The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Zero-Shot Learning with Attribute Selection*

Yuchen Guo,† Guiguang Ding,† Jungong Han,‡ Sheng Tang§
†School of Software, Tsinghua University, Beijing 100084, China
‡School of Computing and Communications, Lancaster University, Lancaster, LA1 4YW, UK
§Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China
{yuchen.w.guo,jungonghan77}@gmail.com, dinggg@tsinghua.edu.cn, ts@ict.ac.cn

Abstract

Zero-shot learning (ZSL) is regarded as an effective way to construct classification models for target classes which have no labeled samples available. The basic framework is to transfer knowledge from (different) auxiliary source classes having sufficient labeled samples with some attributes shared by target and source classes as bridge. Attributes play an important role in ZSL but they have not gained sufficient attention in recent years. Previous works mostly assume attributes are perfect and treat each attribute equally. However, as shown in this paper, different attributes have different properties, such as their class distribution, variance, and entropy, which may have considerable impact on ZSL accuracy if treated equally. Based on this observation, in this paper we propose to use a subset of attributes, instead of the whole set, for building ZSL models. The attribute selection is conducted by considering the information amount and predictability under a novel joint optimization framework. To our knowledge, this is the first work that notices the influence of attributes themselves and proposes to use a refined attribute set for ZSL. Since our approach focuses on selecting good attributes for ZSL, it can be combined to any attribute based ZSL approaches so as to augment their performance. Experiments on four ZSL benchmarks demonstrate that our approach can improve zero-shot classification accuracy and yield state-of-the-art results.

Introduction

Image classification, whose goal is to identify the category of instances in an image, is an active research topic in machine learning and computer vision communities. Recently, benefiting from the fast development of deep learning techniques (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2014; He et al. 2016; Huang et al. 2016), the image classification accuracy on many benchmarks, including the large-scale ImageNet (Russakovsky et al. 2015), has been improved tremendously and even surpassed human-level performance. It should be noticed that the progress in image classification relies heavily on a large-scale training set which provides sufficient labeled samples for each category. A large number of labeled samples are easy to collect for common categories. However, as Lampert, Nickisch, and Harmeling (2014) have pointed out, there are at least tens of thousands of basic object categories in the world, and much more fine-grained ones. In reality, the object categories follow a long-tail distribution, where most of them occur infrequently such that it is expensive to collect a large number of labeled samples for them. Moreover, new concepts, such as a new type of electronic device like iPhone8, may occur in the Web everyday. It is also difficult to find sufficient exemplars for these new concepts. Therefore, how to train classification models for these uncommon or new categories which have very limited labeled samples, and no samples in the extreme case, is a practical problem and has attracted considerable research interest (Farhadi et al. 2009; Lampert, Nickisch, and Harmeling 2014; Al-Halah, Tapaswi, and Stiefelhagen 2016; Guo et al. 2017a).

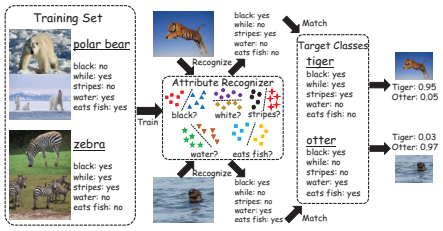
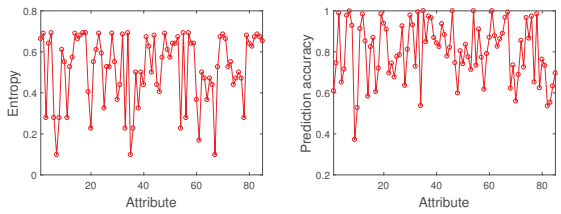


Figure 1: The basic framework for attribute-based ZSL.

To address this problem, zero-shot learning (ZSL) has been introduced as a promising solution (Farhadi et al. 2009). It is observed that although the labeled sample for some target classes is not given, there are always a large number of different auxiliary classes having sufficient labeled samples. So the key is to find a bridge to transfer supervised knowledge from auxiliary classes to target classes. One widely used way is class attributes which define the properties of the corresponding class and are shared between source and target classes, which is briefly illustrated in Figure 1. For example, we can define attributes like "stripes", "four legs", and "water" for animals. Then we can train attribute recognizers (classification or regression models) using images and attribute information from auxiliary classes which are different but related to target classes. Then given a test image from a target

*This research was supported by the National Natural Science Foundation of China (Grant No. 61571269) and the Royal Society Newton Mobility Grant (IE150997). Corresponding author: Guiguang Ding.



(a) Information amount (b) Predictability

Figure 2: Properties of attributes on AwA2.

class which is unseen before, these attribute recognizers can produce the attributes of the image. Finally, by computing the similarity between the test image's attributes with each target class' attributes, a prediction score (e.g., probability) for each target class is obtained and the final output is given based on the score.

Observations and Contributions

While previous works mostly pay attention to building effective recognizers (Socher et al. 2013; Xian et al. 2016) or matching strategies (Zhang and Saligrama 2015; Fu et al. 2015b) or both (Norouzi et al. 2013; Fu et al. 2015a), the key building block in ZSL, the attributes themselves, does not seem to receive comparable attention. Previous works implicitly treat attributes equally ignoring they basic statistical properties. For example, in Direct Attribute Prediction (DAP) (Lampert, Nickisch, and Harmeling 2014), one of seminal ZSL works, a binary classifier is trained for each attribute and the attribute distance is simply measured by the probability distance between attribute vectors. In this way, an uncertain attribute prediction and a certain attribute prediction have the same contribution to the distance measure, which is obviously unreasonable. In fact, the attributes have different properties such that we should treat them in different manners. In particular, we notice two important properties which have significant impacts. We use AwA2 (Xian et al. 2017) dataset for illustration which has 50 classes and 85 binary attributes. The first is information amount of an attribute which indicates how the attribute can help to distinguish classes. For a binary attribute, we use p to denote the ratio of classes having this attribute and $1-p$ to denote the ratio of other classes. Then we can use entropy $-p\log p-(1-p)\log(1-p)$ as the information amount. A tiny entropy indicates that almost all or none classes have this attribute so that it contributes little to classification. We plot the entropy of 85 attributes of AwA2 in Figure 2(a). We can observe some attributes have small entropy, like "tusks" and "plankton". In fact, only a very small part of classes have these attributes and including these attributes seems to overfit the dataset so that a model generalizes badly on test set. The second is predictability indicating the likelihood an attribute can be correctly predicted from an image. Given an attribute, if it is very difficult to be recognized from an image, including it is helpless for ZSL and even harmful because a wrong prediction on this attribute may lead the model to the wrong direction. We use the "train" set in AwA2 to train 85 binary SVMs as attribute recognizers and test them on the "val" set. The attribute prediction accuracy for each attribute is plotted in Figure 2(b). The accuracy of some attributes is near or below 50% which is the level of random guess. For example, the accuracy of attributes "inactive", "smelly", and "solitary" is about 50% because they are difficult to recognize by using only visual information extracted from the image. Therefore, we should not expect to obtain useful information from them.

Based on these observation, we argue that not all attributes are necessary and helpful for ZSL and different attributes have different importance. Consequently, it is not a good choice to treat them equally as in most previous ZSL approaches. Inspired by these results, we propose to perform attribute selection in the attribute set to find informative and predictable attributes and then construct ZSL models based on the selected subset. We consider two criteria, information amount, and predictability, in a joint optimization framework. In this way, a set of good attributes are selected which lead to better ZSL model since useless and "noisy" attributes are removed. Because our approach focuses on the attribute level, not ZSL model level, we can combine it with any existing ZSL approaches, like DAP, by using the selected attributes as input. With better attributes, the performance of these ZSL models can be further improved. In summary, we make the following contributions in this paper:

1. We show that the attributes in ZSL benchmarks have different properties, including information amount and predictability. Previous ZSL works ignore the diversity and treat each attribute equally such that they are influenced by "noisy" attributes. Consequently, their accuracy is limited.

2. We propose a novel attribute selection framework for ZSL. By simultaneously considering information amount and predictability of each attribute in a joint optimization framework, we select the most valuable attributes for subsequent ZSL classification models to improve their accuracy.

3. We combine our attribute selection approach with several ZSL classification models. Experiments on four benchmark datasets demonstrate the state-of-the-art performance and that ZSL accuracy is indeed improved by the selected attributes with an observable margin, validating the efficacy and necessity of the proposed attribute selection approach.

Preliminaries and Related Works

Problem Definition and Notations

Zero-shot learning problem can be described as follows. Our goal is to build classification models for a set of target classes $\mathcal{C}^t=\{c_1^t,...,c_{k_t}^t\}$ which have no labeled samples available. At test stage, given a test image $x^t\in\mathbb{R}^d$ as image feature, we predict its class label $c(x^t)\in\mathcal{C}^t$. Since there is no label information for $\mathcal{C}^t$, we need another set of source classes $\mathcal{C}^s=\{c_1^s,...,c_{k_s}^s\}$ which have n_s labeled training samples $\mathcal{D}^s=\{(x_1^s,y_1^s),...,(x_{n_s}^s,y_{n_s}^s)\}$ where x_i is image feature and $y_i\in\mathcal{C}^s$ is class label. In ZSL setting, source classes are different from target classes, i.e., $\mathcal{C}^s\cap\mathcal{C}^t=\emptyset$. In order to transfer supervision knowledge between classes, for each class $c\in\mathcal{C}^s\cup\mathcal{C}^t$, an attribute vector $\mathbf{a}_c\in\mathbb{R}^q$ for it. We summarize some frequently used notations in Table 1.

Table 1: Notations and descriptions.

Notation	Description	Notation	Description
x	feature	n	#samples
y	label	d	#dimension
$\mathbf{a}$	class attribute	q	#attributes
f	model	k	#classes
w	weight	α, β, γ	parameters

Related Works

As surveyed in (Xian et al. 2017), ZSL usually consists of two steps. The first step is feature embedding or attribute recognition, which is a kind of multi-modality matching problem (Zheng, Tang, and Shao 2016; Zheng and Shao 2016), and the second step is attribute matching, which can be summarized briefly as the following formulation:

$$c(x) = \text{argmax}_{c \in \mathcal{C}^t} \mathcal{S}(\varphi(x), \mathbf{a}_c) \quad (1)$$

where $\varphi(x)$ is an attribute recognizer which can be classifier (Lampert, Nickisch, and Harmeling 2014) or regressor (Socher et al. 2013), and $\mathcal{S}(\cdot, \cdot)$ is a similarity measure function. To learn the function φ, source classes and their labeled images are used:

$$\min_{\varphi} \sum\nolimits_{i=1}^{n_s} \mathcal{L}(\varphi(x_i^s), \mathbf{a}_{y_i^s}) \quad (2)$$

where $\mathcal{L}(\cdot, \cdot)$ is a loss measure between recognized attributes and true attributes. By solving this loss function, we obtain φ. As the attributes are shared between source and target classes, the attribute recognizer φ trained using source classes can also work for target classes (e.g., the "stripes" recognizer trained with "tiger" can help to recognize "stripes" in "zebra"), which is a fundamental assumption in ZSL.

Different ZSL approaches mainly share the general formulation above, but may have different choices for the function φ, the similarity measure $\mathcal{S}$ for test, and the loss measure $\mathcal{L}$ for training, in their specific formulations. For example, in DAP (Lampert, Nickisch, and Harmeling 2014), binary classifier, a weighted inner product similarity, and classification loss are used for φ, $\mathcal{S}$, and $\mathcal{L}$. In Cross Modal Transfer (Socher et al. 2013), they use a combination of linear projection and $\tanh$ function for φ and squared Euclidean distance for $\mathcal{S}$ and $\mathcal{L}$. In Attribute Label Embedding (Akata et al. 2016), they adopt linear projection for φ, inner product similarity for $\mathcal{S}$, and weighted approximate ranking loss (Usunier, Buffoni, and Gallinari 2009) for $\mathcal{L}$. In Simple ZSL (Romera-Paredes and Torr 2015), they utilize linear projection, inner product similarity and squared Euclidean distance respectively. Bucher, Herbin, and Jurie (2016) propose to use linear projection, Mahalanobis distance, and hinge loss. In Latent Embedding Model (LatEm) (Xian et al. 2016), multiple linear projections with latent variables, inner product similarity, and ranking loss are employed. In fact, many other ZSL approaches (Frome et al. 2013; Akata et al. 2015; Changpinyo et al. 2016; Guo et al. 2017b) follow the general formulation. We cannot review them all due to the space limitation. Please refer to (Xian et al. 2017) and (Guo et al. 2017a) for more detailed discussion.

Zero-shot Learning with Attribute Selection

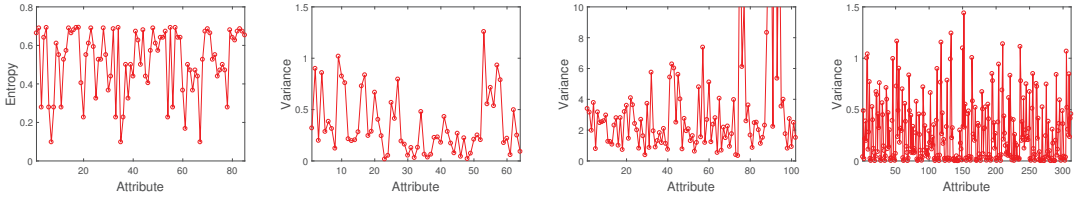
Properties of Attributes

In Eq. (1) and (2), the attributes in the dataset are utilized without any discrimination. For example, in CMT, the distance between a class' attribute vector $\mathbf{a}_c$ and a sample's predicted attribute vector $\mathbf{a}_i = \varphi(x_i)$ is $d(\mathbf{a}_i, \mathbf{a}_c) = \|\mathbf{a}_i - \mathbf{a}_c\|^2 = \sum_{j=1}^{q}(a_{cj} - a_{ij})^2$. In the inner product similarity case (Romera-Paredes and Torr 2015; Akata et al. 2016; Xian et al. 2016), we also have $\mathcal{S}(\mathbf{a}_i, \mathbf{a}_c) = \sum_{j=1}^{q} a_{ij} a_{cj}$. This phenomenon indicates that all attributes have the same weight for the similarity measure regardless of the properties of attributes themselves. However, it is straightforward to see this is unreasonable. For example, one attribute can be hardly predictable such that the attribute recognizer always gives a wrong prediction. In this situation, its wrong attribute prediction may lead to small similarity to a correct class and large similarity to a wrong class. If this attribute is selected, it may act as noise which affects the whole model.

Noticing this, we argue that not all attributes are helpful for ZSL and removing some of them can improve ZSL accuracy. In this paper, we consider two important properties of attributes. The first property is information amount which indicates how the attribute can help to distinguish classes. It is expected that an attribute can provide as much information as possible. This property is widely considered when a human performs classification. For example, when a human plays "twenty questions" game[1] to guess an animal category, asking whether an animal lives in water (i.e., attribute "water") seems more informative than whether it has tusks (i.e., attribute "tusks") and the former leads to faster arrival to the answer. In addition, given an attribute with low information amount, a correct attribute prediction does not help to identify a class, but a wrong prediction may hurt the performance. However, the attributes in benchmark datasets have different information amount. To demonstrate this, we use four benchmark datasets AwA2 (Xian et al. 2017), aPascal-aYahoo (Farhadi et al. 2009), SUN (Patterson and Hays 2012), and CUB (Wah et al. 2011)[2]. For AwA2 with binary attributes, we use entropy of an attribute to measure its information amount, where a larger entropy indicates this attribute can well separate classes. For the other datasets with continuous attributes, we use the variance of attributes (i.e., variance of $a_{cj}, \forall c$) as measurement, where a larger variance indicates different classes are more separatable on this attribute. We plot the information amount of different attributes for four benchmarks in Figure 3. Obviously, we observe the information amount of attributes varies a lot where some attributes have very low information amount, such as "tusks" and "plankton" in AwA2, where they appear only in a few classes. Analogous to principle components analysis, removing components (attributes) with low variance or entropy leads to better performance in some cases since noisy information is removed. Considering the information amount difference between attributes, it seems unreasonable to treat them equally as in previous ZSL approaches.

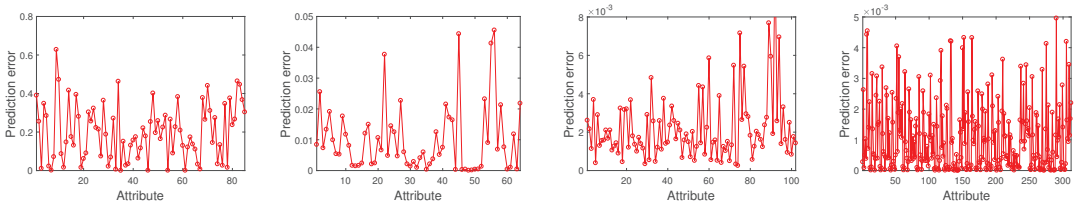
[1] https://en.wikipedia.org/wiki/Twenty_Questions

[2] We use the datasets, including features, labels, attributes, and



(a) AwA2 (b) aPY (c) SUN (d) CUB

Figure 3: The information amount of attributes, measured by entropy for binary attributes and variance for continuous attributes.



(a) AwA2 (b) aPY (c) SUN (d) CUB

Figure 4: The predictability of attributes, measured by classification or squared error for binary or continuous attributes.

The second property is the predictability of attributes. If an attribute is hard to recognize, e.g., it has large classification or regression error, this attribute may have negative impact on the ZSL system because it is likely to bring in wrong information. Therefore, it is important to check whether an attribute is predictable from images. Here we also use four benchmark datasets mentioned above. We train attribute recognizers (binary SVM classifier for AwA2 and linear projection function for the others) using "train" sets and test them on "val" sets. The prediction error is measured by classification error $\sum_{i,j} \mathbb{1}(\varphi_j(x_i), a_{ij})/nq$ for binary attributes on AwA2 where φ_j is a binary classifier for the j-th attribute and $\mathbb{1}(x, y)$ returns 1 if $x \neq y$ or 0 otherwise, and squared error $\sum_{i,j}(\varphi_j(x_i) - a_{ij})^2/nq$ where φ_j is a linear regressor for continuous attributes on the other datasets. The prediction error is plotted in Figure 4. As can be observed, different attributes have diverse predictability and some attributes seem hard to predict. For example, there are several attributes in AwA2 whose classification error is around or even above 50% which is the level of random guess. We notice that these attributes include "inactive", "domestic", and "smelly" which are almost impossible to be predicted based only on visual information, and "spots" and "patches" whose characteristics are not significant in images. Although some of them have high information amount, their low predictability may lead to mismatching to class attributes which degrades final accuracy which should be considered in ZSL.

Attribute Selection

Based on the above analysis, we demonstrate that different attributes have different information amount and predictability and thus we should not treat them equally as previous works. So attribute selection is necessary for ZSL. Based on Figure 3 and 4, one straightforward and naive strategy is to select attributes whose information amount is larger than a threshold and prediction error smaller than another threshold. We denote this strategy as naive attribute selection (NAS). Experiments show that ZSL models can already be augmented by the selected attributes even if NAS is used. However, NAS is a model independent method which cannot be optimized with ZSL model jointly. Considering the ultimate task is to construct ZSL model, performing ZSL model optimization and attribute selection simultaneously seems to be a better choice, which is elaborated as follows.

We simultaneously consider ZSL model construction, information amount maximization, and predictability maximization in a joint optimization framework as follows:

$$\begin{aligned}\min_{w_j,\varphi_j,\mu_j,f} \mathcal{O} &= \sum_{i=1}^{n_s} \mathcal{L}_{\text{ZSL}}(f(x_i), \{w_1, ..., w_q\}, \{\mathbf{a}_1, ..., \mathbf{a}_{k_s}\}, y_i) \\ &+ \alpha \sum_{i=1}^{n_s}\sum_{j=1}^{q} w_j \mathcal{L}_p(\varphi_j(x_i), a_{ij}) - \beta \sum_{i=1}^{n_s}\sum_{j=1}^{q} w_j(\varphi_j(x_i) - \mu_j)^2 \\ &+ \gamma \sum_{j=1}^{q} w_j^2, \quad \text{s.t. } w_j \geq 0, \sum_{j=1}^{q} w_j = 1\end{aligned} \tag{3}$$

where w_j is the weight for the j-th attribute which will be further used for attribute selection, φ_j is an attribute recognizer used to measure predictability, μ_j is an auxiliary variable used to measure information amount (variance), and f is the target ZSL model. The objective function consists of three parts. The first part is model based loss for ZSL which can use previous works (Romera-Paredes and Torr 2015; Akata et al. 2016; Xian et al. 2016). The second part is to measure the predictability where $\mathcal{L}_p$ is attribute prediction

data splits, given by: http://www.mpi-inf.mpg.de/zsl-benchmark

loss which can be defined based on attributes and models. The third part takes into account the variance of attributes which is a measure of information amount where classes are more discriminative if this attribute has larger variance. Compared to NAS, Eq. (3) is model-aware and data-aware, which better fits the task and thus leads to better results.

We can optimize Eq. (3) in an alternative manner where we optimize one variable while fixing the others. To derive the optimization algorithm, we need to specify the choice of functions in Eq. (3). In fact, it is easy to combine attribute selection with state-of-the-art models. For example, we can simply use a linear projection $\varphi_j(x_i) = x_i p_j^t$ where $p_j \in \mathbb{R}^d$ is the projection parameter for φ_j, and squared Euclidean error $\mathcal{L}_p(a,b) = (a-b)^2$. When combined with ESZSL (Romera-Paredes and Torr 2015), $\mathcal{L}_{\text{ZSL}}$ is defined as:

$$\mathcal{L}_{\text{ZSL}} = \sum_{c=1}^{k_s} (x_i \cdot \mathbf{U} \cdot (w \circ \mathbf{a}_c)^T - I(c, y_i))^2 \quad (4)$$

where $\circ$ is element-wise multiplication, and $I(a,b)$ is an indicator function which is 1 if $a = b$ or -1 otherwise, and $\mathbf{U} \in \mathbb{R}^{d \times q}$ is model parameters for f in ESZSL. By fixing the other variables, the partial derivatives of $\mathcal{L}_{\text{ZSL}}$ to $\mathbf{U}$ is:

$$\frac{\partial \mathcal{L}_{\text{ZSL}}}{\partial \mathbf{U}} = 2 \sum_{c=1}^{k_s} x_i^t \cdot (x_i \cdot \mathbf{U} \cdot (w \circ \mathbf{a}_c)^T - I(c, y_i)) \cdot (w \circ \mathbf{a}_c) \quad (5)$$

Then we can use Stochastic Gradient Descent (SGD) to optimize $\mathbf{U}$. To optimize w_j, we need to rewrite $\mathcal{O}$ as follows:

$$\mathcal{O}_w = w \cdot \mathbf{B} \cdot w^T + w \cdot \mathbf{h}^T + m, \text{ s.t. } w_j \geq 0, w \cdot \mathbf{1}_q^T = 1 \quad (6)$$

where $\mathbf{B} = \gamma \mathbf{I}_q + \sum_i \mathbf{G}_i^T \cdot \mathbf{G}_i$, $\mathbf{h} = -2 \sum_i \mathbf{G}_i^T \cdot \mathbf{z}_i^T + \alpha \mathbf{l}_p - \beta \mathbf{u}$, m is a constant not related to w, $\mathbf{G}_i = \{(x_i \cdot \mathbf{U}) \circ \mathbf{a}_c; c = 1, ..., k_s\} \in \mathbb{R}^{k_s \times q}$, $\mathbf{z}_i = \{-1, 1\}^{k_s}$ where $z_{ic} = 1$ if $c = y_i$ or -1 otherwise, $\mathbf{l}_p = \{\sum_i (\varphi_j(x_i) - a_{ij})^2, j = 1, ..., q\} \in \mathbb{R}^q$, and $\mathbf{u} = \{\sum_i (x_i) - \mu_j)^2, j = 1, ..., q\} \in \mathbb{R}^q$. Minimizing Eq. (6) is a standard quadratic programming problem which can be solved efficiently by well-established tools. In this paper, we use MATLAB function `quadprog`[3]. By fixing the other variables, optimizing φ_j is quite simple:

$$\begin{aligned} &\min_{p_j} \alpha \sum_{i=1}^{n_s} (x_i p_j^t - a_{ij})^2 - \beta \sum_{i=1}^{n_s} (x_i p_j^t - \mu_j)^2 \\ &\Rightarrow p_j = (\alpha \mathbf{A}_j \cdot \mathbf{X} - \beta \mu_j \mathbf{1}_{n_s})((\alpha - \beta)\mathbf{X}^T \cdot \mathbf{X} + \epsilon \mathbf{I}_d)^{-1} \end{aligned} \quad (7)$$

where $\mathbf{A}_j = [a_{1j}, ..., a_{n_s j}]$, $\mathbf{X} = \{\mathbf{x}_i; i = 1, ..., n_s\}$, and ϵ is a small positive number to avoid numeric problem. Then we just need to update $\mu_j = \frac{1}{n_s} \sum_{i=1}^{n_s} \varphi_j(x_i)$. By iteratively updating these variables by the above rules until convergence, we will finally obtain weight w_j for each attribute.

For some ranking based loss and linear projection and similarity, like ALE (Akata et al. 2016), LatEm (Xian et al. 2016), SJE (Akata et al. 2015), and DEVISE (Frome et al. 2013), the ZSL loss $\mathcal{L}_{\text{ZSL}}$ is generally defined as follows:

$$\mathcal{L}_{\text{ZSL}} = \sum_{c=1}^{k_s} r_{ic} [\Delta(y_i, c) + x_i \cdot \mathbf{U} \cdot (w \circ \mathbf{a}_c)^T - x_i \cdot \mathbf{U} \cdot (w \circ \mathbf{a}_{y_i})^T]_+ \quad (8)$$

[3]http://cn.mathworks.com/help/optim/ug/quadprog.html

Table 2: The statistics of datasets.

	AwA2	aPY	SUN	CUB
#source class	40	20	645	150
#source sample	30,512	7,415	12,900	8,821
#target class	10	12	72	50
#target sample	7,913	7,924	1,440	2,967
#attributes	85	64	102	312

where $\Delta(a,b) = 1$ if $a = b$ or 0 otherwise and $r_{ic} \in [0,1]$ is a sample-label based weight defined in these approaches. For example, in SJE, $r_{ic} = 1$ if $c = \text{argmax}_c \Delta(y_i, c) + x_i \cdot \mathbf{U} \cdot (w \circ \mathbf{a}_c)$ or 0 otherwise. In DEVISE and LatEm, $r_{ic} = 1 (\forall c)$. For ALE, r_{ic} is a ranking based weight (Usunier, Buffoni, and Gallinari 2009). The partial derivative to $\mathbf{U}$ is:

$$\frac{\partial \mathcal{L}_{\text{ZSL}}}{\partial \mathbf{U}} = \sum_{c=1}^{k_s} r_{ic} g_{ic} x_i^T (w \circ (\mathbf{a}_c - \mathbf{a}_{y_i})) \quad (9)$$

where $g_{ic} = 1$ if $\Delta(y_i, c) + x_i \cdot \mathbf{U} \cdot (w \circ \mathbf{a}_c)^T - x_i \cdot \mathbf{U} \cdot (w \circ \mathbf{a}_{y_i})^T \geq 0$ or 0 otherwise. Analogously, we can redefine the the variables in Eq. (6) in this problem, where $\mathbf{B} = \gamma \mathbf{I}_q$, $\mathbf{h} = \sum_i \sum_c x_i \cdot \mathbf{U} \circ (\mathbf{a}_c - \mathbf{a}_{y_i}) + \alpha \mathbf{l}_p - \beta \mathbf{u}$. Here we remove the $[\cdot]_+$ operation to simplify the problem. Then we can also use Eq. (7) to update φ_j and iterate these steps towards convergence.

Moreover, for approaches whose goal is to predict the attributes directly, like DAP and CMT, $\mathcal{L}_{\text{ZSL}}$ is equivalent to $\mathcal{L}_p$. So it is straightforward to combine their loss to Eq. (3).

After solving Eq. (3) we obtain the weight for each attribute. Then we can use them for attribute selection. One simple strategy is hard selection where we only preserve the top q_s attributes. The other is soft selection where we assign weight w_j to each attribute for ZSL model training. We will compare them in the next section. Then based on these selected (weighted) attributes, we can train the ZSL models. Because Eq. (3) takes ZSL model into consideration, the selected attributes can improve their accuracy significantly.

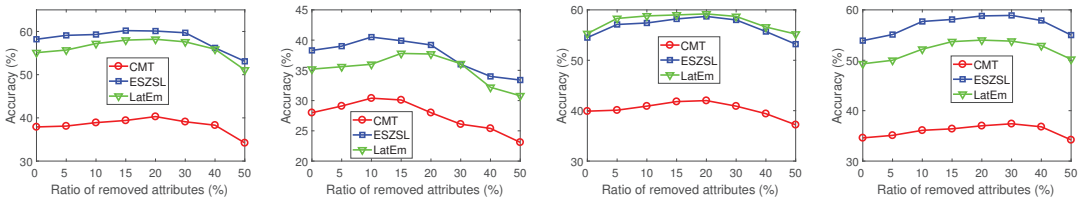
Experiment

Settings

Following Xian et al. (2017), we use AwA2 (Xian et al. 2017), aPascal-aYahoo (Farhadi et al. 2009), SUN (Patterson and Hays 2012), and CUB (Wah et al. 2011) benchmark datasets, whose statistics are summarized in Tabel 2. We use train set and val set which contain source classes and samples for training, and use test set which has target classes and samples for evaluating. As suggested in (Xian et al. 2017), we use *per-class averaged top-1 accuracy* for evaluation:

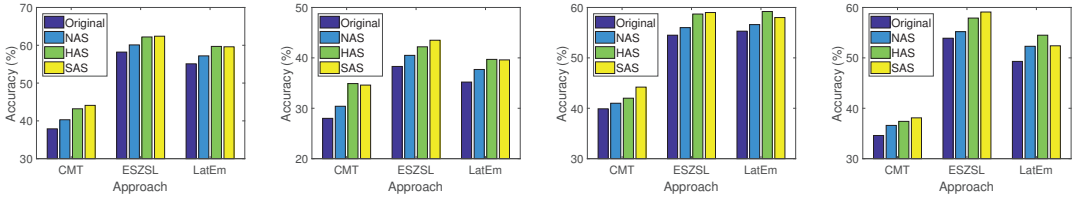
$$\text{Accuracy} = \frac{1}{k_t} \sum_{c \in \mathcal{C}^t} \frac{\#\text{correct predictions in } c}{\#\text{samples in } c} \quad (10)$$

As an important property, our attribute selection can be combined with many ZSL approaches because they focus on how to use attributes while our approach focuses on



(a) AwA2, NAS (b) aPY, NAS (c) SUN, HAS (d) CUB, HAS

Figure 5: The accuracy with respect to the ratio of removed attributes (dataset, attribute selection strategy).



(a) AwA2 (b) aPY (c) SUN (d) CUB

Figure 6: The accuracy with respect to different attribute selection strategies.

how to choose attributes. In this paper, we combine our approach with DAP (Lampert, Nickisch, and Harmeling 2014), CMT (Socher et al. 2013), ESZSL (Romera-Paredes and Torr 2015), SJE (Akata et al. 2015), ALE (Akata et al. 2016), and LatEm (Xian et al. 2016), because they are the most representative ZSL works and easy to implement. We first solve Eq. (3) based on the specific $\mathcal{L}_{\text{ZSL}}$ for different approaches. Then based on the selected attributes, we retrain ZSL models which are used for evaluation. When retraining models, as suggested by (Xian et al. 2017), we use "train" set for training and "val" set for validation to choose hyper-parameters and use both of them with optimal values for the final model.

Ablation Study

We propose three attribute selection strategies. The first strategy direct finds top ranked q_s attributes based on information amount and predictability, which is termed as naive attribute selection (NAS). The second strategy solves Eq. (3) to obtain attribute weight w_j and selects top q_s attributes with the largest weights, which is called hard attribute selection (HAS). The third terms also obtains w_j. But it assigns weights to attributes to train ZSL models without removing attributes, which is termed as soft attribute selection (SAS).

In the first experiment, we investigate the influence of the number of selected attributes on ZSL performance. We consider NAS and HAS because they directly remove attributes and we use CMT, ESZSL, and LatEm in this experiment. In Figure 5, we plot the ZSL accuracy with respect to the ratio of removed samples ($r = 1 - q_s/q$) for different ZSL approaches, datasets, and selection strategies. Generally, we have the following two main observations from the results.

Firstly, at the beginning, removing a small part of attributes (e.g., 5% to 20%) usually leads to higher accuracy, which demonstrates that not all attributes are necessary for ZSL and some of them can be even harmful. In fact, the first removed attributes have low information amount and low predictability. They can be regarded as noise to some extent for ZSL. Moreover, some attributes with low predictability will cause large $\mathcal{L}_{\text{ZSL}}$ because they are difficult to recognize and the optimization procedure may focus on them to minimize their loss such that the information of other attributes is not well captured. Therefore, removing them makes ZSL model concentrate more on important information in other attributes such as the valuable characteristics that can help to distinguish classes, are well captured by ZSL models.

Secondly, when more attributes are removed (e.g., 30% to 50%), the accuracy drops observably. The results are quite reasonable because the procedure will progressively remove more informative and predictable attributes as it goes on. Consequently, too many useful attributes are removed such that the model is not given sufficient knowledge for ZSL. This phenomenon also indicates that most of the attributes designed for the existing benchmarks are useful for ZSL.

In the second experiment, we compare the accuracy of different attribute selection strategies, i.e., NAS, HAS, and SAS. The comparison is summarized in Figure 6. For NAS and HAS, we remove 20% attributes because this ratio always leads to best performance as suggested in Figure 5. We can draw the following three conclusions from the results.

Firstly, compared to the original attributes, the selected attributes, no matter which strategy is employed, all achieve better performance with observable margin which is consistent with the results in Figure 5. This is another evidence which demonstrates the necessity of attribute selection.

Secondly, we can observe that HAS and SAS yield significantly better performance than NAS. As introduced above, NAS is a model-independent strategy where the selection does not consider the property of the subsequent ZSL model.

Table 3: Zero-shot accuracy comparison on benchmarks. Numbers in brackets are relative performance gains.

	AwA2	aPY	SUN	CUB	Average
Norouzi et al. (2013)	44.5	26.9	38.8	34.3	36.13
Zhang and Saligrama (2015)	61.0	34.0	51.5	43.9	47.60
Changpinyo et al. (2016)	46.6	23.9	56.3	55.6	45.60
Kodirov et al. (2015)	54.1	8.3	40.3	33.3	34.00
Frome et al. (2013)	59.7	39.8	56.5	52.0	52.00
CMT (Socher et al. 2013)	37.9	28.0	39.9	34.6	35.10
CMT + AS	42.77(+**4.87**)	34.22(+**6.22**)	43.40(+**3.50**)	37.81(+**3.21**)	39.55(+**4.45**)
DAP (Lampert, Nickisch, and Harmeling 2014)	46.1	33.8	39.9	40.0	39.95
DAP + AS	48.29(+**2.19**)	34.87(+**1.07**)	42.27(+**2.37**)	41.55(+**1.55**)	41.75(+**1.80**)
ESZSL (Romera-Paredes and Torr 2015)	58.6	38.3	54.5	53.9	51.33
ESZSL + AS	61.71(+**3.11**)	43.02(+**4.72**)	58.90(+**4.40**)	58.21(+**4.31**)	55.46(+**4.13**)
LatEm (Xian et al. 2016)	55.8	35.2	55.3	49.3	48.9
LatEm + AS	59.07(+**3.27**)	38.82(+**3.82**)	58.09(+**2.79**)	52.82(+**3.52**)	52.20(+**3.30**)
SJE (Akata et al. 2015)	61.9	32.9	53.7	53.9	50.6
SJE + AS	62.59(+**0.69**)	35.12(+**2.22**)	53.77(+**0.07**)	55.10(+**1.20**)	51.64(+**1.04**)
ALE (Akata et al. 2016)	62.5	39.7	58.1	54.9	53.80
ALE + AS	64.39(+**1.89**)	43.44(+**3.74**)	60.52(+**2.42**)	54.81(−**0.09**)	55.79(+**1.99**)

Considering the ultimate goal is to construct classification models and attribute selection is a step to the goal, it is necessary to combine the information from ZSL model in attribute selection. In some cases, an informative and predictable attribute cannot fit a ZSL model well. In Eq. (3), we simultaneously optimize the attribute selection and ZSL model learning in a joint optimization framework so that the selected attributes are informative, predictable, and compatible with ZSL models, which results in better performance.

Thirdly, HAS and SAS have comparable performance and one performs better in some cases and ZSL approaches while the other performs better in some other cases. The reason is two folds. On one hand, as mentioned previously, treating all attribute equally is not reasonable because different attributes have different properties. So, SAS well addresses this problem and assigns different weights to different attributes. In this way, the importance of attributes is reflected in their weights and the ZSL model may better capture the intrinsic knowledge in the attributes and achieve better performance. On the other hand, incorporating weights into ZSL model training makes the problem more complicated which is likely to affect the performance. But hard selection can avoid this problem because it directly removes low-weight attributes. As suggested in Figure 5, most attributes are useful and assigning the same weight to them seems acceptable. For simple approaches like ESZSL, the first issue has larger impact so that SAS performs better. For more complicated approaches like LatEm, the second issue seems more dominant and thus HAS yields superior results.

Benchmark Comparison

We combine the proposed attribute selection (AS) with 6 ZSL approaches introduced above. We use HAS which removes 20% attributes and SAS, and the choice is made based on the performance on "val" set. The comparison is summarized in Table 3, where the numbers in brackets are the relative improvements given by AS. We can observe that the accuracy of ZSL approaches is significantly improved by AS. In particular, the average improvement on four datasets and six approaches is $\mathbf{2.79}\%$ which is a large improvement for ZSL considering its difficulty. Moreover, there are 18 out of 24 approach-dataset combinations achieving more than 2% improvements, which indicates that the proposed attribute selection is consistently necessary for different approaches and datasets. In addition, combined with AS, the best results on four datasets are increased by $\mathbf{1.89}\%$, $\mathbf{3.64}\%$, $\mathbf{2.42}\%$, and $\mathbf{2.61}\%$ respectively ($\mathbf{2.64}\%$ on average), which also demonstrates the effectiveness of attribute selection.

Moreover, in 24 approach-dataset combinations, we observe that CMT and ESZSL always use soft AS while ALE and SJE typically choose hard AS. As discussed above, soft AS is more difficult to optimize when combined with ZSL approaches but it can lead to better results. So simple approaches like ESZSL and CMT work better with SAS while complicated approaches like ALE and SJE worse. This is an interesting phenomenon. In our future work, we will try to find a better way to combine soft AS and complicated approaches like ALE. Moreover, by using naive AS, we can combine AS with more complicated approaches (Kodirov et al. 2015; Changpinyo et al. 2016). But how to combine more effective strategies, HAS and NAS, with them is still a challenge, which will be investigated in our future study.

Conclusion

In this paper we consider the key building block for ZSL, attributes. Previous ZSL approaches treat all attribute equally without considering their properties. We notice different attributes have different information amount and predictability in real-world datasets. Based on this observation, we propose a novel attribute selection approach for ZSL which simultaneously considers the information amount and predictability of an attribute in a joint optimization framework. Based on the selected attributes, we can train any ZSL approaches. Experiments on several datasets demonstrate the proposed

attribute selection can significantly and consistently improve ZSL accuracy and yield state-of-the-art- results.

References

Akata, Z.; Reed, S. E.; Walter, D.; Lee, H.; and Schiele, B. 2015. Evaluation of output embeddings for fine-grained image classification. In *CVPR*.

Akata, Z.; Perronnin, F.; Harchaoui, Z.; and Schmid, C. 2016. Label-embedding for image classification. *IEEE Trans. Pattern Anal. Mach. Intell.*

Al-Halah, Z.; Tapaswi, M.; and Stiefelhagen, R. 2016. Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. In *CVPR*.

Bucher, M.; Herbin, S.; and Jurie, F. 2016. Improving semantic embedding consistency by metric learning for zero-shot classiffication. In *ECCV*.

Changpinyo, S.; Chao, W.; Gong, B.; and Sha, F. 2016. Synthesized classifiers for zero-shot learning. In *CVPR*.

Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. A. 2009. Describing objects by their attributes. In *CVPR*.

Frome, A.; Corrado, G. S.; Shlens, J.; Bengio, S.; Dean, J.; Ranzato, M.; and Mikolov, T. 2013. Devise: A deep visual-semantic embedding model. In *NIPS*.

Fu, Y.; Hospedales, T. M.; Xiang, T.; and Gong, S. 2015a. Transductive multi-view zero-shot learning. *IEEE Trans. Pattern Anal. Mach. Intell.*

Fu, Z.; Xiang, T.; Kodirov, E.; and Gong, S. 2015b. Zero-shot object recognition by semantic manifold distance. In *CVPR*.

Guo, Y.; Ding, G.; Han, J.; and Gao, Y. 2017a. Zero-shot learning with transferred samples. *IEEE Trans. Image Processing*.

Guo, Y.; Ding, G.; Han, J.; and Gao, Y. 2017b. Zero-shot recognition via direct classifier learning with transferred samples and pseudo labels. In *AAAI*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Huang, G.; Liu, Z.; Weinberger, K. Q.; and van der Maaten, L. 2016. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*.

Kodirov, E.; Xiang, T.; Fu, Z.; and Gong, S. 2015. Unsupervised domain adaptation for zero-shot learning. In *ICCV*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2014. Attribute-based classification for zero-shot visual object categorization. *IEEE TPAMI*.

Norouzi, M.; Mikolov, T.; Bengio, S.; Singer, Y.; Shlens, J.; Frome, A.; Corrado, G.; and Dean, J. 2013. Zero-shot learning by convex combination of semantic embeddings. *CoRR* abs/1312.5650.

Patterson, G., and Hays, J. 2012. SUN attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*.

Romera-Paredes, B., and Torr, P. H. S. 2015. An embarrassingly simple approach to zero-shot learning. In *ICML*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. S.; Berg, A. C.; and Li, F. 2015. Imagenet large scale visual recognition challenge. *IJCV*.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.

Socher, R.; Ganjoo, M.; Manning, C. D.; and Ng, A. Y. 2013. Zero-shot learning through cross-modal transfer. In *NIPS*.

Usunier, N.; Buffoni, D.; and Gallinari, P. 2009. Ranking with ordered weighted pairwise classification. In *ICML*.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical report.

Xian, Y.; Akata, Z.; Sharma, G.; Nguyen, Q. N.; Hein, M.; and Schiele, B. 2016. Latent embeddings for zero-shot classification. In *CVPR*.

Xian, Y.; Lampert, C. H.; Schiele, B.; and Akata, Z. 2017. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *CoRR* abs/1707.00600.

Zhang, Z., and Saligrama, V. 2015. Zero-shot learning via semantic similarity embedding. In *ICCV*.

Zheng, F., and Shao, L. 2016. Learning cross-view binary identities for fast person re-identification. In *IJCAI*.

Zheng, F.; Tang, Y.; and Shao, L. 2016. Hetero-manifold regularization for cross-modal hashing. *IEEE TPAMI*.