Neurocomputing 395 (2020) 229–236

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Visual concept conjunction learning with recurrent neural networks

Kongming Liang[a,b], Hong Chang[a,*], Shiguang Shan[a], Xilin Chen[a,b]

[a] *Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China*
[b] *University of Chinese Academy of Sciences, Beijing 100049, China*

ARTICLE INFO

Article history:
Received 29 July 2017
Revised 28 November 2017
Accepted 8 December 2017
Available online 13 July 2019

Keywords:
Attribute learning
Concept conjunction
Visual relationship detection
Image retrieval

ABSTRACT

Learning the conjunction of multiple visual concepts shows practical significance in various real world applications (e.g. multi-attribute image retrieval and visual relationship detection). In this paper, we propose Concept Conjunction Recurrent Neural Network (C^2RNN) to tackle this problem. With our model, visual concepts involved in a conjunction are mapped into the hidden units and combined in a recurrent way to generate the representation of the concept conjunction, which is then used to compute a concept conjunction classifier as the output. We also present an order invariant version of the proposed method based on attention mechanism to learn the tasks without pre-defined concept order. To tackle concept conjunction learning from multiple semantic domains, we introduce a multiplicative framework to learn the joint representation. Experimental results on multi-attribute image retrieval and visual relationship detection show that our method achieves significantly better performance than other related methods on various datasets.

1. Introduction

To understand the richness of the visual world, computer needs to perceive all the existing semantic concepts in an image and further reason over them. As the basic building blocks of image understanding, object recognition [1] aims to know what kind of objects appear in an image and object detection [2] focuses on accurately localizing each of the appearing objects. Based on that, attribute learning provides a promising way for computer to understand image content from holistic perception (e.g., color, shape, etc.) to the presence or absence of local parts of each objects. Traditional attribute learning takes each single visual attribute as a mid-level feature to benefit many computer vision problems (e.g., object recognition [3], face verification [4] and zero-shot classification [5]).

Similar to the above three tasks, traditional visual learning problem is dedicated to recognizing single concept from a specific semantic domain. However, learning the conjunction of multiple concepts shows more practical significance. By taking attribute as an example of concept, learning attribute conjunctions can be used to retrieve relevant images based on multi-attribute query in the form of $\{attribute_1, attribute_2, \ldots\}$. Previous works on multi-attribute retrieval have shown its effectiveness in discovering the objects with specified characteristics [6,7], e.g., searching people based on certain facial descriptions [8] and matching products according to users' requirements [9]. Beyond attribute conjunction learning, visual relationship detection (VRD) tries to learn the conjunction of concepts from multiple semantic domains (i.e., objects and predicates). By localizing a pair of objects, it can reason the predicate between the objects, which forms a conjunction as $\{object_1, predicate, object_2\}$. In that way, VRD can be used as the intermediate level task for image caption generation and visual question answering and benefit image understanding with respect to holistic semantics.

A common approach to tackle concept conjunction learning is transforming the problem into multiple single concept learning tasks. Specifically, a binary classifier is built for each single concept, then the result of multiple concept prediction is generated by summing up the scores of all single concept classifiers. Though this kind of combination is simple and shows good scalability, it has two main drawbacks. Firstly, the correlation between concepts is ignored because of the separate training of each concept classifier. Secondly, concept classification results are sometimes unreliable since abstract linguistic properties can have very diverse visual manifestations which need a large amount of data to recognize the patterns. This situation may get worse with larger number of concepts appearing in a conjunction simultaneously. For example, when the conjunction length is three, an unreliable concept classifier may affect $\binom{C-1}{2}$ conjunctions (C is the total number of concepts).

Instead of training a classifier for each concept separately, a more promising approach is to directly learn from the concept

[*] Corresponding author.
E-mail address: hong.chang@vipl.ict.ac.cn (H. Chang).

https://doi.org/10.1016/j.neucom.2017.12.068
0925-2312/© 2019 Published by Elsevier B.V.

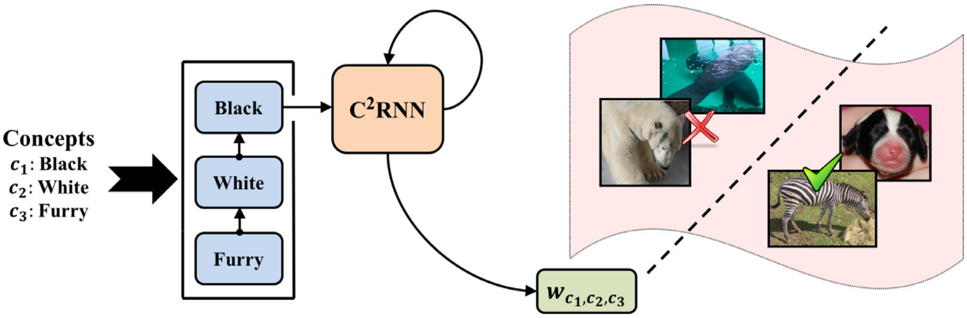

Fig. 1. An illustration of concept conjunction recurrent neural network (C^{2}RNN).

conjunctions. Since conjunctions of multiple concept may lead to very characteristic appearances, training a classifier that detects the conjunctions as a whole may produce more accurate results. For example, training a classifier to predict whether the animal is (black & white & stripe) leads to a specific concept "Zebra". However, straightforward training classifiers for concept conjunctions is not a good choice. Firstly, the number of concepts appearing in a conjunction is not fixed and the number of concept conjunctions grows exponentially w.r.t. the number of the concepts. To learn a conjunction containing three concepts, we need to build $\binom{C}{3}$ classifiers for all possible concept conjunctions. Secondly, there are only a small number of positive examples for each multiple concept conjunction (a positive sample must have multiple concepts simultaneously), which brings the learning process more difficulties. With data bias problem, some concept conjunction classifiers may perform even worse than simply adding the scores of disjoint single concept classifiers. Thirdly, the correlation between concept conjunctions is not well explored, since the queries which share common concepts are considered to be independent from each other. Last but not the least, the concepts within a conjunction may come from different semantic domains. For example, the conjunction defined in visual relationship detection contains both objects and predicates. Therefore, how to combine the concepts from different semantic domains is also very critical.

In this paper, we propose a novel concept conjunction recurrent neural network (C^{2}RNN) to tackle multiple concept conjunction learning problem. As shown in Fig. 1, the input sequences of C^{2}RNN are the concepts appearing in the conjunction with a predefined order. Each of the input concepts is then embedded into the hidden units and combined in a recurrent way to generate the representation for the concept conjunction. The conjunction representation is further used to generate the classifier for recognizing the specific concept conjunction. As the multiple concepts in each conjunction are processed by the network recurrently, the number of parameters of our model do not increase with the length of conjunction. Compared with straightforward multiple concept learning methods, our proposed C^{2}RNN model can appropriately model the complex relationship among different concept conjunctions by sharing the representations of single concepts. We also introduce a data weighting strategy to address the data bias problem in multiple concept conjunction learning. Finally, we introduce a multiplicative model to explicitly learn the conjunctions which consist of the concepts from different semantic domains.

The rest of this paper is organized as follows. We first introduce some related works in the following section. In Section 3, we present the concept conjunction recurrent neural network in detail. Experimental results are then shown in Section 4. Finally, Section 5 concludes the paper.

2. Related work

Multi-Attribute Query [1]: Guillaumin et al. [10] propose to use a weighted nearest-neighbour model to predict the tags of a test image which can directly support multi-word query based image retrieval. Petterson and Caetano [11] present a reverse formulation to retrieve sets of images by considering labels as input. In this way, they can directly optimize the convex relaxations of many popular performance measures. By leveraging the dependencies between multiple attributes, Siddiquie et al. [12] explicitly model the correlation between query-attributes and non-query attributes. For example, for a query such as "man with sunglasses", the correlated attributes like beard and mustache can also be used to retrieve relevant images. Since training a classifier for a combination of query attributes may not always perform well, Rastegari et al. [13] propose an approach to determine whether to merge or split attributes based on the geometric quantities. Different from the above methods, we propose to explicitly learn all the single attribute embeddings and combine them in a recurrent way to generate the representation of attribute conjunction.

Multi-label Learning: Read et al. [14] extend traditional binary relevance method by predicting multiple attributes progressively in an arbitrary order. For instance, one label is predicted first. Then the prediction result is appended at the end of the input feature which is used as the new feature to predict the second label. Finally, the multiple label predictions are formed into a classifier chain. Since a single standalone classifier chain model can be poorly ordered, the authors also propose an ensemble method in a voting scheme. Zhang and Wu [15] exploit different feature sets to benefit the discrimination of multiple labels. This method exploits conducting clustering analysis on the positive and negative instances and then performs training and testing referring to the clustering results. Different from multi-label learning problems, the task we deal with here is to model label conjunctions instead of multiple separate labels.

Label embedding: Since deep learning provides a powerful way to learn data representations, many researchers replace handcrafted feature designing to automatically feature learning. Meanwhile, how to represent labels is also a key issue for machine learning methods. A common way is Canonical Correlation Analysis (CCA) [16] which maximizes the correlation between data and labels by projecting them into a common space. Another promising way is to learn the embedding of labels by leveraging other possible sources as prior information. Akata et al. [17] propose to embed category labels into attribute space under the assumption that attributes are shared across categories. Frome et al. [18] represent category labels with the embedding learned from textual data in an unsupervised way. Hwang et al. [19] jointly embed all seman-

[1] The multi-attribute we denote here may refer to other statements such as keywords or multi-label in other literature.

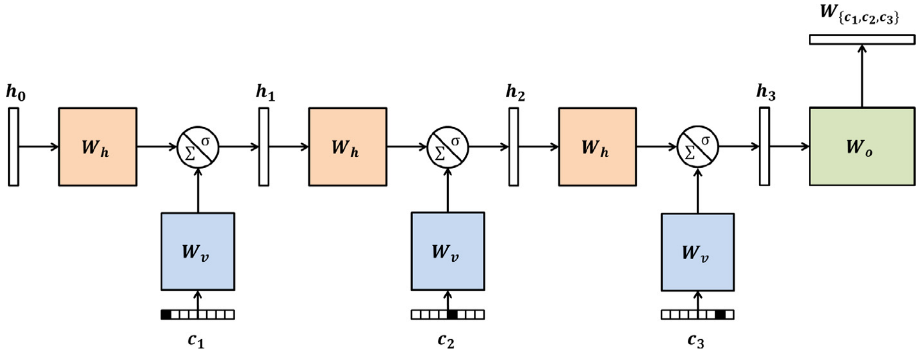

Fig. 2. An Illustration of Concept Conjunction Recurrent Neural Network (C^{2}RNN) with three concepts.

tic entities including attributes and super-categories into the same space by exploiting taxonomy information. But so far, there is no work on learning the conjunction representation of multiple labels to the best of our knowledge.

Visual relationship detection: Beyond object recognition, visual relationship detection localizes a pair of objects and classifying the predicate between them. The task is to predict a conjunction of two objects and their predicate. Lu el al. [20] propose to use the union of the paired objects and leverage the language priors to predict multiple relationships per image. Zhang et al. [21] propose a translation embedding framework by modeling the predicate as the difference of two paired objects. Liang et al. [22] propose a novel deep reinforcement learning framework to recognize all the appearing visual relationships sequentially. By integrating a triplet proposal procedure, Li et al. [23] only detect the relationships for the candidates paired objects using message passing.

3. Our method

The problem we aim to address here is to present and learn concept conjunctions which are relevant to an image. Intuitively, a conjunction usually consists of single concepts and the correlation between them is usually strong. Therefore we determine to learn from all conjunctions jointly. Firstly, we propose to use the recurrent neural network to model the conjunction function for multiple concepts. The model can not only reveal the representation of concept conjunctions but also output the concept conjunction classifiers. Secondly, we propose a method based on attention mechanism for learning the conjunctions where the order of concepts does not matter. Thirdly, we introduce a multiplicative framework to combine the representations of the conjunctions from different semantic domains. Finally, we integrate output of the C^{2}RNN into a classification model (e.g. logistic regression). The parameters of recurrent neural network and the classification model are optimized simultaneously using back propagation. We also propose a weighting version of our model to tackle data imbalance problem.

3.1. Concept conjunction learning in pre-defined order

Let $\mathcal{C} = \{\mathbf{C}^1, \mathbf{C}^2, \dots, \mathbf{C}^M\}$ be a set of M multiple concept conjunctions. The mth conjunction is represented as a matrix $\mathbf{C}^m = (\mathbf{c}_1^m, \mathbf{c}_2^m, \dots, \mathbf{c}_{T_m}^m) \in \{0,1\}^{C \times T_m}$, where C is the number of predefined concepts and T_m is the number of concepts appearing in the mth conjunction. $\mathbf{c}_t^m$ is a one-hot query vector where the non-zero item represents the current concept.

With the pre-defined order, our model takes multiple concepts as input and outputs the representation of their conjunction. More specifically, for the mth concept conjunction, the one-hot query vectors $\mathbf{c}_t^m$ $(t = 1, \dots, T_m)$ corresponding to the concepts involved in the query are input sequentially to our model. The subscript t decides the input order. We learn the multiple concept conjunction in a recurrent way, as illustrated in Fig. 2. In this model, the first t concepts of the mth conjunction can be represented as:

$$\begin{aligned}\mathbf{h}_t^m &= f_h\left(\mathbf{c}_t^m, \mathbf{h}_{t-1}^m\right) \\ &= \sigma\left(\mathbf{W}_v \mathbf{c}_t^m + \mathbf{W}_h \mathbf{h}_{t-1}^m + \mathbf{b}_h\right), \end{aligned} \tag{1}$$

where f_h is a *conjunction function* to model the relationship of all the concepts belonging to the mth conjunction. $\mathbf{W}_v \in \mathbb{R}^{H \times A}$ and $\mathbf{W}_h \in \mathbb{R}^{H \times H}$ are *embedding* and *conjunction matrix* respectively, where H is the number of hidden units of the recurrent network. $\mathbf{h}_0^m \equiv \mathbf{h}_0$ represents the initial hidden state. $\mathbf{b}_h$ is the bias and $\sigma(\cdot)$ is an element-wise non-linear function which is chosen to be sigmoid in this paper.

From Eq. (1), we can see that each column of parameter matrix $\mathbf{W}_v$ can be considered as single concept representation, noting that the input vector is in one-hot form. Therefore, all the concept conjunctions share the same concept-level representations. In this way, the parameter growth problem for long conjunction is well addressed.

After computing $\mathbf{h}_{T_m}^m$ of the last concept vector with the C^{2}RNN, we obtain the hidden representation of the whole concept conjunction:

$$h(\mathbf{C}^m) = \mathbf{h}_{T_m}^m. \tag{2}$$

3.2. Order invariant concept conjunction learning

Recurrent neural networks are well suited to model sequential data. However, the input concepts are sometimes not naturally organized as a sequence since the underlying conditional dependency between concepts are not known. To learn the order invariant conjunction representation, we propose to introduce the recurrent attention mechanism into the learning procedure.

Attention mechanism has been successfully applied in generating image caption [24], handwriting recognition [25], action recognition [26] and machine translation [27]. And it has been used to model the input and output structure of a sequence to sequence framework [28]. Inspired by the previous works, we propose to integrate the attention mechanism into our model to tackle the ordering problem. The pipeline is shown in Fig. 3. The proposed network reads all the input concepts according to an attention vector, instead of processing the concepts one by one at each step. The attention vector is a probability vector indicating the relevance of all pre-defined concepts to the current conjunction. And it is automatically modified at each processing step and recurrently contributes to the representation of the concept conjunction. Intuitively, we first initialize the input attention vector for the mth conjunction as:

$$\mathbf{a}_1^m = \frac{\sum_{i=1}^{T_m} \mathbf{c}_i^m}{T_m}. \tag{3}$$

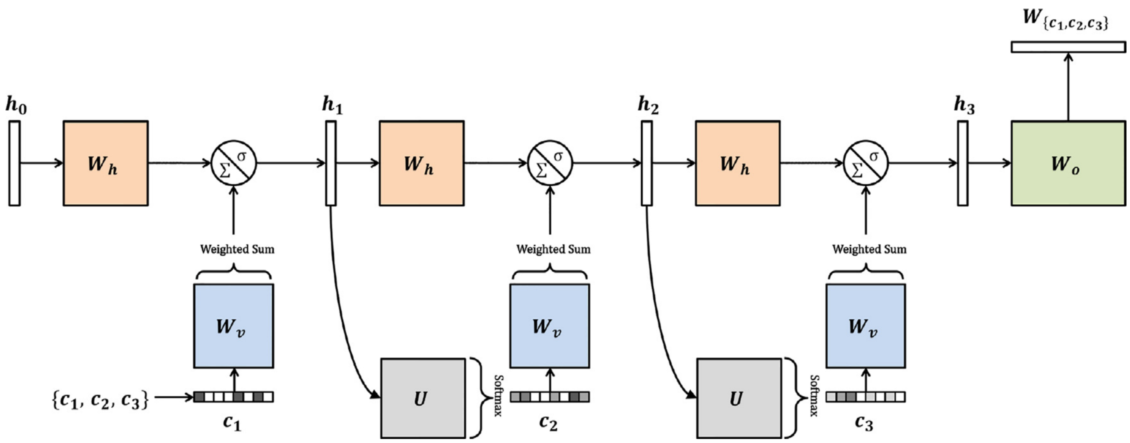

Fig. 3. C^{2}RNN with attention mechanism (C^{2}RNN-ATN).

In this way, the network will first take all the input concepts into attention. Then we refined the attention vector and learn the concept conjunction step by step using the recurrent neural network with attention mechanism. In the *s*th step, the concept conjunction and attention vector are generated as follows:

$$\mathbf{h}_s^m = f_h\left(\mathbf{a}_s^m, \mathbf{h}_{s-1}^m\right) = \sigma\left(\mathbf{W}_v\mathbf{a}_s^m + \mathbf{W}_h\mathbf{h}_{s-1}^m + \mathbf{b}_h\right), \tag{4}$$

$$\mathbf{a}_{s+1}^m = Softmax(\mathbf{U}\mathbf{h}_s^m) = \frac{1}{\sum_{j=1}^A e^{\mathbf{U}_j^\mathrm{T}\mathbf{h}_s^m}} \begin{bmatrix} e^{\mathbf{U}_1^\mathrm{T}\mathbf{h}_s^m} \\ e^{\mathbf{U}_2^\mathrm{T}\mathbf{h}_s^m} \\ \vdots \\ e^{\mathbf{U}_A^\mathrm{T}\mathbf{h}_s^m} \end{bmatrix}, \tag{5}$$

where the *attention matrix* $\mathbf{U} \in \mathbb{R}^{H\times A}$ transforms the hidden units into the attention vector of the next processing step. Other parameters are consistent with the definition in Section 3.1.

By using a recurrent attention model, the output of C^{2}RNN is invariant to the input concept order. In addition, by considering the concepts not existing in the current conjunction, this model can leverage the co-occurrence information to enhance the current conjunction. Therefore, an unreliable concept might piggyback on an co-occurring concept that has abundant training data and easier to predict.

3.3. Concept conjunction learning with multiple semantic domains

The concepts appearing in a conjunction may come from different semantic domains. Here we take visual relationship detection as an example. The $\mathbf{C}^m$ is composed of two part $\mathbf{C}_{obj}^m$ and $\mathbf{C}_{pre}^m$ which denote the concepts in the object and predicate domain respectively. For the object domain, the concept order is pre-defined. For example, in {*person, bike*} → *ride*, the relative position of person and bike can not be changed since person is the subject and bike is the object. Though the conjunctions in the predicate domain only contain one concept, we can still use C^{2}RNN to learn their representations. Therefore, the conjunction representations for two involved domains can be acquired as $h_{obj}(\mathbf{C}_{obj}^m)$ and $h_{pre}(\mathbf{C}_{pre}^m)$.

To learn the joint representation of the conjunctions from two semantic domains, we propose to combine the above two conjunction representations in a multiplicative way inspired by the previous work [29]. The resulting conjunction representation is as following:

$$h(\mathbf{C}^m) = h_{obj}\left(\mathbf{C}_{obj}^m\right) \odot h_{pre}\left(\mathbf{C}_{pre}^m\right), \tag{6}$$

where $\odot$ denotes element-wise multiplication. Then the conjunction representation can be used to generate the classifier of the corresponding conjunction which consists of the concepts from multiple semantic domains.

3.4. Concept conjunction classification

Based on the representations of the concept conjunction, we further stack one layer on top of the recurrent units and compute the *concept conjunction classifier* $\mathbf{w}_m$ as following:

$$\mathbf{w}_m = f_o(h(\mathbf{C}_m)) = \mathbf{W}_o h(\mathbf{C}_m) + \mathbf{b}_o. \tag{7}$$

Here, the regression function f_o is chosen to be in a linear form, though more complex form can be considered. The parameter $\mathbf{W}_o$ and $\mathbf{b}_o$ are the output matrix and bias respectively. In this way, the concept embeddings of the current conjunction are combined in a recurrent way to learn the complex relationship between the concepts. After that we use the output as the *m*th multiple concept conjunction classifier. The model parameters of the conjunction and output functions are denoted as $\Theta = \{\mathbf{W}_v, \mathbf{W}_h, \mathbf{W}_o, \mathbf{b}_h, \mathbf{b}_o, \mathbf{h}_0\}$.

Suppose there are N labeled instances, $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^D$ denotes the D-dimensional input feature vector, and $\mathbf{y}_i \in \{0, 1\}^C$ indicates the absence and presence of all concepts. The concept label can be expressed in matrix form as $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] \in \{0, 1\}^{C\times N}$. In order to recognize the multiple concept conjunction $\mathbf{C}^m$, we resort to a classification model to estimate the labels $\mathbf{Y}$.

Since learning each concept is a binary classification problem, we make use of logistic regression to predict the absence or presence of multiple concepts. The loss function with respect to the *m*th concept conjunction is expressed as the following negative log likelihood:

$$\begin{aligned} L(\mathbf{x}_i, \mathbf{y}_i, \mathbf{C}^m; \Theta) = &-\tilde{\mathbf{y}}_{im} log\left(\sigma(\mathbf{w}_m^T\mathbf{x}_i)\right) \\ &-(1-\tilde{\mathbf{y}}_{im}) log\left(1-\sigma(\mathbf{w}_m^T\mathbf{x}_i)\right), \end{aligned} \tag{8}$$

where $\tilde{\mathbf{y}}_{im} = (\mathbf{y}_i^T\mathbf{c}_1^m \,\&\, \mathbf{y}_i^T\mathbf{c}_2^m \,\&\, \ldots \,\&\, \mathbf{y}_i^T\mathbf{c}_{T_m}^m)$ and & denotes the bit-wise operation *AND*. $\mathbf{w}_m$ is parameter vector of the concept conjunction classifier computed from Eq. (7).

Generally speaking, the presences of some concepts are usually much less than its absence. This situation is even worse for multiple concept conjunction learning since the positive sample must have multiple concepts simultaneously. To tackle the sample imbalance problem, we evolve our formulation with *data weighting* procedure inspired by Guillaumin et al. [10,30]. The resulting loss function is rewritten as the following weighted negative log-likelihood function:

$$\begin{aligned} L_w(\mathbf{x}_i, \mathbf{y}_i, \mathbf{C}^m; \Theta) = &-c_m^+\tilde{\mathbf{y}}_{im} log\left(\sigma(\mathbf{w}_m^T\mathbf{x}_i)\right) \\ &-c_m^-(1-\tilde{\mathbf{y}}_{im}) log\left(1-\sigma(\mathbf{w}_m^T\mathbf{x}_i)\right), \end{aligned} \tag{9}$$

where $c_m^+ = N/(2\times N_m^+)$ and $c_m^- = N/(2\times N_m^-)$ which make the loss weights of all the data sum up to N. N_m^+ (N_m^-) is the number of positive (negative) images for the mth multiple concept conjunction. The experimental results show that the weighted loss function performs better than the original logistic regression.

By combining the proposed C^{2}RNN and multiple concept conjunction classification into a unified framework, the final objective function is formulated in the following form:

$$\arg\min_{\Theta} \frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{M} L_*(x_i, y_i, C^m; \Theta) + \lambda\Omega(\Theta), \tag{10}$$

where $\Omega(\cdot)$ is the weight decay term used to increase the model generalization ability. The parameters λ is used to balance the relative influence of the regularization term. L_* denotes the loss function can be defined as Eq. (8) or its weighted version Eq. (9).

We solve the above optimization problem by using L-BFGS. The derivatives of the logistic regression parameters are calculated and the residual errors are back propagated into the output units of C^{2}RNN. Then the derivatives of Θ can be easily computed with the backpropagation through time algorithm [31]. In this way, our model can be trained in an end-to-end manner.

4. Experiments

We evaluate our method on two kinds of concept conjunction learning tasks: multi-attribute image retrieval and visual relationship detection. For multi-attribute image retrieval, we use three widely used datasets: aPascal [6], ImageNet Attributes [7] and LFWA [32]. Since the concept order is exchangeable, we use the order invariant version of C^{2}RNN. Then we compare it with the weighted loss to verify the effectiveness of tackling data imbalance. For visual relationship detection, we conduct experiments on the recently released VRD dataset [33] and use the multiple domain C^{2}RNN to learn the conjunction representation.

4.1. Datasets

aPASCAL. This dataset contains 6430 training images and 6355 testing images from Pascal VOC 2008 challenge. Each image comes from twenty object categories and is annotated with 64 binary attribute labels. We use the pre-defined test images for testing and randomly split ten percent images from training set for validation. The features we used for all the comparison methods are called DeCAF [34] which are extracted by the Convolutional Neural Networks (CNN). Since attributes are only defined for objects instead of the entire image, we use the object bounding box as the input of CNN.

ImageNet Attributes (INA). ImageNet Attribute dataset contains 9600 images from 384 categories. Each image is annotated with 25 attributes describing color, patterns, shape and texture. 3–4 workers are asked to provide a binary label indicating whether the object in the image contains the attribute or not. When there is no consensus among the workers, the attribute will be labeled as ambiguous for this image. The data with ambiguous attribute are not used for training and evaluating the corresponding conjunctions. We use {60%, 10%, 30%} random split for training/validation/test. And we also use DeCAF to do feature extraction.

LFWA. The labelled images on this dataset are selected from the widely used face dataset LFW [35]. It contains 5749 identities with totally 13,233 images. Each image is annotated with forty face attributes. Different from the above two datasets, LFWA gives a fine-grained category description. We use {60%, 10%, 30%} random split on the whole images for training/validation/test. We use VGG-Face descriptor [36] to extract feature for each image.

VRD. VRD dataset contains 5000 images with 4000 images for training and 1000 images for testing. The relationship instance is defined to be an object region appearing in the images. In total, there are 37,993 instances from 100 object categories. The number of predicates is 70. We use VGG16 [37] to extract the feature from the last fully connected layer for each object region. An input instance is a concatenation of the features from two paired object region.

Table 1
Valid multi-attribute queries.

Dataset	# of attributes	Double Queries	Triple Queries
aPascal	64	546	2224
INA	25	186	262
LFWA	40	771	9126

4.2. Experimental settings

For multi-attribute image retrieval, we generate the queries based on the dataset annotation to evaluate our methods. A query is considered to be valid when there are positive samples on train/validation and test simultaneously. We consider double and triple attribute queries for comparison. The detail information is shown in Table 1. To evaluate the performance for multi-attribute based image retrieval, we use the AUC (Area Under ROC) and AP (Average Precision) as the evaluation metric for each query. Since the number of attribute conjunctions is large, the resulting performance is hard to visualize for comparison. Therefore, we choose to use the mean AUC and mean AP to reflect the average performance of all the methods. We compare our approach with four baseline methods: TagProp [10], RMLL [11], MARR [12] and LIFT [15]. For TagProp, we use the logistic discriminant model with distance-based weights and the number of K nearest neighbours is chosen on the validation set. For RMLL and MARR, the loss function to be optimized is defined as Hamming loss which is also used in the original papers for retrieval tasks. Since TagProp, RMLL and LIFT do not support for multi-attribute query directly, we sum up the single attribute prediction scores as the confidence of multi-attribute query following the suggestion in [12]. The ratio parameter r of LIFT is set to be 0.1 as suggested in the paper.

For visual relationship detection, we first generate the object proposals and use each pair of the proposals as an input instance to further predict their predicate. Since the annotations of VRD is not complete, we choose to use Recall@50 and Recall@100 as the evaluation metrics following the original paper [33]. We compare our proposed method with Joint CNN, VP (Visual Appearance) and LP (Language Prior). For Joint CNN, the network contains three components: subject recognition, object recognition and predicate recognition. For VP and LP, the methods are based on [20] where VP only leverages visual feature and LP integrates the prior from a language model.

4.3. Multi-attribute image retrieval

We calculate the mean AUC and mean AP of double and triple attribute queries for all the comparison methods. The experimental results are shown in Tables 2 and 3 for double and triple attribute queries respectively. Comparing the results of RMLL and MARR, we can see that MARR surpasses RMLL on the different types of

Table 2
Experimental results for double attribute query.

Method	Eval	TapProp	RMLL	MARR	LIFT	C^{2}RNN
aPascal	mAUC	0.8807	0.8876	0.9040	0.8797	**0.9371**
	mAP	0.3361	0.3274	0.3336	0.3383	**0.3869**
INA	mAUC	0.8832	0.9166	0.8945	0.8902	**0.9450**
	mAP	0.2269	0.2126	0.1780	0.1953	**0.2605**
LFWA	mAUC	0.8113	0.8293	0.8210	0.8205	**0.8549**
	mAP	0.4075	0.4097	0.4209	**0.4372**	0.4370

Table 3
Experimental results for triple attribute query.

Method	Eval	TapProp	RMLL	MARR	LIFT	C^2RNN
aPascal	mAUC	0.8921	0.8988	0.9139	0.8910	**0.9360**
	mAP	0.2723	0.2497	0.2582	0.2640	**0.3034**
INA	mAUC	0.8927	0.9539	0.9375	0.9163	**0.9677**
	mAP	0.2001	0.1829	0.1375	0.1521	**0.2726**
LFWA	mAUC	0.8177	0.8367	0.8284	0.8247	**0.8665**
	mAP	0.2273	0.2218	0.2355	0.2473	**0.2499**

queries on aPascal and LFWA but fails on INA dataset. This is because the number of attribute on INA is too small and the correlation between them is not strong as aPascal and LFWA. So the performance of MARR decreases since this method relies on strong attribute correlation. From the results on all the three datasets, we can see that our method achieves better performance on all types of multi-attribute queries. Therefore recurrent neural network is beneficial for modelling the complex relationship of multiple attributes. We also conduct experiments comparing order invariant C^2RNN and pre-defined order version on the aPascal dataset. For double attribute query, pre-defined order C^2RNN achieves 0.9356 and 0.3758 according to mAUC and mAP respectively. This further demonstrates the effectiveness of order invariant C^2RNN.

For C^2RNN, the number of hidden units is chosen to be 100 for LFWA and 60 for aPascal and INA. For double attribute query, the value of lambda is 10^{-3} on INA and 10^{-2} on aPascal and LFWA. As for triple attribute query, the optimal values of lambda are 10^{-3}, 10^{-2} and 10^{-1} for INA, aPascal and LFWA respectively. And the optimal processing step for C^2RNN is two. Inspired by the tied weight strategy in pre-training autoencoder [38], we constrain $\mathbf{W}_v$ and $\mathbf{U}$ to share the parameters.

4.3.1. Data weighting strategy.

Images possessing all the query attributes are considered to be positive for training. Therefore, the positive samples are usually scarce when a query contains multiple attributes. To explicitly show this phenomenon, we calculate the positive sample ratio (N_m^+/N) for the queries in which the number of attributes ranges from one to three. For each type of the queries, we partition them into five parts according to the positive sample ratio and calculate the corresponding proportion for each part. The results are shown in the second row of Fig. 4. On all the three datasets, positive samples takes less than 10% for most of the double or triple queries. Therefore, how to solve data imbalance problem is essential for multi-attribute query based image retrieval. Then we train the proposed models by using the loss functions defined in Eqs. (8) and (9) on aPascal respectively. The performance of the two versions with and without using data weighting for C^2RNN are shown in Table 4. From the results, we can see the method using data weighting procedure consistently performs better than the original version when the positive data is imbalance.

4.3.2. Attribute embedding

In this section, we validate the quality of the learned embedding matrix. We first calculate the ground truth correlation matrix which can reflect the correlation information between attributes.

Table 4
Comparison C^2RNN with its weighted version on the aPascal dataset.

Dataset	Single Queries	Double Queries	Triple Queries
C^2RNN	0.9310	0.9327	0.9285
C^2RNN-Weighted	0.9318	0.9371	0.9360

Let $\mathbf{R} \in \mathbb{R}^{A\times A}$ be the correlation matrix, where the correlation score between attribute i and j is computed following [39]:

$$R_{i,j} = \frac{\mathbf{Y}_{i,:}^T \mathbf{Y}_{j,:}}{\mathbf{Y}_{i,:}^T \mathbf{1} + \mathbf{Y}_{j,:}^T \mathbf{1} - \mathbf{Y}_{i,:}^T \mathbf{Y}_{j,:}}. \tag{11}$$

From the definition, we can see that two attributes are strongly correlated if they have a large number of images in common. Intuitively, the attribute embedding learned by C^2RNN is expected to reflect the correlation between attributes. So we visualize the similarity matrix of the learned attribute embeddings on all the three datasets in Fig. 5. The similarity score for a pair of attributes is calculated by using their cosine distance. Comparing the ground truth correlation matrix and the learned similarity matrix, we can see most of the correlated attributes are close on the embedding space.

4.3.3. Attention mechanism

We validate the effectiveness of learning attribute conjunction with a recurrent attention vector in this section. Since the learned conjunction is order invariant, we can tackle the task with no predefined concept dependence. Moreover, the concepts not appearing in the current conjunction can also be used to generate the final representation of concept conjunction if they are correlated. We visualize some of the attention vectors learned by our method for both double and triple attribute queries on LFWA dataset. As shown in Fig. 6, the query contains “Chubby” will also take “Double Chin” into attention to generate the representation of attribute conjunction and an “Attractive” person with “Bushy Eyebrows” is probably “Male”.

4.4. Visual relationship detection

In this section, we validate the proposed method on visual relationship detection. For an input image, the task is to learn a concept conjunction $\{object_1, \ predicate, \ object_2\}$. We measure our proposed method on two conditions: Predicate detection and relationship detection. For Predicate detection, the input is an image and a set of objects. Our model need to predict the predicate between any two paired objects. For relationship detection, we need to detect all the objects in the image and then recognize their predicates. For fair comparison, we use the same detection results as [33]. The detected object bounding box is correct if it has at least 0.5 overlap with the ground truth bounding box. As shown in Table 5, our proposed framework achieves much better performance compared with Joint CNN and VP. Since our proposed method uses only visual feature as Joint CNN and VP, the performance gain comes from two parts. Firstly, the proposed C^2RNN effectively models the relationship within each single domain (object domain and predicate domain). Secondly, it also combine the

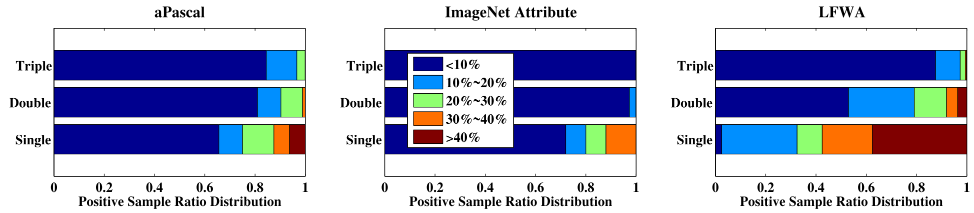

Fig. 4. The positive sample ratio distribution on the three datasets for multi-attribute based image retrieval.

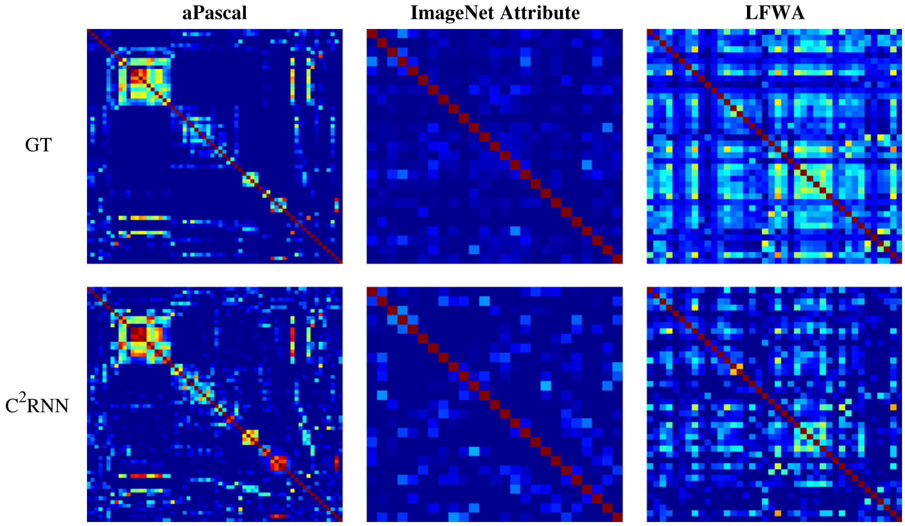

Fig. 5. Concept similarity matrix on the embedding space.

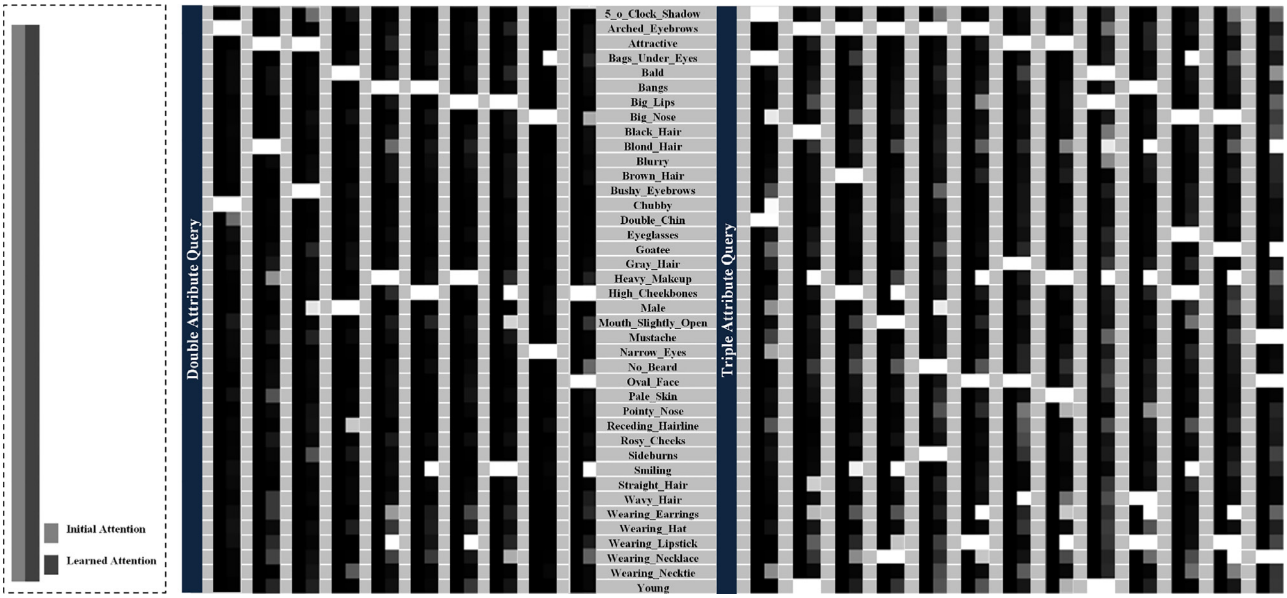

Fig. 6. Attention vector visualization on LFWA dataset.

Table 5
Valid multi-attribute queries.

	Relationship Detection		Predicate Detection	
	Recall@50	Recall@100	Recall@50	Recall@100
Joint CNN [37]	0.07	0.09	1.47	2.03
VP [20]	1.58	1.85	7.11	7.11
LP [20]	13.86	14.70	**47.87**	47.87
C²RNN	**16.01**	**18.50**	40.98	**52.11**

two domains representation properly in a multiplicative way. Even without using the language prior, our proposed method still outperforms LP on visual relationship detection.

5. Conclusion

We propose a new type of recurrent neural network for learning multiple concept conjunction. Different from previous methods, our model explicitly learns the concept embedding and generates the representation of concept conjunction by recurrently combining the learned concept embeddings. Based on that, we present an order invariant version based on attention mechanism for learning the task without pre-defined concept order. In addition, we propose a variant of our method using data weighting strategy to mitigate the data imbalance problem. Finally, we propose an effective way for learning the conjunction from multiple semantic domains. We conduct experiments on two tasks: multi-attribute image retrieval and visual relationship detection. Experimental results show the significant improvement over the other comparison methods on both tasks.

Declarations of interest

None.

Acknowledgments

This work is partially supported by 973 Program under contract No. 2015CB351802, and Natural Science Foundation of China under contracts Nos. 61390515.

References

[1] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[2] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149.
[3] Y. Wang, G. Mori, A discriminative latent model of object classes and attributes, in: Proceedings of the Computer Vision–ECCV, Springer, 2010, pp. 155–168.
[4] N. Kumar, A.C. Berg, P.N. Belhumeur, S.K. Nayar, Attribute and simile classifiers for face verification, in: Proceedings of the IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 365–372.
[5] C.H. Lampert, H. Nickisch, S. Harmeling, Attribute-based classification for zero-shot visual object categorization, IEEE Trans. Pattern Anal. Mach. Intell. 36 (3) (2014) 453–465.
[6] A. Farhadi, I. Endres, D. Hoiem, D. Forsyth, Describing objects by their attributes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR, IEEE, 2009, pp. 1778–1785.
[7] O. Russakovsky, L. Fei-Fei, Attribute learning in large-scale datasets, in: Proceedings of the International Workshop on Parts and Attributes European Conference of Computer Vision (ECCV), 2010.
[8] N. Kumar, P. Belhumeur, S. Nayar, Facetracer: a search engine for large collections of images with faces, in: Proceedings of the Computer Vision–ECCV 2008, Springer, 2008, pp. 340–353.
[9] A. Kovashka, D. Parikh, K. Grauman, Whittlesearch: Image search with relative attribute feedback, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 2973–2980.
[10] M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid, Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation, in: Proceedings of the IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 309–316.
[11] J. Petterson, T.S. Caetano, Reverse multi-label learning, in: Proceedings of the Advances in Neural Information Processing Systems, 2010, pp. 1912–1920.
[12] B. Siddiquie, R.S. Feris, L.S. Davis, Image ranking and retrieval based on multi--attribute queries, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 801–808.
[13] M. Rastegari, A. Diba, D. Parikh, A. Farhadi, Multi-attribute queries: To merge or not to merge? in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 3310–3317.
[14] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, Mach. Learn. 85 (3) (2011) 333–359.
[15] M.-L. Zhang, L. Wu, Lift: multi-label learning with label-specific features, IEEE Trans. Pattern Anal. Mach. Intell. 37 (1) (2015) 107–120.
[16] D.R. Hardoon, S. Szedmak, J. Shawe-Taylor, Canonical correlation analysis: an overview with application to learning methods, Neural Comput. 16 (12) (2004) 2639–2664.
[17] Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid, Label-embedding for attribute-based classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 819–826.
[18] A. Frome, G.S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al., Devise: a deep visual-semantic embedding model, in: Proceedings of the Advances in Neural Information Processing Systems, 2013, pp. 2121–2129.
[19] S.J. Hwang, L. Sigal, A unified semantic embedding: relating taxonomies and attributes, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 271–279.
[20] C. Lu, R. Krishna, M. Bernstein, L. Fei-Fei, Visual relationship detection with language priors, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 852–869.
[21] H. Zhang, Z. Kyaw, S.-F. Chang, T.-S. Chua, Visual translation embedding network for visual relation detection, in: Proceedings of the IEEE Conference on Computer vision and Pattern Recognition, 2017.
[22] X. Liang, L. Lee, E.P. Xing, Deep variation-structured reinforcement learning for visual relationship and attribute detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[23] Y. Li, W. Ouyang, X. Wang, VIP-CNN: A visual phrase reasoning convolutional neural network for visual relationship detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[24] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: neural image caption generation with visual attention, in: Proceedings of the 32nd International Conference on Machine Learning, 2015, pp. 2048–2057.
[25] A. Graves, Supervised Sequence Labelling, Springer, 2012.
[26] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, X. Xie, et al., Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks., in: Proceedings of the AAAI, 2, 2016, p. 8.
[27] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, Proceedings of the International Conference on Learning Representations (2015).
[28] O. Vinyals, S. Bengio, M. Kudlur, Order matters: sequence to sequence for sets, in: Proceedings of the 4th International Conference on Learning Representations, 2016.
[29] K. Liang, H. Chang, S. Shan, X. Chen, A unified multiplicative framework for attribute learning, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2506–2514.
[30] G. King, L. Zeng, Logistic regression in rare events data, Pol. Anal. 9 (2) (2001) 137–163.
[31] P. Werbos, Backpropagation through time: what it does and how to do it, Proc. IEEE 78 (10) (1990) 1550–1560, doi:10.1109/5.58337.
[32] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3730–3738.
[33] C. Lu, R. Krishna, M. Bernstein, L. Fei-Fei, Visual relationship detection with language priors, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 852–869.
[34] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DECAF: a deep convolutional activation feature for generic visual recognition, in: Proceedings of The 31st International Conference on Machine Learning, 2014, pp. 647–655.
[35] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: a database for studying face recognition in unconstrained environments, Technical Report, 2007.
[36] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, in: Proceedings of the British Machine Vision Conference, 2015.
[37] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
[38] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.
[39] B. Sigurbjörnsson, R. Van Zwol, Flickr tag recommendation based on collective knowledge, in: Proceedings of the 17th International Conference on World Wide Web, ACM, 2008, pp. 327–336.

Kongming Liang received the Bachelor's degree from China University of Mining & Technology-Beijing, China, in 2012; Currently, he is a Ph.D candidate in Institute of Computing Technology, Chinese Academy of Science since 2012. From Sep 2016 to Oct 2017, he was a joint Ph.D. Student of machine learning group in Carleton University, Canada. His research interests cover computer vision and machine learning, especially visual attribute learning and holistic image understanding based on deep neural networks.

Hong Chang received the Bachelor's degree from Hebei University of Technology, Tianjin, China, in 1998; the M.S. degree from Tianjin University, Tianjin, in 2001; and the Ph.D. degree from Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 2006, all in computer science. She was a Research Scientist with Xerox Research Centre Europe. She is currently an Associate Researcher with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. Her main research interests include algorithms and models in machine learning, and their applications in pattern recognition, computer vision, and data mining.

Shiguang Shan received M.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1999, and Ph.D. degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2004. He joined ICT, CAS in 2002 and has been a Professor since 2010. He is now the deputy director of the Key Lab of Intelligent Information Processing of CAS. His research interests cover computer vision, pattern recognition, and machine learning. He especially focuses on face recognition related research topics. He has published more than 200 papers in refereed journals and proceedings in the areas of computer vision and pattern recognition. He has served as Area Chair for many international conferences including ICCV'11, ICPR'12, ACCV'12, FG'13, ICPR'14, ICASSP'14, and ACCV'16. He is Associate Editors of several international journals including IEEE Trans. on Image Processing, Computer Vision and Image Understanding, Neurocomputing, and Pattern Recognition Letters. He is a recipient of the China's State Natural Science Award in 2015, and the China's State S&T Progress Award in 2005 for his research work.

Xilin Chen received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1988, 1991, and 1994, respectively, where he was a Professor from 1999–2005. He has been a Professor with the Institute of Computing Technology, Chinese Academy of Sciences (CAS), since 2004. He is also a FiDiPro Professor from 2012–2015 in Oulu University. He has authored one book and over 200 papers in refereed journals and proceedings in the areas of computer vision, pattern recognition, image processing, and multimodal interfaces. He is a fellow of the China Computer Federation.