} \cdot \text{Recall@K}}{\text{Prec@K} + \text{Recall@K}} \tag{10}$$

where

$$\text{Prec@K} = \underset{(u,i)\in S_{test}}{\text{avg}} \frac{|\hat{T}_{u,i} \cap T_{u,i}|}{K} \qquad \text{Recall@K} = \underset{(u,i)\in S_{test}}{\text{avg}} \frac{|\hat{T}_{u,i} \cap T_{u,i}|}{|T_{u,i}|}$$

$$\hat{T}_{u,i} = \text{Top}(u, z_i, o_i, K) = \underset{t\in T}{\overset{K}{\text{argmax}}}\, \hat{y}(u, z_i, o_i, t)$$

The best learning rate α are searched within the range $\{10^{-2}, 10^{-3}, 10^{-4}\}$ and the best L2-regularization λ are found from the range $\{10^{-5}, 10^{-6}, 10^{-7}\}$. The proposed models **FM-IC-OD** and **PITF-IC-OD** are compared to following personalized tag recommendation approaches that are based only on the users' preference: **PITF** [16] and **FM** [13].

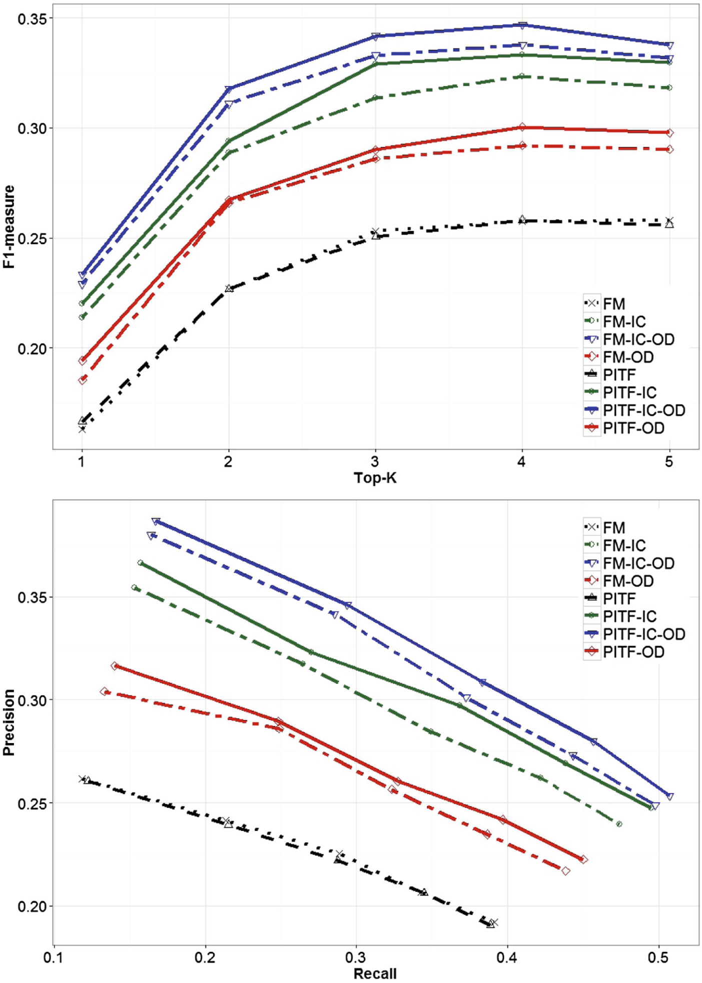


Fig. 4. F1-measure and Precision-Recall for NUS-WIDE-1

Moreover, these models are also compared to the factorization models using visual features or object detection features: **FM-OD**, **PITF-OD**, **FM-IC** and **PITF-IC**.

5.3 Results

As shown in Figs. 4 and 5, the personalized models **FM** and **PITF** which do not consider content information have the worst performance. They solely depend on the users' preferences and their power in catching the interaction between new images with other elements is not effective. In the NUS data set, most images in the test set do not appear in the training set and their latent parameters are not learned.

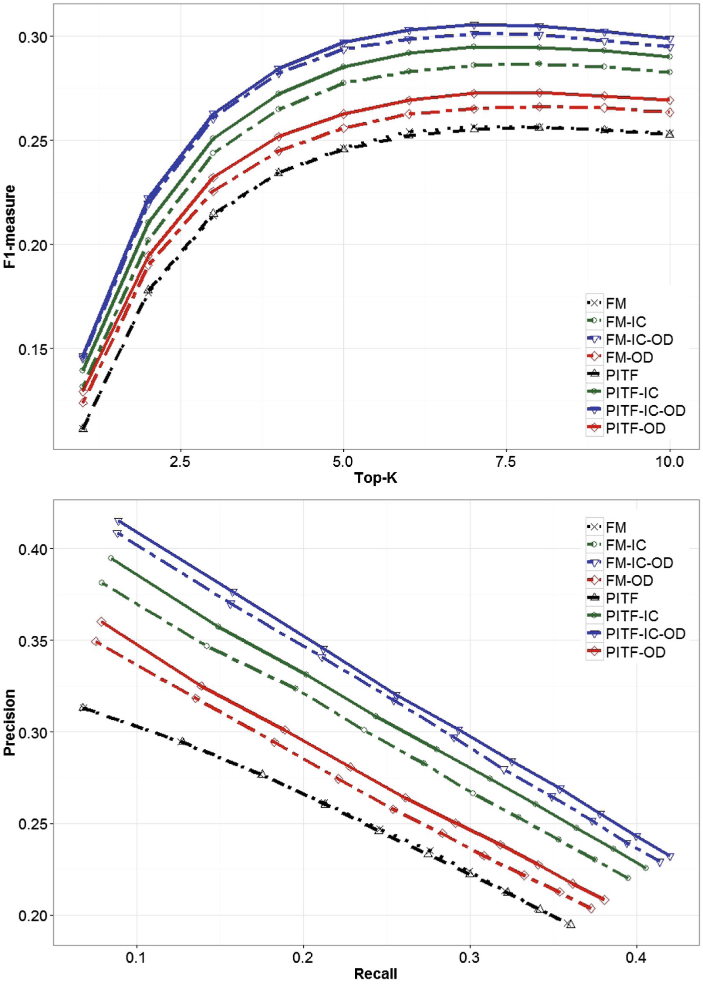
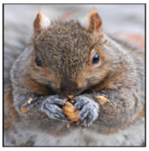
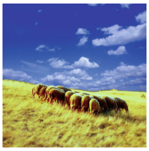
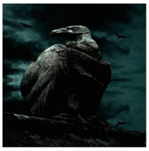
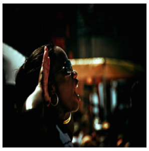


Fig. 5. F1-measure and Precision-Recall for NUS-WIDE-2

The claim that image-features improve the prediction quality is clearly shown in these figures. These features boost the performance from 1% to more than 3%. The object detection features are less effective than other features while the combination of image-based features helps improving accuracy the most effectively. Because the most popular tags in these data set are related to color such as blue or green, the object detection cannot capture these information and the models using them will miss these tags.

Otherwise, the visual features capture more unique information of a given images. For this reason, the performance of the models using visual features is better than the models using only the information of object detection. The combination of image-based features prove the powerful abilities in boosting performance. They can capture general object information and unique visual features of a given image. So these features are richer than other features and the accuracy of the **FM-IC-OD** and **PITF-IC-OD** model provide the best results.

Table 3. Examples top recommended tags of factorization models using different types of features

Image	Ground truth	FM-OD	FM-IC	FM-IC-OD
	wildlife cute squirrel	animal wildlife cat eating squirrel	cat pet dog wildlife lion	wildlife squirrel animal cat eating
	mountain sheep	sheep germany landscape field deutschland	germany deutschland landscape green england	sheep germany landscape field france
	clouds gothic stone bird dark castle wall	bird sun sky water tree blue silhouette beautiful prey sunset	wales cloud water black fresh silhouette sun fab mountain waterfall	bird water lens black white sun aqua sky fab wales
	dark candid sunglasses people city	film people candid dark street	film candid dark night shoes	people film dark candid city

Moreover, the PITF-based models generally work better than the FM-based in most cases. They separate the latent features of tags depending on the elements that they interact with. So they can capture the different representative of tags and combine the scores computing for each interaction into the final score. The difference between the PITF-based and FM-based approaches is clearly in the models using visual or object features while the performances of the models using both features are nearly compatible.

Examples in Table 3 show that the proposed models can capture the visual-based tags and object-tags compared to the models that are purely based on one type of image features. For example, **FM-IC-OD** recommends to a given user the object-based tag as "bird" and the visual-based tag as "black" in the fourth images.

6 Conclusion

In this paper, we showed how to extract image features using transfer learning. We used two different data sets in order two train two different neural networks for the task of image classification and object detection. We used these networks to extract powerful image features in order to improve the performance of the current state of the art for tag recommendation. Our proposed approach is able to recommend tags related to objects in images, tags representing image attributes and tags which are typically chosen by a user. For this reason, the performance of the models has been improved at least up to 1%. The experiments show that different types of image-based features improve the accuracy of tag recommendation in different levels. In the future, the contents used in the recommendation are not only limited on the information of images but are also broadened to contents of users such as vocabularies of users or their social activities.

References

1. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: NUS-WIDE: a real-world web image database from national university of Singapore. In: Proceedings of the ACM International Conference on Image and Video Retrieval, p. 48 (2009)
2. Garg, N., Weber, I.: Personalized, interactive tag recommendation for flickr. In: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 67–74 (2008)
3. Gong, Y., Jia, Y., Leung, T., Toshev, A., Ioffe, S.: Deep convolutional ranking for multilabel image annotation. arXiv preprint arXiv:1312.4894 (2013)
4. Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in folksonomies. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 506–514. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74976-9_52
5. Li, J., Wang, J.Z.: Real-time computerized annotation of pictures. IEEE Trans. Pattern Anal. Mach. Intell. **30**(6), 985–1002 (2008)

6. Li, X., Snoek, C.G., Worring, M.: Learning tag relevance by neighbor voting for social image retrieval. In: Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval, pp. 180–187 (2008)
7. Marinho, L.B., Hotho, A., Jäschke, R., Nanopoulos, A., Rendle, S., Schmidt-Thieme, L., Stumme, G., Symeonidis, P.: Recommender Systems for Social Tagging Systems. Springer Science & Business Media, Heidelberg (2012). https://doi.org/10.1007/978-1-4614-1894-8
8. Miller, G.A.: Wordnet: a lexical database for English. Commun. ACM **38**, 39–41 (1995)
9. Qian, X., Liu, X., Zheng, C., Du, Y., Hou, X.: Tagging photos using users' vocabularies. Neurocomputing **111**, 144–153 (2013)
10. Rae, A., Sigurbjörnsson, B., van Zwol, R.: Improving tag recommendation using social networks. In: Adaptivity, Personalization and Fusion of Heterogeneous Information, pp. 92–99 (2010)
11. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
12. Redmon, J., Farhadi, A.: YOLO9000: Better, faster, stronger. arXiv preprint arXiv:1612.08242 (2016)
13. Rendle, S.: Factorization machines. In: 2010 IEEE 10th International Conference on Data Mining (ICDM), pp. 995–1000 (2010)
14. Rendle, S., Balby Marinho, L., Nanopoulos, A., Schmidt-Thieme, L.: Learning optimal ranking with tensor factorization for tag recommendation. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 727–736 (2009)
15. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461 (2009)
16. Rendle, S., Schmidt-Thieme, L.: Pairwise interaction tensor factorization for personalized tag recommendation. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 81–90 (2010)
17. Sigurbjörnsson, B., Van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: Proceedings of the 17th International Conference on World Wide Web, pp. 327–336 (2008)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
19. Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., Yan, S.: CNN: single-label to multi-label. arXiv preprint arXiv:1406.5726 (2014)