Boosting Attribute Recognition with Latent Topics by Matrix Factorization

Donghui Li
National Engineering Research Center of Digital Life, State-Province Joint Laboratory of Digital Home Interactive Applications, School of Data and Computer Science, Sun Yat-sen University, Guangzhou, 510006, China. E-mail: lidh6@mail2.sysu.edu.cn

Zhuo Su
School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China (Corresponding Author, E-mail: suzhuo3@mail.sysu.edu.cn)

Hanhui Li
National Engineering Research Center of Digital Life, State-Province Joint Laboratory of Digital Home Interactive Applications, School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. E-mail: lihanhui@mail2.sysu.edu.cn

Xiaonan Luo
National Engineering Research Center of Digital Life, State-Province Joint Laboratory of Digital Home Interactive Applications, Sun Yat-sen University, Guangzhou 510006, China. E-mail: lnslxn@mail.sysu.edu.cn

Attribute-based approaches have attracted lots of attention in visual recognition tasks recently. These approaches describe images by using semantic attributes as the mid-level feature. However, the low recognition accuracy becomes the biggest barrier that limits their practical applications. In this paper, we propose a novel framework named Boosting Attribute Recognition (BAR) in still image recognition task. Inspired by matrix factorization, our framework simultaneously explores latent relationships from the aspect of attribute and image-level. Furthermore, in order to apply our framework in large-scale visual recognition tasks, we present both offline and online learning implementation within the framework. Extensive experiments on three datasets demonstrate that our framework achieves a sound accuracy of attribute prediction in both human and animal attribute recognition tasks.

Introduction

Image recognition, as an essential stage for constructing image search engine, has been explored over the last decades, including product image search engine (Li, Xu, Luo & Lin, 2014). Lots of effective solutions have been proposed. However, some state-of-the-art models (Sharma & Jurie, 2011) just emphasize on the relationships between low-level vision cues (shape, contour, or more complex structures) and categories of objects (for humans: male and female, old and young; for animals: dog, cat, lion, and so on). Obviously, the semantic meanings of objects are ignored, which are significant to distinguish categories of objects. In recent years, since attribute-based classification models have effective geometric expression and semantic description (Bourdev, Maji, & Malik, 2011; Bourdev, Maji, & Malik, 2011; Farhadi, Endres, Hoiem, & Forsyth, 2009; Russakovsky & Li, 2010), they are introduced into the image classification problem (Lampert, Nickisch, & Harmeling, 2014).

Image attributes are human-named properties of objects. These semantic attributes could be parts of objects (“has paw”, “crossed arms”), shapes of objects (“lean”, “sitting”), materials of objects (“furry”, “jeans”), or discriminative descriptions of objects (“birds have feet but fishes do not”, “women wear skirts but men do not”), even descriptions of unseen characteristics of objects (“smelly”, “sensitive”). In a word, attributes are human-defined properties of objects, which could be nouns, adjectives, adverbs and so on.

However, due to the complexity of attributes, how to boost the attribute prediction accuracy becomes a challenging problem in the attribute-based computer vision tasks. In this paper, we propose a novel framework named Boosting Attribute Recognition (BAR) to boost the accuracy of attribute prediction. BAR explores the relationships among instances, attributes and latent factors by utilizing matrix factorization methods. The main contributions of this paper are as follows:

- We propose a framework to discover potential relationships among instances and attributes in a "reduced" joint latent factor space via matrix factorization. Then we utilize the discovered relationships to boost attribute recognition.

- Both the online and offline learning version of our framework is proposed to satisfy practical requirements. Offline learning is for the circumstance that all training data for BAR are available at the beginning, while online learning is for the situation that the training data become available in a sequential fashion.

In the following, we will review some representative work with respect to attribute-based image classification and matrix factorization.

Related Work

Attribute-Based Image Classification

Attributes were first introduced into image classification tasks in 2009 (Farhadi, Endres, Hoiem, & Forsyth, 2009; Lampert, Nickisch, & Harmeling, 2009). In recent years, relative attributes (Parikh & Grauman, 2011) are studied to enable richer textual descriptions, which are more precise for human interpretation. Besides, relative attributes are used to improve the retrieval results in various image search engines (Kovashka, Parikh, & Grauman, 2012; Wang & Zhang, 2011). One of the most valuable properties of introducing attributes of images is to tackle the problem of object classification in the case of zero-shot learning, when training and test classes are disjoint (Lampert, Nickisch, & Harmeling, 2009; Suzuki, Sato, Oyama, & Kurihara, 2014; Chen, Gallagher, & Girod, 2012). Especially, in the case of one-shot learning, it usually just provides none or a few labeled training samples at the test phrase (Guo & Wang, 2013).

Since attribute learning is significant to high-level tasks, exploring relationships among attributes have attracted more and more attentions in recent years (Lampert, Nickisch, & Harmeling, 2014; Sharma & Jurie, 2011; Li, Shan, & Chen, 2012; Turakhia & Parikh, 2013; Han, Wu, Lu et al., 2012; Han, Yang, Ma, et al., 2014). Lampert et al. proposed two methods to solve the zero-shot learning problem by using an attribute layer to transfer information between classes (Lampert, Nickisch, & Harmeling, 2014). Sharma et al. proposed an efficient algorithm to learn the spatial partitioning with bag-of-features representation for image classification tasks (Sharma & Jurie, 2011). However, how to make an instance establish a relation to other instances and what are the relations between attributes and instances have not been studied in Lampert’s and Sharma’s work. These defects make their models hard to obtain a sound attribute prediction accuracy. Li et al. proposed the relative forest algorithm to achieve more accurate attribute prediction (Li, Shan, & Chen, 2012), inheriting the idea of learning ranking function for each attribute (Parikh & Grauman, 2011).

Song et al. used a high dimensional vector to represent each attribute in the subject space, and exploited the relationships among attributes in that space to improve face verification (Song, Tan, & Chen, 2012). Practically, lots of vision tasks would benefit from the attribute-based recognition approaches (Farhadi, Endres, Hoiem, & Forsyth, 2009; Lampert, Nickisch, & Harmeling, 2009; Yu & Aloimonos, 2010). Binary attributes are general in lots of cases, such as “has pads”, “has glasses” and “has tail”.

Matrix Factorization

Matrix factorization could be utilized to boost the accuracy of attribute recognition via finding the latent factors. Mathematically, matrix factorization reformulates the matrix with the product of multiple matrices which have simpler structures or more familiar characters.

There are kinds of matrix factorization approaches, such as Triangular Factorization (TF) (Taussky & Todd, 2006), QR Factorization (Chu & George, 1987) and Singular Value Decomposition (SVD) (Kalman, 1996). A series of new factorization approaches were also presented in the last decades, such as Non-negative Matrix Factorization (NMF) (Lee & Seung, 2001), Latent Factor Model (LFM) (Koren, Bell, & Volinsky, 2009), and so forth. Recently, matrix factorization was applied to recommendation system to realize the latent factor models between users and items (Koren, Bell, & Volinsky, 2009). In its basic form, matrix factorization formulates both items and users by the vectors of factors derived from the item rating patterns. As the Netflix Prize competition (Koren, Bell, & Volinsky, 2009); Takács, Pilászy, Németh, & Tikk, 2008) demonstrated, matrix factorization approaches are superior to classic nearest neighbor approaches for product recommendations, as it allows the incorporation of additional information such as implicit feedback, temporal effects, and confidence levels. In order to populate a database of a universal schema, Riedel et al. applied matrix factorization to learn latent feature vectors for entity tuples and relations (Riedel, Yao, McCallum, & Marlin, 2013). The experiments showed that these latent models could achieve higher accuracy than traditional classification approaches.

As for SVD, it conveys important geometrical and theoretical insights about linear transformations, due to its interesting and attractive algebraic properties (Kalman, 1996). In recent years, SVD has been broadly applied to computer vision and recommendation systems (Zhang & Li, 2010; Sali, 2008; Rajwade, Rangarajan, & Banerjee, 2013). In the face recognition system, Zhang et al. utilized discriminative K-SVD for dictionary learning (Zhang & Li, 2010). And in the recommendation system, Sali took advantage of SVD for predicting movie ratings (Sali, 2008).

As for NMF, a matrix could be regarded as a non-negative matrix if all entries of the matrix are non-negative. NMF has been demonstrated to be able to learn parts of faces and semantic features of text (Lee & Seung, 1999).

Preliminary

In this section, we will first introduce joint latent factor space, SVD and NMF for establishing our framework. Then we will elaborate how to collect the predicted attribute values as an instance-attribute matrix, which is the baseline algorithm of this paper.

Joint Latent Factor Space

Intuitively, each image is related to some topics, such as “land living animals”, “aquatic animals”. Likewise, each attribute is associated with some topics, such as “living environment”, “living habitats”. Some topics could be shared between instances and attributes. We expect to discover and utilize the relationships among instances, attributes and those topics to boost attribute recognition. Inspired by the latent factor model (Skrondal & Rabe-Hesketh, 2007), we want to generate a joint latent factor space where both instances and attributes can be characterized. In the space, each semantic latent factor represents a topic. We expect to model the relationships among instances, attributes and latent factors well as the locations of instances and attributes in that space. As illustrated in Figure 1, a two-dimensional joint latent factor space describes the distributions of instances and attributes, and shows us the relationships among instances, attributes and latent factors.

Once the joint latent factor space is generated, we can easily find a nearest neighbor-hood from training data for each test image in that space. Then we can use the ground truth attribute labels of each test image’s own nearest neighbor-hood to refine its attribute prediction.

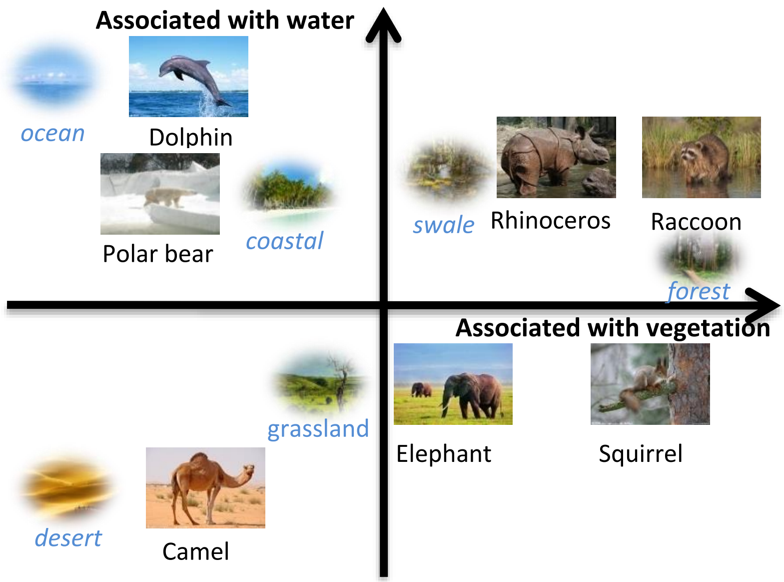


FIG. 1. A simple example of latent factor approach. Assume that we are interested in two topics, namely two latent factors: "Associated with vegetation" and "Associated with water". The approach projects instances and attributes into the same joint latent factor space. We expect similar instances or attributes are near to each other in that space. In other words, we want to discover a space where the probability for an instance having a certain attribute is nearly proportional to the dot product of the instance's and the attribute's locations in that space. For example, according to the locations of instances and attributes, attribute "ocean" belongs to a dolphin while attribute "desert" and "forest" do not. Similarly, we expect a hippopotamus to have attribute "swale" but not attribute "coastal" or "desert".

For attributes, the latent factors could: (1) measure obvious dimensions such as the relations between animal and water, or the greening degrees of the attributes; (2) measure less well-defined dimensions such as mutual exclusion of attributes; (3) measure completely uninterpretable dimensions. For instances, each factor measures the probability of the instance having the attributes that score high on the corresponding attribute factor.

Denote the factors set as D, such that the dimensionality of the joint latent factor space is $|D|$. Accordingly, each instance i is associated with a vector $A_i \in R^{|D|}$, each attribute j is associated with a vector $B_j \in R^{|D|}$. For a given instance i, the elements of A_i measure the extent to which the instance possesses those factors, and it could be positive or negative. This is same as B_j with a given attribute j. The probability P_{ij} for an instance i to have an attribute j is modeled as their inner product in that space. In other words, the resulting dot product approximates the probability of an attribute j belonging to an instance i as $P_{ij} = A_i B_j^T$.

Let G be the set of pairs (A_i, B_j), P'_{ij} denotes the ground truth probability for instance i to have an attribute j. In order to compute the mapping of each pair with regard to instance and attribute to the latent factor vectors $(A_i, B_j) \in R^{|D|}$, we need to solve the following optimization problem by minimizing the regularized squared error on the set of known probabilities,

$$\min_{A^*, B^*} \sum_{(i,j)\in G} \left(P'_{ij} - A_i B_j^T\right) + \lambda(\| A_i \|^2 + \| B_j \|^2). \tag{1}$$

Here, A^* and B^* are the latent factor vectors that we need to map each pair (A_i^*, B_j^*) into the latent factor space, and λ is the parameter for regularization.

Actually, the latent factor vectors do not require high accuracy evaluation. The reason is that some factors are redundant and useless for exploring relationships, and we just need to discover and utilize the most significant ones. Thus, we utilize matrix factorization to find out suitable latent factor models, meanwhile sort the energies of them in descendant order. That is, the abovementioned joint latent factor space that we need is generated by matrix factorization methods. This space allows us to obtain a better accuracy of attribute recognition by exploring relationships among instances, attributes and latent factors. In order to further reduce the redundant and useless latent factors, some significant candidates are selected according to some constraints to form a new "reduced" joint latent factor space.

Singular Value Decomposition

Singular Value Decomposition (SVD) (Kalman, 1996) is a well-established matrix factorization model for identifying latent semantic factors. Given an arbitrary matrix $P \in R^{n\times m}$, SVD produces two orthogonal matrices $U \in R^{n\times n}$ and $V \in R^{m\times m}$, along with a diagonal matrix $S \in R^{n\times m}$, such that

$$P = USV^T, \tag{2}$$

where S is rectangular with the same size of P and its diagonal entries ($S_{ii} = \sigma_i$) are in descending order. The singular values of P mean the positive ones of S, and the left and right singular vectors of P mean the columns of U and V, respectively. The corresponding relationships of Eq. (2) with instances/attributes/latent factors are as follows:

Let P be the instance-attribute matrix. Row i of P is instance i, column j of P means an attribute j, and the element P_{ij} means the predicted probability value of attribute j for instance i. By factorizing P into three small matrices, the latent factor models (which indicate the relationships among instances, attributes and topics) of P, could be discovered. In descendant order, diagonal entries σ_i of matrix S are the energies of latent factor models (topics). Each row of U corresponds to an instance, and the ith row vector of U shows how the ith instance is mapped to the joint latent factor space (topic space). Each row of V means an attribute and the jth row vector of V shows the location of the jth attribute in the latent factor space.

Non-negative Matrix Factorization

Non-negative Matrix Factorization (NMF) is different from other matrix factorization methods with non-negativity constraints. It allows additive combinations rather than subtractive. Since the whole predicted attribute probabilities are non-negative, it is feasible to incorporate NMF into our framework. Given a non-negative matrix P, NMF finds two non-negative matrix factors W and H such that

$$P \approx WH. \tag{3}$$

As aforementioned, only non-negative entries exist in matrix W and H. If P is an $n \times m$ matrix, then W and H could be regarded as $n \times r$ and $r \times m$ matrices, respectively. Generally, r is smaller than n or m, so that W and H are smaller than original matrix P, resulting in a compressed version of the original data. Similar to SVD, NMF discovers the latent factor models of P, which indicates the relationships among instances, attributes and topics by factorizing P into two small matrices. Each row of W means an instance, and each column of H means an attribute.

The main difference between SVD and NMF is that the entries of W and H are non-negative. The non-negative constraints of W and H make an object be decomposed into a non-negative linear combination of its different parts. In addition, both W and H possess special physical meanings to distinguish NMF from other decomposition approaches. W is treated as a set composed of r vectors, and the vectors of W form a linear space. Each column of H is an approximation of the projection of the corresponding column of P in that space.

Baseline Algorithm

The baseline algorithm here means that we use traditional manipulation steps of attribute-based image classification task to perform attribute prediction. Given a training set $(x_1, x_2, \dots, x_{N_1}) \in X_{N_1}$ of size N_1, a test set $(x_1, x_2, \dots, x_{N_2}) \in X_{N_2}$ of size N_2, and an attributes space $(a_1, a_2, \dots, a_M) \in A^M$, M attribute classifiers $(f_1, f_2, \dots, f_M): X^d \to A^M$ are defined to map test images into the attributes space, where d is the dimension of low-level features of images. More specifically, we first extract the low-level features of each image. And different kinds of low-level features are concatenated together as a vector to represent the image. Then for each attribute, we use its positive samples (images that present the attribute) and negative samples (images that do not present the attribute) to train a classifier.

Next, we use each of the trained attribute classifiers to predict whether an image presents that attribute or not. Thus, we can use the predicted existing attribute probabilities vector to represent the images.

Note that, when attribute classifiers are trained, we use them to predict existing probabilities of attributes in images from both test set (X_{N_2}) and training set (X_{N_1}), as shown in Figure 2. Let $I \in R^{N \times M}$ be the original predicted instance-attribute matrix. I is generated from the baseline algorithm with $N = N_1 + N_2$. Row n of I is instance n, column m of I means attribute m, and element I_{nm} means the predicted existing probability of attribute m for instance n.

In the next section, we will describe how to incorporate SVD and NMF models to achieve better accuracy of attribute prediction.

Methods

Now we will explicitly illustrate how to incorporate SVD and NMF into the attribute prediction step of attribute-based work. The pipeline of our framework is shown in Figure 2.

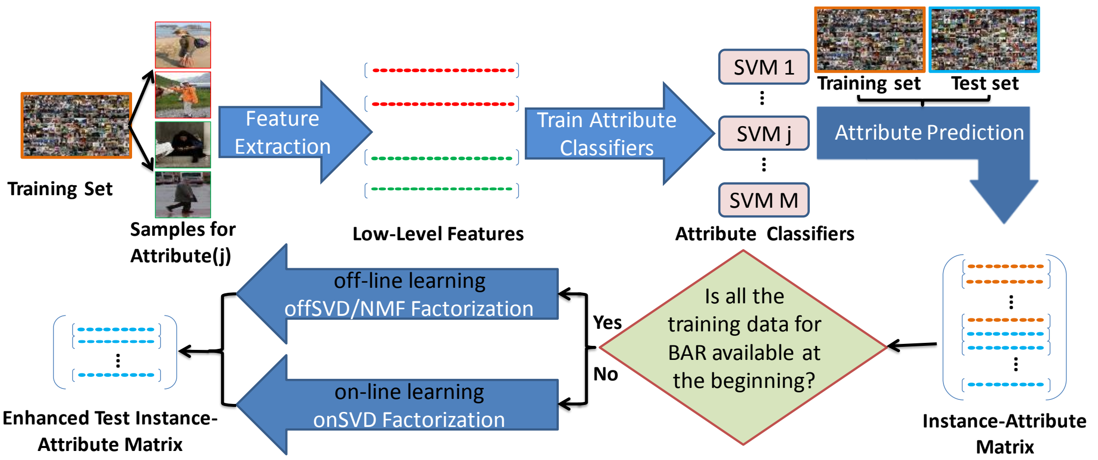


FIG. 2. The pipeline of our framework. Given training images, we first extract their low-level image features. Then a SVM classifier is trained for each attribute by using the extracted features. After performing attribute prediction step, we utilize SVD and NMF matrix factorization on instance-attribute matrix to boost attribute recognition. In the framework, offline learning is established for the circumstance that all training data for BAR are available at the beginning, and online learning is for the circumstance that the training data come in a sequential mode.

Overview of Our Model

To boost the accuracy of attribute prediction, we focus on discovering and utilizing the potential relationships of instances and those of attributes after the attribute classifiers have given attribute prediction for test data. Considering whether all training instances are available at the test phrase, we design two learning modes of the proposed framework: One is offline learning, which means all training data are available at the beginning. We generate a "reduced" joint latent factor space by the traditional SVD model (Kalman, 1996) and the NMF decomposition model (Lee & Seung, 2001). The other is online learning, which means retraining the models when training data come in sequence. Online learning is necessary because sometimes the scale of training data is too large to be stored in the memory. What is more, due to the limitation of application environment, not all training data are available at the beginning. For online learning, the "reduced" joint latent factor space is generated by a new updating SVD model (Berry, Dumais, & O'Brien, 1995).

At the test phase of BAR, given a test instance, the training set allows us to find a certain number of neighbors whose attribute patterns are similar to those of the given instance. Here, both training and test instances are represented in the "reduced" joint latent factor space. Enhanced attribute prediction for each test instance is made by using the ground truth attribute labels of its own neighbors.

Enhancing Attribute Prediction within Offline Learning

It is appropriate to use offline learning approach when all training data and test data are available at the beginning. After using the trained SVM attribute classifiers to perform attribute prediction on both training images and test images, we'll get the instance-attribute matrix I. Then SVD and NMF are utilized to complete the offline learning of BAR framework. In the following, we refer to these two approaches as BAR-offSVD and BAR-offNMF, respectively.

Implementation of SVD within offline learning. The first step of BAR-offSVD is to decompose matrix I by SVD and obtain three matrices which satisfy

$$I = USV^T. \tag{4}$$

In Eq. (4), matrix $U \in R^{N\times N}$ tells us how instances are related to latent factors (topics), matrix $V \in R^{M\times M}$ demonstrates relationships between attributes and latent factors. And matrix $S \in R^{N\times M}$ shows us the energies of corresponding latent factors. For example, a running old man is more like a walking old man than a sitting young girl. Similarly, attribute "long skirt" is more related with attribute "short skirt" than attribute "men suit". In order to remove the useless and redundant latent factors, we select k diagonal entries of maximum values from matrix S to obtain $S_k \in R^{k\times k}$. The value of k is determined by restraining S_k to keeps more than 95% information of S. Correspondingly, we get the reduced matrices $U_k \in R^{N\times k}$ and $V_k \in R^{k\times M}$. Denote the "reduced" I as I_k, which is an approximation of I and expressed as

$$I_k = U_k S_k V_k^T. \tag{5}$$

Now instances and attributes are mapped to the "reduced" joint latent factor space by row vectors of U_k and V_k, respectively.

The intuition behind reduction procedure is that some latent factors are redundant for characterizing both of instances and attributes. In Figure 3, assume that there are only four person images (the rectangular ones) and four person attributes (texts below those four images). Obviously, only two latent factors "Gender" and "Temperature" can project the four instances and the four attributes into the same joint latent factor space perfectly. In other words, other latent factors such as "Person's tastes" or "Reading hobbies" are redundant for characterizing instances and attributes in this case. Thus, after utilizing matrix factorization to discover latent factors, it is necessary to perform reduction procedure to remove the redundant ones. Note that instances and attributes marked in yellow are independent of these two factors, which means the latent factors between particular instances and attributes are specific. Hence we need to utilize matrix factorization methods to discover the specific latent factors. In the following, we denote the joint latent factor space without reduction procedure as "full" space.

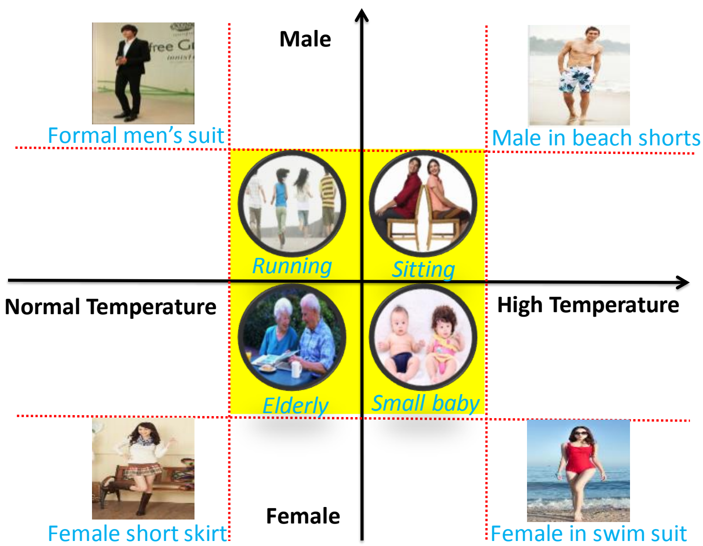


FIG. 3. An example to illustrate the intuition behind reduction procedure.

The second step of BAR-offSVD is to enhance attribute prediction of test images. Note that, any learning-based classification technique could be used to achieve this goal. We prefer to use K Nearest Neighbors (KNN) algorithm because it is easy to apply and bring out the effectiveness of our latent factor approach. Definition of “distance” depends on the representation of instances in the “reduced” joint latent factor space, rather than the low-level features representation or the predicted attributes probabilities representation. We build up a k-d tree to store the reduced instance-attribute matrix. This data structure allows us to access to each instance and its neighbors in the “reduced” joint latent factor space effectively. For each instance of test set, we refine its attribute prediction result by using majority voting of its neighbors from the training set.

Implementation of NMF within offline learning. Similar to BAR-offSVD, the second implementation of offline learning is to utilize NMF model in BAR framework, which we refer to as BAR-offNMF. For each instance of test set, BAR-offNMF also finds the corresponding instance-neighbor in the “reduced” joint latent factor space to get its enhanced attribute prediction. BAR-offNMF is distinguished from BAR-offSVD in two steps. One is the factorization step and the other is the finding instance-neighborhood step. In the first one, BAR-offNMF decomposes matrix I by NMF to obtain two non-negative entries factors W and H, which satisfy

$$I = WH, \tag{6}$$

where $W \in R^{N\times r}$, $H \in R^{r\times N}$, and r is the selected latent factor dimensionality for W and H. We choose an optimal r through a series of experiments. In the second one, finding the neighbors for instances of test set is performed on W space instead of U_k space.

Enhancing Attribute Prediction within Online Learning

Offline learning provides a feasible way when the whole data are available at the beginning. However, it is not convenient to deal with big scale data due to computer memory limit. What is more, in the real environment it is usually hard to get all data at the beginning. On the contrary, they are available in a sequential mode. Thus, an alternate choice is to consider online learning for the BAR framework.

Under the online learning circumstance, only parts of training data of the baseline algorithm are available at the beginning. In other words, training data of BAR become available in a sequential mode. For the offline learning, we apply traditional SVD and NMF on the instance-attribute matrix, which is composed of training data and test data of the baseline algorithm. Thus, when new training data or test data of the baseline algorithm become available, the traditional SVD and NMF solutions are not applicable any more. To address this problem, we apply an updating SVD approach introduced by Berry (Berry, Dumais, & O’Brien, 1995). Denote the new training data of BAR as $F \in R^{Q\times M}$, that is Q new training/test instances of baseline algorithm with all M predicted attributes probabilities. To fold-in the vector F into an existing “reduced” joint latent factor space of Eq. (5), a projection F' of F onto the span of the current attributes vectors (rows of V_k) is determined by

$$F' = FV_kS_k^{-1}. \tag{7}$$

At the first period of BAR, not all data are available at the beginning. We perform SVD operation to the instance-attribute matrix composed of the known data to get a “reduced” joint latent factor space. Then project the new data into that space by applying Eq. (7). This procedure allows us to complete the online learning mode of BAR framework. The rest processes are the same as those of offline learning. That is, for each instance of test data of baseline algorithm, we find a certain number of neighbors from both the known and the new training data of baseline algorithm. Enhanced attribute prediction for that instance is made by using the ground truth attribute labels of its own neighbors.

Controlling the Scale of Training Set

Sometimes a large scale dataset is available, however we do not need that many images for training, limited to computer memory size and training time. Therefore, we control the scale of training data of BAR in two ways. One is based on Self-Organizing and Incremental Neural Networks (SOINN) algorithm (Furao & Hasegawa, 2006; Shen & Hasegawa, 2010) to reduce the scale of training data. SOINN is an unsupervised online learning approach, which has a similar function as clustering methods. It models data distribution in the form of a network where nodes represent data and edges represent relationships between nodes. The output of SOINN algorithm is a new dataset, which is different from the input dataset. However, when reducing the scale of training data, the output should be a subset of the input dataset, which means the output images are the representative ones of input images. Hence, we have modified the SOINN algorithm to meet our requirement. By performing the modified SOINN on the large scale training set, we can achieve a much smaller training set with compact structure.

The other way to reduce the size of training data is to randomly choose a subset of training set. Here, the scale of the randomly chosen training subset is the same as that of the "reduced" training set generated by the modified SOINN. Thus, we can compare the experimental results of these two methods in a fair way.

Experiments

In this section, we demonstrate the effectiveness of the proposed BAR framework in both offline and online learning circumstances. Denote the approaches using the offline SVD and NMF as offSVD and offNMF, respectively. Similarly, we denote the approach using the update SVD as onSVD. The approaches offSVD and offNMF mean we perform SVD and NMF on the matrix composed of both training and test data. And the approach onSVD means we first perform offline SVD on the matrix composed of training data to get a "reduced" joint latent factor space. Then we apply Eq. (7) to project the test data into that space.

Datasets

Three datasets are used in this paper: the "Animals with Attributes" (AWA) dataset (Lampert, Nickisch, & Harmeling, 2009), the Human Attributes (HAT) dataset (Sharma & Jurie, 2011) and the Attributes of People (AP) dataset (Bourdev, Maji, & Malik, 2011). More information about the three datasets are as follows.

AWA: it contains 30475 images of animals from 50 classes and each class is denoted with 85 human-named attributes. We randomly choose 80% of AWA as training data, and the rest images as test data.

The 85 attributes defined in AWA dataset are as follows:

1. black	18. lean	35. flys	52. fish	69. plains
2. white	19. flippers	36. hops	53. meat	70. forest
3. blue	20. hands	37. swims	54. plankton	71. fields
4. brown	21. hooves	38. tunnels	55. vegetation	72. jungle
5. gray	22. pads	39. walks	56. insects	73. mountains
6. orange	23. paws	40. fast	57. forager	74. ocean
7. red	24. longleg	41. slow	58. grazer	75. ground
8. yellow	25. longneck	42. strong	59. hunter	76. water
9. patches	26. tail	43. weak	60. scavenger	77. tree
10. spots	27. chewteeth	44. muscle	61. skimmer	78. cave
11. stripes	28. meatteeth	45. bipedal	62. stalker	79. fierce
12. furry	29. buckteeth	46. quadrapedal	63. newworld	80. timid
13. hairless	30. strainteeth	47. active	64. oldworld	81. smart
14. toughskin	31. horns	48. inactive	65. arctic	82. group
15. big	32. claws	49. nocturnal	66. coastal	83. solitary
16. small	33. tusks	50. hibernate	67. desert	84. nestspot
17. bulbous	34. smelly	51. agility	68. bush	85. domestic

HAT: it contains 9344 images of persons with 27 attributes annotations. There are 3500/3500/2344 images in training/validation/test set, respectively. We use the default split of training/validation/test set to train attribute classifiers. Note that, after using the validation set to get the best parameters, we add all the images of validation set into training set. Thus in the following experiments the training set of HAT has 7000 images.

The 27 attributes defined in HAT dataset are as follows:

1. female 2. frontalpose 3. profilepose 4. turnedback 5. upperbody 6. standing	7. runwalk 8. crouching 9. sitting 10. armsbent 11. elderly 12. middleaged	13. young 14. teen 15. kid 16. baby 17. tanktop 18. tshirt	19. casualjacket 20. mensuit 21. longskirt 22. shortskirt 23. smallshorts 24. lowcuttop	25. swimsuit 26. weddingdress 27. bermudashorts

AP: It contains 4013 training images and 4022 test images. Each person image is annotated with 9 ground truth human attributes. We also use the default split of training/test set to train attribute classifiers of AP.

The 9 attributes defined in AP dataset are as follows:

1. is_male 2. has_long_hair 3. has_glasses	4. has_hat 5. has_t-shirt 6. has_long_sleeves	7. has_shorts 8. has_jeans 9. has_long_pants

Training Attribute Classifiers

For AWA, we concatenate the six kinds of low-level features to train SVM classifiers of animal attributes (Chang & Lin, 2011). The six kinds of low-level features are HSV color histograms, SIFT, rgSIFT, PHOG, SURF, and local self-similarity histograms. For the two human datasets HAT and AP, since HOG feature (Dalal & Triggs, 2005) in combination with SVM classifier has been widely used in image recognition field, especially in pedestrian detection domain, we use Felzenszwalb's HOG (FHOG) features (Felzenszwalb, Girshick, McAllester, & Ramanan, 2010) of images to train SVM classifiers of human attributes.

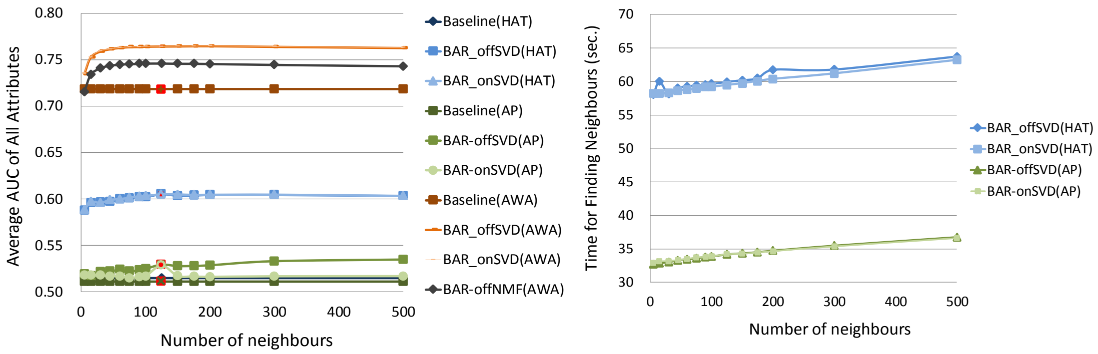
We also set up experiments on HAT dataset to demonstrate the effectiveness of BAR framework with other three different kinds of classifiers. Back-Propagation Neural Networks classifier (BPNN) (Goh, 1995), Naive Bayes classifier (NB) (Vikramkumar, B, Vikramkumar, & Trilochan, 2014), and Random Forest classifier (RF) (Bosch, Zisserman, & Muñoz, 2007). BPNN is a kind of multilayer feed-forward network to be used widely. Its training is based on error back propagation algorithm. NB calculates the posterior probability of a certain object via Bayesian formula using its prior probability. And RF is a kind of composition classifier that combines decision tree classifiers together to give the output.

For each attribute, when training classifiers we choose 100 positive samples and 100 negative samples from the default training set (24380 images for AWA, 7000 images for HAT and 4013 images for AP) as that attribute's training samples. Note that in all experiments we measure the precision of attribute prediction by area under ROC curve (AUC).

Baseline Algorithm vs Proposed BAR Framework

We use the trained attribute classifiers to test all images of the three datasets. For each dataset, we collect the predicted attribute values as an instance-attribute matrix. Each row of the matrix represents one image and each column of the matrix means the predicted values for the corresponding attribute. Note that, training images of BAR are represented by the predicted attributes values in all the following experiments. We refer to the attributes values predicted by the attribute classifiers as the performance of baseline algorithm. Because in most research these predicted attributes values are directly used to do image search, person action recognition or other computer vision tasks, without using any approaches to boost the attribute prediction accuracy first.

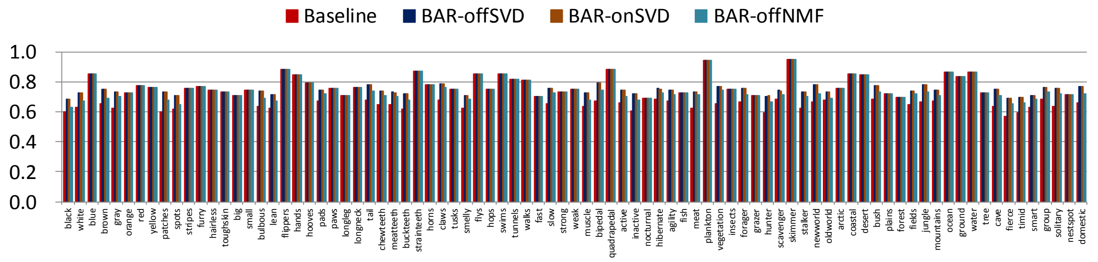
Let us denote the neighbor set of an instance n by Nei. In order to compare the performance of different sizes of neighbors (different values of $|Nei|$), we apply the proposed BAR-offSVD, BAR-offNMF and BAR-onSVD approaches on the three instance-attribute matrices of the three datasets using the default training/test split of them. Results of different sizes of neighbors are shown in Figure 4. From Figure 4 we can see that after applying the proposed approaches, the attribute prediction accuracy has been enhanced, especially for AWA and HAT datasets. Meanwhile, from Figure 4(a) we can see that performance gets better when $|Nei|$ gets larger. Correspondingly, time consumption in Figure 4(b) also increases gradually.



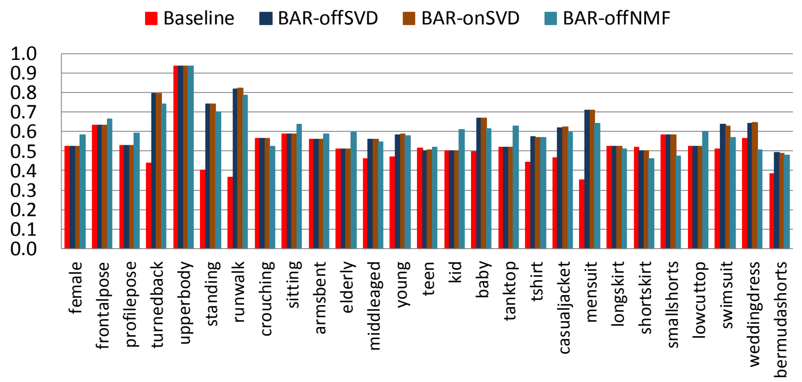
(a) (b)

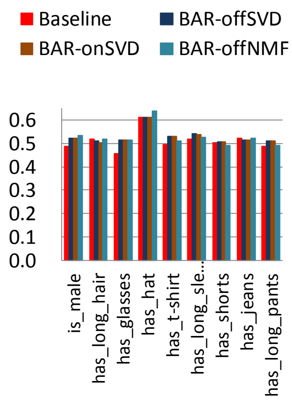
FIG. 4. Results of different sizes of neighbors $|Nei|$. (a) shows the average attribute AUC values corresponding to different values of $|Nei|$ for AWA, HAT and AP datasets, respectively. (b) gives the consuming time for finding the corresponding different numbers of neighbors ($|Nei|$) for each instance.

To better illustrate the different performances of baseline algorithm and BAR framework, the prediction accuracy of individual attribute is shown in Figure 5. We compare the performance of baseline algorithm with those of BAR-offSVD, BAR-offNMF and BAR-onSVD approaches while $|Nei| = 125$.



(a)



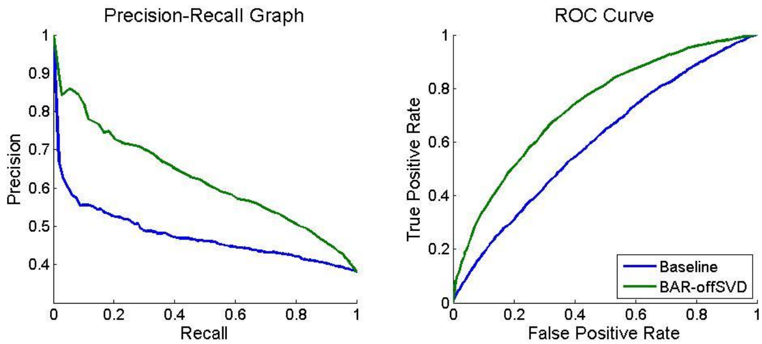


(b) (c)

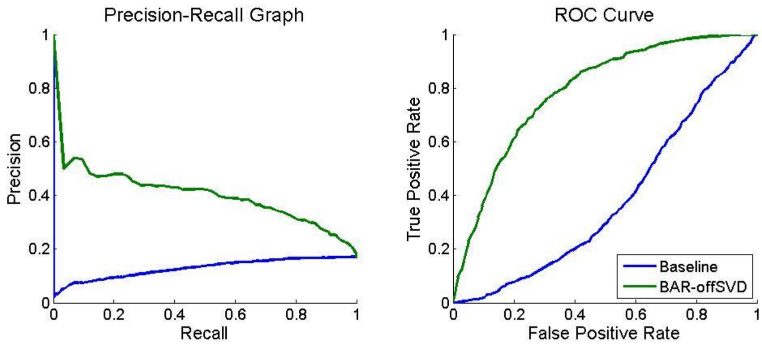
FIG. 5. Quality of individual attribute predictors combined with BAR-offSVD, BAR-onSVD and BAR-offNMF approaches. The baseline algorithm and the proposed approaches are presented on three datasets. (a) For AWA dataset. (b) For HAT dataset. And (c) for AP dataset.

From Figure 5, we can see that the proposed BAR-offNMF and BAR-onSVD approaches outperform baseline algorithm. This shows the importance of discovering and utilizing the relationships among instances, attributes and latent factors (topics). It also shows that our proposed BAR framework could be applied to boost attributes recognition in not only human dataset but also animal dataset. The reason that our approaches has a better performance for HAT dataset than AP dataset is that there are 27 specified human attributes in HAT while there are only 9 annotated attributes of people in AP. Thus, when choosing the most important k latent factors by constraining I_k keeps more than 95% information of I, it can reduce more redundant information for HAT dataset than AP dataset.

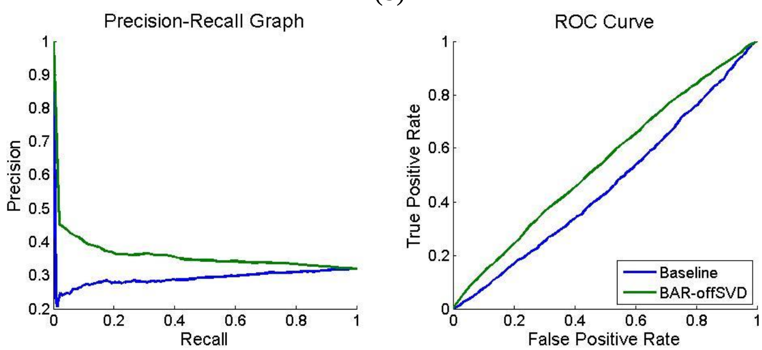
Note that, within BAR framework some attributes can be learned perfectly. For example, AUC values of attributes "plankton" and "skimmer" in AWA dataset can reach as high as 0.97 and 0.98, respectively. For another example, attribute "runwalk" in HAT dataset has an increasement of more than 0.45 (the AUC values of Baseline, BAR-offSVD and BAR-onSVD are 0.3668, 0.8198 and 0.8212, respectively). Note also that the performance of BAR-offSVD is better than that of baseline algorithm for most of the attributes, with the largest increasement of 0.1288 for attribute "patches" in AWA dataset, the largest increasement of 0.453 for attribute "runwalk" in HAT dataset and the largest increasement of 0.561 for attribute "has_glasses" in AP dataset. The precision-recall graphs and the ROC curves of attributes "patches", "runwalk" and "has_glasses" are shown in Figure 6.



(a)



(b)



(c)

FIG. 6. Precision-recall graphs and the ROC curves. (a) is for attribute "patches" in AWA. (b) is for attribute "runwalk" in HAT. And (c) is for attribute "has_glasses" in AP.

In order to demonstrate the effectiveness of BAR framework more convincingly, we also give the average attributes AUC values of 10 times cross validation experiments, shown in Table 1. Here, we use the default training/test sets of the datasets. From Table 1, we can see that the average precision of attribute prediction has been greatly increased by utilizing matrix factorization methods to discover the relationships among instances, attributes and latent factors (topics), with roughly 9% for the largest increasement. Note that, for AWA dataset, since the training images and parameters for SVM classifiers are different from our earlier work, the performance in this work is a little different from that in the earlier work (D. Li, Z. Su, H. Li & X. Luo, 2015).

TABLE 1. Average attributes AUC values of 10 times cross validation experiments (SVM classifiers).

Datasets	Baseline	BAR-offNMF	BAR-offSVD	BAR-onSVD
AWA(%)	71.82	74.58	76.43	76.42
HAT(%)	51.46	60.13	60.53	60.51
AP(%)	51.11	52.78	52.94	52.85

As aforementioned, the reason that our approaches have a better performance on HAT dataset than AP dataset is that the numbers of attributes specified for humans are quite different (27 for HAT and 9 for AP). If a large scale of human attributes is annotated, then there may be some redundant ones for us to reduce by applying BAR framework, and there would be more relationships among instances, attributes and latent factors for us to utilize. Thus, we expect our approaches have a better performance when more attributes are defined (within the same area, such as human datasets).

To be more convincing, we also compare the performances of our methods with those of some state-of-the-art methods. The result is shown in Table 2. As for AWA dataset, the average attributes AUC in the work of Lampert, Nickisch, and Harmeling (2014) is 72.4%. Although training images and parameters for SVM classifiers in our experiments are different from theirs, the performance of our baseline algorithm is almost as good as theirs (71.82% vs 72.4%). Thus, we could say that the performance of BAR outperforms the work of Lampert, Nickisch, and Harmeling (2014) (76.43% vs 72.4%). As for HAT, the mean average precision (mAP) of the 27 attributes is 53.8% in the work of Sharma and Jurie (2011). To compare with their work, we also calculate the mAP of attributes for HAT dataset. Again, though images and parameters of SVM classifiers in our experiments are different from those in the work of Sharma and Jurie (2011), the mAP of our baseline algorithm is 53.86%, which almost equals to theirs. Thus, as the mAP of BAR-offSVD is 66.51%, which is much better than the work of Sharma and Jurie (2011), we could see that the proposed BAR framework could help us to boost accuracy of attribute prediction.

TABLE 2. Comparison with state-of-the-art methods (SVM classifiers, $|Nei| = 125$).

Dataset	Lampert et al.	Baseline	BAR-offSVD
AWA(AUC, %)	72.4	71.82	76.43
Dataset	Sharma et al.	Baseline	BAR-offSVD
HAT(mAP, %)	53.8	53.86	66.51

It is well-known that SVM classifier is powerful to solve high dimensional and non-linear problems with high generalization performance (Chang & Lin, 2011). As demonstrated in Table 1, our proposed BAR framework achieves a much better attribute prediction accuracy than baseline algorithm when using SVM classifier as attribute classifiers. In the following, we want to demonstrate the effectiveness of BAR when four different kinds of classifiers are applied. Except SVM, the other three kinds of classifiers used in the experiments are Back-Propagation Neural Networks classifier (BPNN) (Goh, 1995), Naive Bayes classifier (NB) (Vijaykumar, B, Vikramkumar, & Trilochan, 2014) and Random

Forest classifier (RF) (Bosch, Zisserman, & Muñoz, 2007). BPNN is easy to trap in local minima and thus affect global optimization. NB and RF ignore the relationships among low-level features.

To figure out whether the proposed BAR framework still works with these kinds of classifiers, we set up experiments on HAT dataset when $|Nei| = 125$. In each experiment, we train SVM/BP/NB/RF classifiers for attributes in the baseline algorithm process, respectively. Then, we apply the trained SVM/BP/NB/RF classifiers to predict attributes of images. Next BAR framework is applied to boost attribute prediction accuracy. Experimental results are shown in Table 3. We can see that our BAR framework can boost the accuracy of attribute prediction with different kinds of classifiers. Thus, attribute-based image classification tasks with different kinds of classifiers can be applied in the BAR framework to boost their attribute prediction accuracy, which is especially important for their following image classification processes.

TABLE 3. Average attributes AUC values for HAT dataset when different kinds of classifiers are applied in the baseline algorithm ($|Nei| = 125$).

Classifiers	Baseline	BAR-offNMF	BAR-offSVD	BAR-onSVD
SVM(%)	51.46	60.13	60.53	60.51
BP(%)	50.12	56.72	57.60	57.59
NB(%)	49.34	56.44	57.11	57.14
RF(%)	48.63	60.51	61.68	61.77

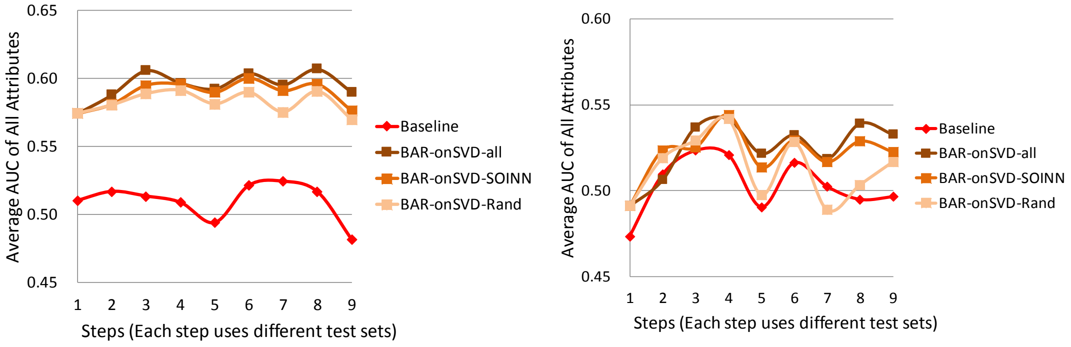
Offline Learning vs Online Learning

Actually, in Figure 4 we can see that BAR-onSVD approach achieves a good performance, like BAR-offSVD approach. In this part, we set up more experiments to compare the performance of BAR-offSVD approach with that of BAR-onSVD approach. For the two human datasets HAT and AP we randomly separate the training images into 10 subsets and index them from 1 to 10. Thus each subset contains 700 images for HAT dataset and 401 images for AP dataset.

There are 9 steps in this experiment. At each step, we add images of one subset into training data and use next subset as test data. In order to get the performances of increasing training images of BAR, we also use the default test set of HAT dataset to do experiments at each step. More specifically, at the first step we use the first subset as training data and the second subset as test data of BAR. Meanwhile, we use the 2344 test images of HAT to test the performances of BAR-offSVD and BAR-onSVD. At the second step we use the first subset plus the second subset as training data and the third subset as test data of BAR. And the same test data are the 2344 test images of HAT. The rest steps could be done in the same manner.

As aforementioned, we will reduce the scale of training data when it is too large. Here, we refer the experimental results of adding the whole images of next subset into training data at each step as BAR-offSVD-all and BAR-onSVD-all. Similarly, we refer the experimental results of using the modified SOINN algorithm and randomly choose a subset method to reduce the scale of training data as BAR-offSVD-SOINN, BAR-onSVD-SOINN, BAR-offSVD-Rand and BAR-onSVD-Rand.

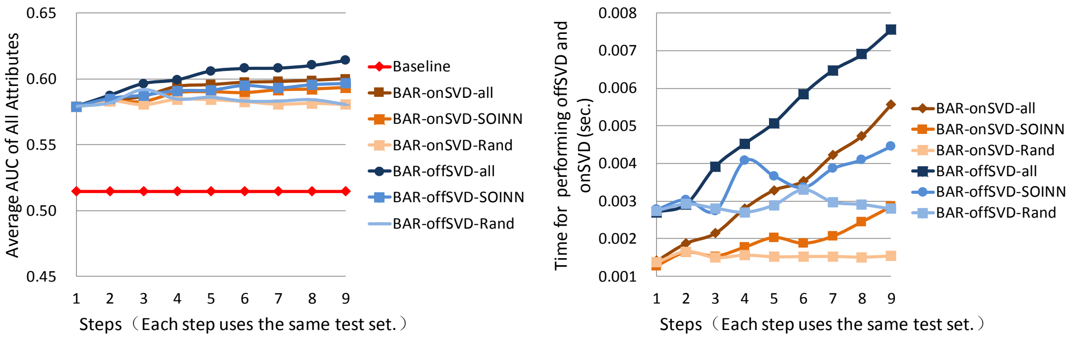
Figure 7 shows the average AUC of attributes when using next subset as test data of BAR. From Figure 7, we can see that though the best performance is achieved when the scale of training data is not reduced, the other two methods that reduce the scale of training data also have a higher accuracy than baseline algorithm.



(a) (b)

FIG. 7. Average attributes AUC of the two human datasets, when the next subset is used as test data of BAR in each of the 9 steps. We control the scale of training data in two ways. One is to apply the modified SOINN algorithm on training data and the other is to randomly choose a subset of the training data. (a) Result of HAT dataset. (b) Result of AP dataset.

Figure 8(a) gives the results when using the default test set as test data at each step for HAT. It shows that the performance of BAR-onSVD is almost as good as that of BAR-offSVD. Thus, we can apply BAR-onSVD instead of BAR-offSVD in online learning circumstance. That is, when only some training data of baseline algorithm are available, we could use BAR-onSVD approach to boost the attribute accuracy. First we perform offline SVD on the instance-attribute matrix composed of the known training data to get a “reduced” joint latent factor space. Then when new data arrive, we just need to apply Eq. 7 to project them into that space. In other words, the results indicate that not all data are needed to be available at the first beginning.



(a) (b)

FIG. 8. (a) Accuracy of HAT. Average attributes AUC of dataset HAT, when the default test set is used as test data of BAR in each of the 9 steps. (b) Consuming time of HAT. The runtime (train + test) of comparison between BAR-offSVD and BAR-onSVD is recorded. Here, the runtime of BAR-onSVD-all, BAR-onSVD-SOINN, and BAR-onSVD-Rand include not only the training time (performing traditional SVD on the instance-attribute matrix composed of the known data), but also the test time (projecting the new coming data into the existed “reduced” joint latent factor space).

Figure 8(a) also shows that the AUC curve goes up gradually when more and more new data are added. The reason is that the new data could naturally give some benefit by better populating the “reduced” joint latent space, hence *KNN* algorithm could return more reliable neighbors.

Figure 8(b) gives the consuming time of performing BAR-offSVD and BAR-onSVD on the instance-attribute matrices when using the default test set of HAT as test data at each of the 9 steps. We can see that BAR-onSVD is less time consuming than BAR-offSVD. This is convenient and time saving in real application.

From Figure 7 and Figure 8(a), we note that in all the 9 steps BAR-offSVD and BAR-onSVD get a better accuracy of attribute prediction than baseline algorithm. This effectively proves the validity of the proposed BAR framework. Because the latent factor models from BAR-offSVD and BAR-onSVD explain the prediction of attribute by characterizing both instances and attributes. Thus, we can discover and utilize the relationships among instances, attributes and latent factors (topics) to boost attribute recognition.

Discussion

There are two main advantages of boosting attribute recognition. The basic one is to give more accurate attribute descriptions for person recognition task (Sharma & Jurie, 2011; Bourdev, Maji, & Malik, 2011). For example, we can tell what attributes the object has and what it was not. In another words, we can use more accurate attributes to describe images and get a better understanding of them, such as "is male", "has long hair". Attributes are especially important when derive new categories of objects become available after the training period. For example, in Figure 9, though we do not know what category this object is subjected to, we can at least tell that it has the attributes "furry", "black", "horns" and so on by using the trained attribute classifiers. Meanwhile we can also tell that it does not has the attributes "strips", "longneck", "bipedal" and so on. This is on account of the sharing between objects characteristic of attributes. In other words, we can use the available images of some objects to train attribute classifiers, and use the trained classifiers to do attributes prediction for other objects.

FIG. 9. A derived new object. There are not have any samples of this object during training period.

The other advantage is that attribute-based computer vision tasks could benefit from enhanced attribute prediction. For example, for human data, a better action recognition performance could be expected in Liu's work (Liu, Kuipers, & Savarese, 2011), since their model recognizes person actions from videos by attributes. For animal data, Lampert et al. tackled zero-shot learning problem by using an attribute layer (Lampert, Nickisch, & Harmeling, 2009). Zero-shot learning means test data are disjoint from training data. Attributes can help us to solve the widely existed "semantic gap" problem, so we can utilize attributes to tackle the zero-shot learning problem. With 85 attributes specified in AWA dataset and an attributes layer is utilized in their work, we expect that by utilizing BAR framework to get an enhanced animal attribute prediction. It would have a contribution for solving the zero-shot learning problem.

Furthermore, experimental results show that our proposed BAR framework is not only applicable to humans data (HAT dataset and AP dataset), but also applicable to animals data (AWA dataset). Thus we can expect the BAR framework could be extended to many other areas, such as scenes data.

The concept of attributes was started in 2009 (Farhadi, Endres, Hoiem, & Forsyth, 2009; Lampert, Nickisch, & Harmeling, 2009). Since then, more and more researchers have studied the problem of attribute prediction. So far, attributes have been widely applied in a lots of computer vision tasks, such as zero-shot learning, relative zero-shot learning, learning with a few examples, attributes-based feedback to classifiers about mistakes, image search systems and so on. Hence, boosting attribute recognition is of great significance for those computer vision tasks.

Conclusion

In this paper, we propose a unity framework called BAR to boost both human and animal attribute recognition. BAR is based on traditional SVD, NMF and a new updating SVD matrix factorization models. The proposed framework discovers relationships among instances, attributes and latent factors (topics). These relationships could be used to boost performance of attribute prediction, and give us a more accurate way to describe images. Besides, our framework could be used in both offline and online circumstances. Offline learning is suitable for the circumstance when all data can be available at the beginning. And online learning is applied to the scenario when only parts of the needed data are available at the beginning. Considering training efficiency when the scale of training data gets larger and larger, we proposed an algorithm to reduce the scale of training data, which is based on SOINN algorithm.

In the future, on the one hand we plan to apply our enhanced attribute prediction results in tasks like person (or animal) action recognition or pedestrian tracking. We will try to solve these problems in a semantic and generalized way. On the other hand, we will try to apply the proposed BAR framework to boost the prediction accuracy of relative attributes, instead of binary attributes. Furthermore, since the average attribute prediction accuracy of both human datasets and animal dataset are not good enough, we can try other methods to further boost attribute recognition. For example, we can utilize Latent Dirichlet Allocation (LDA) algorithm to discover the topic models between instances and attributes, and further utilize this topic information to enhance the attribute prediction accuracy.

Acknowledgments

This work is supported by the National Science Fund of China (No. 61502541, 61370186) NSFC-Guangdong Joint Fund (No. U1135003), and the National Key Technology R&D Program (2011BAH27B01).

References

Li, Y., Xu, S., Luo, X., & Lin, S. (2014). A new algorithm for product image search based on salient edge characterization. Journal of the Association for Information Science and Technology (JASIST), 65(12), 2534−2551.

Sharma, G., & Jurie, F. (2011). Learning discriminative spatial representation for image classification. British Machine Vision Conference (BMVC) (pp. 1−11). Dundee: BMVA.

Lampert, C.H., Nickisch, H., & Harmeling, S. (2014). Attribute-based classification for zero-shot visual object categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 36(3), 453−465.

Bourdev, L., Maji, S., & Malik, J. (2011). Describing people: A poselet-based approach to attribute classification. International Conference on Computer Vision (ICCV) (pp. 1543−1550). Barcelona: IEEE.

Farhadi, A., Endres, I., Hoiem, D., & Forsyth, D. (2009). Describing objects by their attributes. International Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1778−1785). Miami: IEEE.

Russakovsky, O., & Feifei, L. (2010). Attribute learning in large-scale datasets. European Conference on Computer Vision (ECCV) (pp. 1−14). Berlin: Springer.

Lampert, C.H, Nickisch, H., & Harmeling, S. (2009). Learning to detect unseen object classes by between-class attribute transfer. International Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 951−958). Miami: IEEE.

Parikh, D., & Grauman, K. (2011). Relative attributes. International Conference on Computer Vision (ICCV) (pp. 503−510). Barcelona: IEEE.

Kovashka, A., Parikh, D., & Grauman, K. (2012). Whittlesearch: Image search with relative attribute feedback. International Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 2973−2980). Providence: IEEE.

Wang, X., & Zhang, T. (2011). Clothes search in consumer photos via color matching and attribute learning. ACM International Conference on Multimedia (ACM MM) (pp. 1353−1356). Scottsdale: ACM.

Suzuki, M., Sato, H., Oyama, S., & Kurihara, M. (2014). Image classification by transfer learning based on the predictive ability of each attribute. International Multiconference of Engineers and Computer Scientists (IMECS), 2209(1), 75−78.

Chen, H., Gallagher, A., & Girod, B. (2012). Describing clothing by semantic attributes. European Conference on Computer Vision (ECCV) (pp. 609−623). Florence: Springer.

Guo, Z., & Wang, Z.J. (2013). An unsupervised hierarchical feature learning framework for one-shot image recognition. IEEE Transactions on Multimedia (TMM), 15(3), 621−632.

Turakhia, N., & Parikh, D. (2013). Attribute dominance: What pops out?. International Conference on Computer Vision (ICCV) (pp. 1225−1232). Sydney: IEEE.

Han, Y., Wu, F., Lu, X., Tian, Q., Zhuang, Y., & Luo, J. (2012). Correlated attribute transfer with multi-task graph-guided fusion. ACM International Conference on Multimedia (ACM MM) (pp. 529−538). Nara: ACM.

Li, S., Shan, S., & Chen, X. (2012). Relative forest for attribute prediction. Asian Conference on Computer Vision (ACCV) (pp. 316−327). Daejeon: Springer.

Han, Y., Yang, Y., Ma, Z., Shen, H., Sebe, N., & Zhou, X. (2014). Image attribute adaptation. IEEE Transactions on Multimedia (TMM), 16(4), 1115−1126.

Song, F., Tan, X., & Chen, S. (2012). Exploiting relationship between attributes for improved face verification. British Machine Vision Conference (BMVC) (pp. 1−11). Surrey: BMVA.

Yu, X., & Aloimonos, Y. (2010). Attribute-based transfer learning for object categorization with zero/one training example. European Conference on Computer Vision (ECCV) (pp. 127−140). Crete: Springer.

Taussky, O., & Todd, J. (2006). Cholesky, toeplitz and the triangular factorization of symmetric matrices. Numerical Algorithms, 41(2), 197−202.

Chu, E., & George, A. (1987). QR factorization of a dense matrix on a shared-memory multiprocessor. Parallel Computing, 11(1), 55−71.

Kalman, D. (1996). A singularly valuable decomposition: The svd of a matrix. The College Mathematics Journal, 27(1), 2−23.

Lee, D.D., & Seung, H.S. (2001). Algorithms for non-negative matrix factorization. Advances in Neural Information Processing Systems (NIPS), 13(6), 556−562.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. IEEE Computer, 42(8), 30−37.

Takács, G., Pilászy, I., Németh, B., & Tikk, D. (2008). Matrix factorization and neighbor based algorithms for the netflix prize problem. ACM Conference on Recommender Systems (RecSys) (pp. 267−274). Lausanne: ACM.

Riedel, S., Yao, L., McCallum, A., & Marlin, B.M. (2013). Relation extraction with matrix factorization and universal schemas. The Annual Conference of the North American Chapter of the Association of Computational Linguistics (NAACL) (pp. 74−84). Atlanta: NAACL.

Zhang, Q., & Li, B. (2010). Discriminative k-svd for dictionary learning in face recognition. International Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 2691−2698). San Francisco: IEEE.

Sali, S. (2008). Movie rating prediction using singular value decomposition. University of California, Machine Learning Project Report.

Rajwade, A., Rangarajan, A., & Banerjee, A. (2013). Image denoising using the higher order singular value decomposition. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 35(4), 849−862.

Lee, D.D., & Seung, H.S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755), 788−791.

Berry, M.W., Dumais, S.T., & O'Brien, G.W. (1995). Using linear algebra for intelligent information retrieval. Siam Review, 37(4), 573−595.

Skrondal, A., & Rabe-Hesketh, S. (2007). Latent variable modelling: A survey. Scandinavian Journal of Statistics, 34(4), 712−745.

Furao, S., & Hasegawa, O. (2006). An incremental network for on-line unsupervised classification and topology learning. Neural Networks, 19(1), 90−106.

Shen, F., & Hasegawa, O. (2010). Self-organizing incremental neural network and its application. International Conference on Artificial Neural Networks (ICANN) (pp. 535−540). Thessaloniki: Springer.

Chang, C.C., & Lin, C.J. (2011). LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3), 389−396.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. International Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 886−893). San Diego: IEEE.

Felzenszwalb, P.F., Girshick, R.B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 32(9), 1627−1645.

Goh, A.T.C. (1995). Back-propagation neural networks for modeling complex systems. Artificial Intelligence in Engineering, 9(3), 143–151.

Vijaykumar, B, Vikramkumar, & Trilochan, (2014). Bayes and naive bayes classifier. The Journal of Clinical Orthopaedics and Related Research.

Bosch, A., Zisserman, A., & Muñoz, X. (2007). Image classification using random forests and ferns. International Conference on Computer Vision (ICCV) (pp. 1−8). Rio de Janeiro: IEEE.

Li, D., Su, Z., Li, H., & Luo, X. (2015). Boosting accuracy of attribute prediction via svd and nmf of instance-attribute matrix. Pacific-Rim Conference on Multimedia (PCM) (pp. 466−476). Gwangju: Springer.

Liu, J., Kuipers, B., & Savarese, S. (2011). Recognizing human actions by attributes. Intertional Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 3337−3344). Colorado Springs: IEEE.