Zero-shot Recognition via Direct Classifier Learning with Transferred Samples and Pseudo Labels

AAAI Anonymous Submission 182

Abstract

As an interesting and emerging topic, zero-shot recognition (ZSR) makes it possible to train a recognition model by specifying the category's attributes when there are no labeled exemplars available. The fundamental idea for ZSR is to transfer knowledge from the abundant labeled data in different but related source classes via the class attributes. Conventional ZSR approaches adopt a **two-step** strategy in test stage, where the samples are projected into the attribute space in the first step, and then the recognition is carried out based on considering the relationship between samples and classes in the attribute space. Due to this intermediate transformation, information loss is unavoidable, thus degrading the performance of the overall system. Rather than following this two-step strategy, in this paper, we propose a novel **one-step** approach that is able to perform ZSR in the original feature space by using directly trained classifiers. To tackle the problem that no labeled samples of target classes are available, we propose to assign **pseudo labels** to samples based on the reliability and diversity, which in turn will be used to train the classifiers. Moreover, we adopt a robust SVM that accounts for the unreliability of pseudo labels. Extensive experiments on four datasets demonstrate consistent performance gains of our approach over the state-of-the-art two-step ZSR approaches.

Introduction

In the recent years, we have witnessed the emerging of zero-shot recognition (ZSR) in computer vision and related communities. Basically, the objective of ZSR is to build classification models for target classes with no labeled samples (Lampert, Nickisch, and Harmeling 2014). To construct supervised models without supervision information for target classes, existing approaches make use of knowledge from related but different source classes which are well labeled, and transfer the knowledge to target classes via the classes' attributes (Farhadi et al. 2009; Socher et al. 2013). Generally, a two-step strategy in the test stage is adopted. Firstly, based on the supervision information in source classes, a projection function that projects the original features into the attribute space is learned, such as linear projection (Akata et al. 2013) and n-way classifier (Norouzi et al. 2013). Secondly, after projecting a test sample from the target classes into the attribute space, the classification is performed by considering the relationship between the test sample and all target classes in the attribute space. Typically, the relationship can be measured by Euclidean distance (Socher et al. 2013), inner-product similarity (Guo et al. 2016), and manifold distance (Fu et al. 2015). The basic framework of the two-step strategy is illustrated in Figure 1 on the left side.

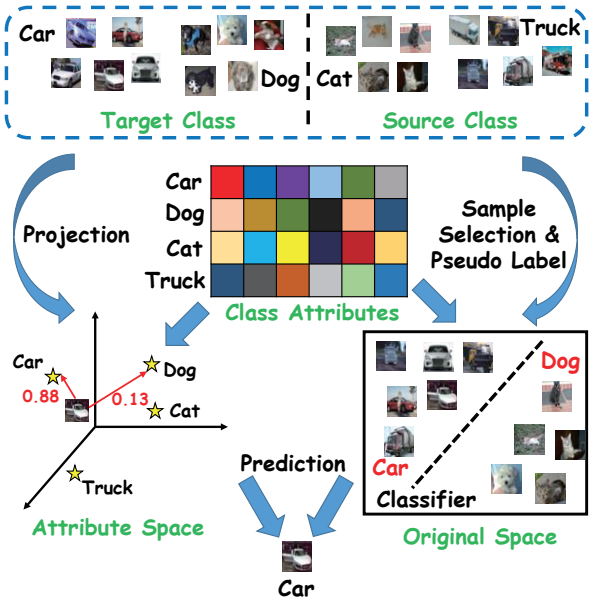


Figure 1: Frameworks of the previous two-step strategy and the proposed one-step approach. Ours does not require the intermediate space in the **test** stage while the previous do.

In this paper, we propose a novel approach that adopts **one-step** strategy in the test stage where classifiers trained in the original feature space are used to enable a sample-to-class prediction, which is briefly illustrated, as opposed to the two-step approach, in Figure 1 on the right side. As our approach does not need an intermediary space during the classification (test) procedure, it can benefit from much less information loss, thus achieving better performance. However, it has to confront the problem that no labeled data is available for the target classes. To address this issue, we propose to use **pseudo labels** in the training phase. For instance, we need to train a car-dog classifier but we only have

Figure 2: Some selected sample images from truck and cat. We use them with pseudo labels to train a car-dog classifier.

labeled samples from truck and cat. Intuitively, we can select some samples from truck class alike to car (measured in the attribute space) and treat them as the labeled samples for car. Similarly, we can select some samples from cat as the labeled samples for dog. Having used the pseudo labels, a classifier can be trained as usual. It is believed that such a scheme is very logical because human being often uses one category to deduce another one as long as they share the same characteristics. In Figure 2, we illustrate 20 selected images (500 in total) from truck and cat respectively. We assign pseudo labels to them, and on top of it, we train a linear car-dog SVM classifier. We found out that the classifier is able to achieve 92.80% recognition accuracy in the test set.

Implementing the above idea in a ZSR system is not that simple, because we have to solve two problems. Firstly, we need to select proper samples for pseudo labeling. Here, we take two aspects into consideration for the sample selection procedure, which are reliability and diversity. The former reflects how a sample is similar to the target classes while the latter one shows how the selected samples cover the different characteristics of target classes. In our approach, we formulate them as a joint quadratic problem. Secondly, we have to deal with the label noise which arises from the fact that the pseudo labels are not the true labels. To alleviate the influence of the label noise, we propose to use a robust SVM classifier. The main contributions of this paper are as below:

- We propose a novel one-step approach for ZSR. Classifiers for target classes are directly trained in the original feature space by using the pseudo labels. Such classifiers allow to directly predict the class labels from test samples without utilizing the attribute space as the intermediary so that less information is leaked in the whole procedure.
- To select good samples for pseudo labeling, we propose a quadratic formulation taking into account the reliability and the diversity of the selected samples simultaneously.
- To cope with the label noise existed in the pseudo labels, we make use of a robust SVM as the classifier which can be derived from the standard SVM and efficiently trained.

Related Work

Now we introduce the key principles for ZSR and ZSR variations. For ease of explanation, we firstly define the notations.

Notations

We have a set of source classes $\mathcal{C}^s = \{c_1^s, ..., c_{k_s}^s\}$ and n_s labeled source samples $\mathcal{D}^s = \{(\mathbf{x}_1^s, \mathbf{y}_1^s), ..., (\mathbf{x}_{n_s}^s, \mathbf{y}_{n_s}^s)\}$, where $\mathbf{x}_i^s \in \mathbb{R}^d$ is the feature vector and $\mathbf{y}_i^s \in \{0,1\}^{k_s}$ is the corresponding label vector which has $y_{ij} = 1$ if the sample i belongs to class c_j^s or 0 otherwise. We are given some target samples $\mathcal{D}^t = \{\mathbf{x}_1^t, ..., \mathbf{x}_{n_t}^t\}$ from k_t target classes $\mathcal{C}^t = \{c_1^t, ..., c_{k_t}^t\}$ satisfying $\mathcal{C}^s \cap \mathcal{C}^t = \emptyset$. The goal of ZSR is to build classification models which can predict the label $c(\mathbf{x}_i^t)$ given $\mathbf{x}_i^t$ with no labeled training data for target classes. For each class $c_i \in \mathcal{C}^s \cup \mathcal{C}^t$, we assign a class attribute representation $\mathbf{a}_i \in \mathbb{R}^q$ to it. We summarize some notations in this paper and the corresponding descriptions in Table 1.

Table 1: Notations and descriptions.

Notation	Description	Notation	Description
$\mathbf{x}_s, \mathbf{x}_t$	features	n_s, n_t	#samples
$\mathbf{y}_s, \mathbf{y}_t$	label vector	d	#dimension
$\mathbf{a}_s, \mathbf{a}_t$	class attribute	q	#attributes
α	weights	k_s, k_t	#classes
r	ranking score	σ, β, μ, C	parameters

Basic Idea of ZSR

Generally, the classification in ZSR can be summarized as:

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

where $\text{sim}(\cdot,\cdot)$ denotes a similarity / distance measure, and $\mathcal{P}$ represents a projection function which is learned as below,

$$\mathcal{P} = \text{argmin}_{\mathcal{P}} \sum_{i=1}^{n_s} \ell(\mathcal{P}(\mathbf{x}_i^s), \mathbf{a}_{c(\mathbf{x}_i^s)}), \quad (2)$$

where $\ell(\cdot,\cdot)$ is a loss function. As the source samples are fully labeled, $c(\mathbf{x}_i^s)$ is known for the above problem. And since the attributes are shared among source and target classes, the projection learned based on the source classes also works in the target classes (Lampert, Nickisch, and Harmeling 2014).

ZSR Variations

The main difference of various ZSR approaches lies in using different projections or/and similarity measures. Lampert et al. (2009) propose Direct Attribute Prediction which adopts binary classifiers for $\mathcal{P}$ and Euclidean distance. Socher et al. (2013) propose a Cross-modal Transfer which uses a nonlinear projection for $\mathcal{P}$ and isometric Gaussian probability. Akata et al. (2013) propose a Label Embedding which adopts a linear projection and Euclidean distance. Fu et al. (2015) propose to use a deep model DeViSE (Frome et al. 2013) for projection and measure the similarity using the semantic manifold distance obtained from absorbing Markov chain process. Jayaraman and Grauman (2014) propose a random forest approach and the prediction error statistics of attributes are considered in the similarity measure. Kodirov et al. (2015) propose to use sparse coding for projection learning. Norouzi et al. (2013) and Fu et al. (2014) consider ZSR under the transductive setting and propose to perform label propagation on a graph to produce the final similarity.

Some recent works (Guo et al. 2016; Romera-Paredes and Torr 2015) have attempted to construct classifiers in the original space, which is similar to this paper. In their approaches,

the parameters of the linear classifier for class c is constructed as $\mathbf{w}_c = \mathbf{a}_c\mathbf{V}'$ where $\mathbf{a}_c$ is the class attribute and $\mathbf{V}$ is the parameter to be learned. Their classification is achieved by

$$c(\mathbf{x}_i^t) = \text{argmax}_c \mathbf{x}_i^t \mathbf{w}_c' = \text{argmax}_c \mathbf{x}_i^t \mathbf{V} \mathbf{a}_c'. \quad (3)$$

In fact, it is equivalent to Eq. (1) where $\mathbf{V}$ is the projection and the similarity is measured by the inner-product similarity. Hence, they also follow the two-step strategy. In addition, they focus on learning $\mathbf{V}$ instead of the hyperplanes of classifiers, which can be regarded as indirect classifier learning. Thus, these approaches are intrinsically different from ours.

The Proposed Approach

Sample Selection

The key idea of the proposed approach is to select samples which are most similar to the target classes for a direct classifier learning. As there is no labeled data for target classes and only class attributes are given, we measure the similarity in the attribute space. To do so, it is required to project the samples into the attribute space. Here, we need to highlight the difference between existing ZSR approaches and ours because we also use the projection to some extent. Existing approaches adopt the projection as an important step for the classification (test), whereas in our approach the attributes are only utilized to select samples for classifier training and we do not need them during the whole classification process.

To find the projection, we can follow the general framework introduced in Eq. (2). Since it is not the focus of this paper, we just utilize the linear projection learned as follows,

$$\min_{\mathbf{P}} \sum_{i=1}^{n_s} \|\mathbf{x}_i^s \mathbf{P} - \mathbf{a}_{c(\mathbf{x}_i^s)}\|_F^2 + \sum_{j=1}^{n_t} \|\mathbf{x}_j^t \mathbf{P} - \mathbf{a}_{\tilde{c}(\mathbf{x}_j^t)}\|_F^2, \quad (4)$$

where $\|\cdot\|_F$ denotes the Frobenius norm of matrix. Here, $\tilde{c}(\mathbf{x}_j^t)$ is the *estimated* label for an unlabeled sample from target classes. We will discuss how to obtain it later. In the above formulation, we also incorporate the information from the target classes such that the learned projection can avoid the domain shift problem (Fu et al. 2014; Kodirov et al. 2015). Denote $\mathbf{X} = [\mathbf{x}_1^s; ...; \mathbf{x}_{n_s}^s; \mathbf{x}_1^t; ...; \mathbf{x}_{n_t}^t]$ and $\mathbf{A} = [\mathbf{a}_{c(\mathbf{x}_1^s)}; ...; \mathbf{a}_{c(\mathbf{x}_{n_s}^s)}; \mathbf{a}_{\tilde{c}(\mathbf{x}_1^t)}; ...; \mathbf{a}_{\tilde{c}(\mathbf{x}_{n_t}^t)}]$, the closed-form solution to the above problem can be written as below,

$$\mathbf{P} = (\mathbf{X}'\mathbf{X} + \epsilon \mathbf{I}_d)^{-1} \mathbf{X}'\mathbf{A}, \quad (5)$$

where $\mathbf{I}_d$ is an identity matrix, ϵ is a small positive value to avoid numeric problem, $(\cdot)^{-1}$ denotes the inverse of matrix and $(\cdot)'$ expresses the transpose. Now, following (Socher et al. 2013), given the projection $\mathbf{P}$, the similarity between a sample $\mathbf{x}_i$ and a target class c_j^t and attribute $\mathbf{a}_{c_j^t}$ is defined as

$$s_i = \mathcal{N}(\mathbf{x}_i \mathbf{P} | \mathbf{a}_{c_j^t}, \mathbf{I}), \quad (6)$$

where $\mathcal{N}$ is the Gaussian distribution. For each sample in the dataset, we can compute the similarity s_i between it and target class c_j^t. Now we can select samples from the dataset to assign pseudo label c_j^t to them based on the similarity. Intuitively, to capture the characteristics of target class, it is expected that the selected samples to be as similar to the target class as possible. This criterion, termed as reliability, can be implemented by the following optimization problem,

$$\min_{r_i} \sum\nolimits_{i=1}^{n} -r_i s_i + \mathcal{R}(\mathbf{r}),\ s.t.,\ \mathbf{r}\mathbf{1}_n' = 1, r_i \geq 0, \quad (7)$$

where r_i is the ranking score for sample i and $\mathcal{R}(\mathbf{r})$ is the regularization term. Then we rank all the samples by their ranking scores and the top-ranked samples are selected to assign pseudo label c_j^t. Here, we use the notation n indiscriminately. In fact, we have $n = n_s$ if the selection is performed only with source samples, $n = n_t$ if only with unlabeled target samples, or $n = n_s + n_t$ if with both source and target samples. In this paper, we adopt a disjoint selection strategy, i.e., we treat source samples and unlabeled target samples independently. First, we select m_s samples by solving Eq. (7) on $\mathcal{D}^s$. Then, another m_t samples are selected from $\mathcal{D}^t$.

In fact, without a proper regularization, Eq. (7) will simply assign large scores to samples with large s_i. However, this result may be very redundant because similar samples may have similar scores. For example, if $\mathbf{x}_i$ has a large s_i based on Eq. (6), another sample $\mathbf{x}_j$ which is highly similar to $\mathbf{x}_i$, also has a large s_j because $\mathbf{x}_i\mathbf{P}$ and $\mathbf{x}_j\mathbf{P}$ are close in the attribute space. For this case, both of them are selected. If so, the selected samples are not diverse enough for training an effective classifier (Bishop and others 2006). To tackle this issue, we propose to use the **diversity** as a regularization term for Eq. (7). Specifically, we first define a heat kernel matrix (Belkin and Niyogi 2001) to measure the similarity between samples as $K_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2)$ where σ is set to the mean Euclidean distance between feature vectors in the candidate set. Note that our direct classifier learning is performed in the original feature space, we expect the selected samples to be diverse in the original space, and thus the kernel matrix is defined in the original space. Then we can utilize the regularization term $\mathcal{R}(\mathbf{r}) = \frac{1}{2}\sum_{i,j} K_{ij} r_i r_j$. Obviously, if $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar (K_{ij} is large), assigning large scores r_i and r_j simultaneously will lead to a large value for $\mathcal{R}(\mathbf{r})$. Hence, selecting diverse samples is equivalent to minimizing this term. With the diversity regularization, the overall objective function can be described as follows

$$\min_{\mathbf{r}} \frac{\beta}{2}\mathbf{r}\mathbf{K}\mathbf{r}' - \mathbf{r}\mathbf{s}',\ s.t.\ \mathbf{r}\mathbf{1}_n' = 1, r_i \geq 0, \quad (8)$$

where β is a trade-off parameter to balance reliability and diversity. It is a standard constrained quadratic programming (QP) problem. We can use well-established tools to solve it, such as the `quadprog` function in Matlab. However, the time complexity is $\mathcal{O}(n^3)$ for a typical QP solver, which is a bit too expensive. Instead, based on the augmented Lagrange multiplier (ALM) framework (Bertsekas 1999), we adopt a more efficient algorithm for this problem in this paper.

We first rewrite Eq. (8) into the standard ALM framework,

$$\min_{\mathbf{r}} \frac{\beta}{2}\mathbf{r}\mathbf{K}\mathbf{r}' - \mathbf{r}\mathbf{s}',\ s.t. \mathbf{r}\mathbf{1}_n' - 1 = 0, \mathbf{r} - \mathbf{u} = 0, u_i \geq 0, \quad (9)$$

where $\mathbf{1}_n$ is a vector with n 1s and $\mathbf{u}$ is an auxiliary vector. The augmented Lagrange function for Eq. (9) is as follows,

$$\begin{aligned} \mathcal{L}(\mathbf{r}, \mathbf{u}, \mu, \eta_1, \eta_2) &= \frac{\beta}{2}\mathbf{r}\mathbf{K}\mathbf{r}' - \mathbf{r}\mathbf{s}' + \frac{\mu}{2}\|\mathbf{r}\mathbf{1}_n' - 1\|^2 \\ &+ \frac{\mu}{2}\|\mathbf{r} - \mathbf{u}\|^2 + (\mathbf{r} - \mathbf{u})\eta_1' + (\mathbf{r}\mathbf{1}_n' - 1)\eta_2, s.t. u_i \geq 0 \end{aligned} \quad (10)$$

Algorithm 1 Optimization algorithm for Eq. (8)

Input: Sample-class similarity vector $\mathbf{s}$;
Sample-sample similarity matrix $\mathbf{K}$;
Output: Ranking score r_i for each sample;
1: Initialize: $\tau > 1$, $\mu > 0$, $r_i = s_i / \sum_{j=1}^{n} s_i$, $\mathbf{u} = \mathbf{r}$, $\eta_1 = \mathbf{0}_n$, and $\eta_2 = 0$;
2: **repeat**
3: Update $\mathbf{A} = \beta\mathbf{K} + \mu\mathbf{I}_n + \mu\mathbf{1}'\mathbf{1}$;
4: Update $\mathbf{b} = \mathbf{s} + \mu\mathbf{1}_n + \mu\mathbf{u} - \eta_1 - \eta_2\mathbf{1}_n$;
5: Update $\mathbf{r}$ by solving linear system $\mathbf{rA} = \mathbf{b}$;
6: Update $\mathbf{u}$ by Eq. (13);
7: Update η_1, η_2 and μ by Eq. (14);
8: **until** Convergence;
9: Return r_i;

where μ is a scalar, η_1 and η_2 are the Lagrange coefficients. To find the solution to Eq. (8), we just need to update the variables in $\mathcal{L}$ iteratively until convergence. The final $\mathbf{r}$ is the global optimum. Please refer to (Bertsekas 1999) for the proof. Specifically, the updating rules for them are as below.

Update r. Obviously, we have the following equivalence.

$$\min_{\mathbf{r}} \mathcal{L} \Leftrightarrow \min \frac{1}{2}\mathbf{rAr}' - \mathbf{rb}', \tag{11}$$

where $\mathbf{A} = \beta\mathbf{K} + \mu\mathbf{I}_n + \mu\mathbf{1}'\mathbf{1}$ and $\mathbf{b} = \mathbf{s} + \mu\mathbf{1}_n + \mu\mathbf{u} - \eta_1 - \eta_2\mathbf{1}_n$. The solution to the above unconstrained problem is given by solving a linear system $\mathbf{rA} = \mathbf{b}$. Apparently, $\mathbf{A}$ is a positive defined matrix and thus the linear system has a unique solution. We use the algorithm proposed by Spielman and Teng (2004) which gives a nearly linear complexity.

Update u. The Lagrange function $\mathcal{L}$ w.r.t. $\mathbf{u}$ is reduced to

$$\min_{u_i \geq 0} \mathcal{L} \Leftrightarrow \min_{u_i \geq 0} \frac{\mu}{2}\|\mathbf{r} - \mathbf{u}\|^2 + (\mathbf{r} - \mathbf{u})\eta_1', \tag{12}$$

and the solution to the nonnegativity-constrained problem is

$$u_i = \max(0, r_i + \eta_{1i}/\mu). \tag{13}$$

Update η_1, η_2 and μ. Following the pipeline of ALM framework, η_1, η_2 and μ are updated respectively as follows,

$$\eta_1 \leftarrow \eta_1 + \mu(\mathbf{r} - \mathbf{u}), \eta_2 \leftarrow \eta_2 + \mu(\mathbf{r}\mathbf{1}_n' - 1), \mu \leftarrow \tau\mu, \tag{14}$$

where $\tau > 1$ is a parameter. The optimization algorithm is shown in Algorithm 1. After applying the above steps, we select some samples from both labeled source samples and unlabeled target samples and assign target class c_j^t for them. We can perform the selection and assignment for each target class. Finally, we obtain a set of samples assigned by pseudo labels for all target classes and train classifiers with them.

Robust SVM

Based on the selected samples and pseudo labels, classifiers can be directly trained in the original feature space, and thus the classification can be achieved in one step from sample to class without using the attribute space as the intermediary. In this paper, we choose the SVM classifier (Cortes and Vapnik 1995) due to its superior performance. However, we need to notice the label noise caused by the fact that the pseudo labels are not the true labels. To achieve better performance, we modify the standard SVM in our scenario. Specifically, suppose we have totally m selected samples $\{\mathbf{x}_1, ..., \mathbf{x}_m\}$ and each sample has a pseudo label from $\mathcal{C}^t$. To handle the multi-class scenario, we train k_t one-vs-all classifiers where each classifier f_c treats class c as positive and the others as negative. With k_t classifiers, the final decision is given by

$$c(\mathbf{x}) = \text{argmax}_{c \in \mathcal{C}^t} f_c(\mathbf{x}). \tag{15}$$

To train the classifier $f_c (c \in \mathcal{C}^t)$, we construct the pseudo label vector $\mathbf{l}^c \in \{-1, 1\}^m$ where $l_i^c = 1$ if the sample $\mathbf{x}_i$ is assigned by the pseudo label c , or $l_i^c = -1$ otherwise. We consider the following dual formulation of SVM learning,

$$\begin{aligned} \min_{\alpha^c} \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i^c \alpha_j^c l_i^c l_j^c K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{m} \alpha_i^c \\ s.t.\ 0 \leq \alpha_i^c \leq C, \sum_{i=1}^{m} \alpha_i^c l_i^c = 0, \end{aligned} \tag{16}$$

where $K(\cdot, \cdot)$ is the kernel matrix. Given a test sample and the learned weights α^c, we have $f_c(\mathbf{x}) = \sum_i \alpha_i^c l_i^c K(\mathbf{x}_i, \mathbf{x})$. In the linear case where $K(\mathbf{x}_i, \mathbf{x}) = \mathbf{x}_i\mathbf{x}'$, we can set $\mathbf{w}_c = \sum_i \alpha_i^c l_i^c \mathbf{x}_i$ and then we have $f_c(\mathbf{x}) = \mathbf{x}\mathbf{w}_c'$. To address the unreliability of the pseudo labels, i.e., label noise, we need to modify the learning objective. In this paper, the label noise is modeled as the label flip (Xiao, Xiao, and Eckert 2012) where each label l_i^c has an i.i.d. probability to be the flipped version of the true label $\tilde{l}_i^c = l_i^c(1 - 2\epsilon_i)$ where ϵ_i is a binary variable with $p(\epsilon_i = 1) = \mu$ (flipped) and $p(\epsilon_i = 0) = 1 - \mu$ (not flipped). We have the expectation $\mathbb{E}[\epsilon] = \mu$ and the variance $\sigma^2 = \mu(1 - \mu)$. To take the flip into account, we can replace l_i^c with $l_i^c(1 - 2\epsilon_i)$. Denote $M_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) l_i^c l_j^c (1 - 2\epsilon_i)(1 - 2\epsilon_j)$, the expected value of M_{ij} w.r.t. ϵ_i is as below,

$$\mathbb{E}_\epsilon[M_{ij}] = \begin{cases} K(\mathbf{x}_i, \mathbf{x}_j) l_i^c l_j^c (1 - 4\sigma^2), & \text{if } i \neq j \\ K(\mathbf{x}_i, \mathbf{x}_j) l_i^c l_j^c, & \text{if } i = j. \end{cases} \tag{17}$$

Denote $\tilde{M}_{ij} = \mathbb{E}_\epsilon[M_{ij}]$. We can rewrite Eq. (16) as follows,

$$\min_{\alpha^c} \frac{1}{2}\alpha^c \tilde{\mathbf{M}} \alpha^{c'} - \alpha^c \mathbf{1}_m', s.t., 0 \leq \alpha_i^c \leq C, \alpha^c \mathbf{l}^{c'} = 0. \tag{18}$$

This formulation is intrinsically identical to the standard SVM dual formulation with a matrix $\tilde{\mathbf{M}}$, which can be efficiently solved by the existing tools, like LIBSVM (Chang and Lin 2011). In fact, we can observe that the label noise only influences the similarity between training samples in the dual formulation, i.e., $i \neq j$. Hence, Eq. (17) actually aims to decrease the similarity between samples to alleviate the influence of the label noise. In the robust SVM, we need to choose a parameter μ to construct $\tilde{\mathbf{M}}$. In fact, the large μ the more robust of SVM to noise is. But a larger μ may give rise to more information loss because it ignores the similarity between samples. On the other hand, if μ is too small, the noise may affect SVM classification significantly. The effect of μ on our approach will be shown in the coming section.

Table 2: The results on four benchmark datasets. The symbol ‡ indicates that the approach is in the transductive setting.

Approach	CIFAR10	Animal with Attributes	aPascal-aYahoo	SUN
Socher et al. 2013	74.82 ± 0.88			
Lampert, Nickisch, and Harmeling 2014		40.5	19.1	52.50
Fu et al. 2015		66.0		
Romera-Paredes and Torr 2015	81.22 ± 1.04	75.32 ± 2.28	24.22 ± 2.89	80.10 ± 0.32
Zhang and Saligrama 2015	86.27 ± 0.65	76.72 ± 0.83	42.90 ± 0.73	79.50 ± 1.22
Al-Halah, Tapaswi, and Stiefelhagen 2016		67.5	37.0	
Changpinyo et al. 2016	84.52 ± 1.07	74.81 ± 0.42	41.77 ± 0.50	78.82 ± 1.16
Fu et al. 2014‡		77.8		
Li and Guo 2015‡		52.06 ± 1.52	25.98 ± 1.19	66.80 ± 0.72
Li, Guo, and Schuurmans 2015‡		56.88 ± 1.74	27.02 ± 1.25	69.21 ± 0.45
Kodirov et al. 2015‡		75.6	26.5	
Guo et al. 2016‡	86.30 ± 0.77	78.47 ± 1.06	39.03 ± 0.77	82.00 ± 0.57
Ours-Inductive	89.52 ± 0.29	79.07 ± 0.58	43.59 ± 0.42	80.04 ± 0.19
Ours-Transductive‡	92.79 ± 0.34	82.90 ± 0.77	50.84 ± 0.81	84.42 ± 0.17

Initialization and Iterative Refinement

In the transductive setting where the unlabeled target samples are available, to solve Eq. (4), the estimated labels $\tilde{c}(\mathbf{x}_j)$ should be given. However, there is no model to estimate them at first. In this paper, we propose to use an iterative refinement procedure to address the initialization problem. Specifically, at the first iteration, we ignore the second part in Eq. (4) and learn the projection only with the labeled source samples. Then we perform the sample selection, pseudo label assignment, and robust SVM training sequentially. Having obtained the classifiers for target classes, the estimated labels can be generated. Next, we can solve Eq. (4) based on the target samples and estimated labels and afterwards it can result in a better projection. With a better projection, the whole procedure is re-executed and we normally expect to generate more effective classifiers for target classes which refine the estimated labels. Therefore, we propose to iteratively refine the estimated labels and models until convergence. In the coming experiment, we will demonstrate that the iterative refinement can always lead to better results. On the other hand, in the inductive setting where no target sample is available. We just need to ignore the second part in Eq. (4) and run the procedure by only one iteration.

Experiment

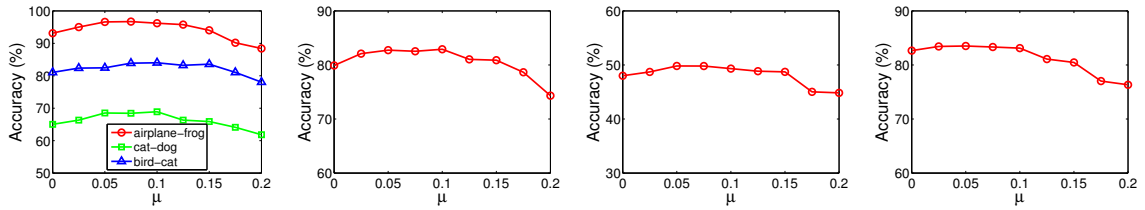
Settings

We conduct experiments on four datasets. The first one is CIFAR10 (Krizhevsky 2009) which has 10 object classes. There are $6,000$ images in each class. Following (Socher et al. 2013), in each split, we select 2 classes as the target classes and the other 8 as the source classes, and thus we have $C_{10}^2 = 45$ different splits. The second database is Animal with Attributes (AwA) (Lampert, Nickisch, and Harmeling 2014) dataset which consists of 50 animal classes and $30,475$ images. Following the split suggested in (Lampert, Nickisch, and Harmeling 2014), 40 classes with $24,295$ images are adopted as the source classes and 10 classes with $6,180$ images are adopted as the target classes. The third one is aPascal-aYahoo (aPY) dataset (Farhadi et al. 2009), in which aPascal has 20 objects designed for PASCAL VOC2008 challenge, such as "people" and "dog". It contains in total $12,695$ images. aYahoo dataset was collected from Yahoo image search. It has 12 classes which are similar but different from the ones in aPascal, such as "centaur" and "wolf". It contains $2,644$ images. We regard aPascal as the source classes and aYahoo as the target classes. The last one is SUN scene recognition dataset (Patterson and Hays 2012). It has 717 scenes and each scene has 20 images. We use 707 classes as the source and 10 as the target following the setting from (Jayaraman and Grauman 2014). For CIFAR10, the class attributes are the 50-dimensional word representation learned by (Huang et al. 2012). For AwA, aPY, and SUN, we use the attributes provided by the dataset. For each image, we utilize the 4096-dimensional features from fc7 layer of the pre-trained AlexNet (Krizhevsky, Sutskever, and Hinton 2012). Finally, the performance is evaluated by the multi-class classification accuracy on the target classes.

To implement our approach, we use the following setting. As introduced above, we select samples for each target class individually. In the inductive setting, for each target class, we select $m_s = 500, 500, 200, 200$ source samples for CIFAR10, AwA, aPY, and SUN respectively. In the transductive setting, we further select $m_t = 500, 200, 50, 10$ from unlabeled target samples. In addition, to determine the parameter value for β for sample selection, μ and C for training robust SVM, we adopt the class-wise cross-validation strategy (Zhang and Saligrama 2015; Guo et al. 2016) where β and C are chosen from $\{10^{-2}, 10^{-1}, 1, 10, 10^2\}$ and μ is from $\{0, 0.025, 0.05, ..., 0.2\}$. We report the results of both inductive and transductive versions of our approach and compare them to the baseline approaches from both settings.

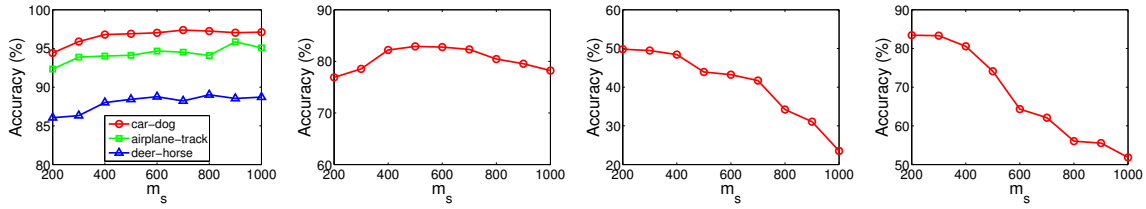
Benchmark Comparison

The results on four benchmark datasets are shown in Table 2. We can observe that the proposed approach outperforms the other two-step approaches with statistical significance in both inductive and transductive settings. This clearly reveals that the one-step classification strategy with direct classifiers learned by transferred samples and pseudo labels is indeed



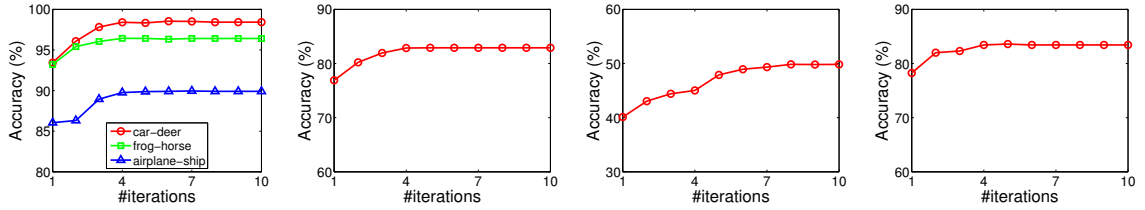
(a) CIFAR10 (b) AwA (c) aPY (d) SUN

Figure 3: The effect of μ.



(a) CIFAR10 (b) AwA (c) aPY (d) SUN

Figure 4: The effect of m_s, the number of samples selected from source classes for each target class.



(a) CIFAR10 (b) AwA (c) aPY (d) SUN

Figure 5: The classification accuracy w.r.t. the number of iterations.

better than the two-step strategy because it suffers from less information loss. Meanwhile, it also proves that our sample selection algorithm can choose samples from source classes that can well capture the characteristics of the target class.

More Results

Now we investigate some important issues of our approach. Because of the space limitation, we only discuss the transductive version. In fact, the inductive one has similar trends.

We first investigate how our approach behaves when varying of μ in robust SVM. The results w.r.t. different values of μ on four datasets are are shown in Figure 3. On the one hand, if μ is too small, the label noise will affect the quality of the classifiers. On the other hand, if μ is too large, the relationship between samples will be neglected such that the information may be not enough to train effective classifiers. Empirically, we can choose $\mu \in [0.05, 0.1]$ for our approach.

The influence of the number of selected samples from source classes for each target class, i.e., m_s, is presented in Figure 4. For CIFAR10, the increase of performance is proportional to the rise in selected sample number. For AwA, more samples lead to better performance when $m_s < 600$ but lead to worse performance when $m_s > 600$. For the other two datasets, increasing m_s may result in worse performance. The reason is as follows. The main assumption for our approach is that there exists some samples in source classes that are very **similar** to target classes and thus we can use them to capture the characteristics of target classes, as illustrated in Figure 2. For CIFAR10, it has a large candidate set and thus there may exist a lot of good samples such that selecting more samples leads to better result. However, in aPY and SUN, the candidate set is small, so that there is only a few good samples ($m_s < 400$). When m_s is large, many dissimilar (bad) samples are mistakenly selected such that they fail to effectively describe the target classes. We leave how to determine m_s automatically to our future work.

In the transductive setting, we propose an iterative algorithm to progressively refine the estimated labels and the models. In Figure 5, we plot the accuracy on the test set w.r.t. the number of iterations. Obviously, the accuracy increases steadily at beginning and remains stable after 10 iterations, which validates the effectiveness of the iterative refinement.

Conclusion

In this paper, we propose a novel one-step approach for ZSR which directly trains classifiers for target classes. In contrast to the existing two-step approaches, our approach does not need the attribute space as the intermediary during the classification (test) stage, thus avoiding the information loss. As there is no labeled sample for target classes, we propose to select samples based on their reliability and diversity and assign pseudo labels to them. We formulate it into a quadratic formulation and solve it under the ALM framework. To address the unreliability of pseudo labels, we propose to train a robust SVM classifier derived from a standard SVM. Experiment on benchmarks demonstrates that the proposed approach is superior to the state-of-the-art ZSR approaches.

References

Akata, Z.; Perronnin, F.; Harchaoui, Z.; and Schmid, C. 2013. Label-embedding for attribute-based classification. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 819–826.

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

Belkin, M., and Niyogi, P. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, 585–591.

Bertsekas, D. 1999. *Nonlinear programming*. Belmont,MA: Athena Scientific.

Bishop, C. M., et al. 2006. *Pattern recognition and machine learning*, volume 1. Springer, New York.

Chang, C., and Lin, C. 2011. LIBSVM: A library for support vector machines. *ACM TIST* 2(3):27.

Changpinyo, S.; Chao, W.; Gong, B.; and Sha, F. 2016. Synthesized classifiers for zero-shot learning. In *2016 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Machine Learning* 20(3):273–297.

Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. A. 2009. Describing objects by their attributes. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1778–1785.

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 *27th Annual Conference on Neural Information Processing Systems 2013*, 2121–2129.

Fu, Y.; Hospedales, T. M.; Xiang, T.; Fu, Z.; and Gong, S. 2014. Transductive multi-view embedding for zero-shot recognition and annotation. In *Computer Vision - ECCV 2014 - 13th European Conference*, 584–599.

Fu, Z.; Xiang, T.; Kodirov, E.; and Gong, S. 2015. Zero-shot object recognition by semantic manifold distance. In *2015 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Guo, Y.; Ding, G.; Jin, X.; and Wang, J. 2016. Transductive zero-shot recognition via shared model space learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*.

Huang, E. H.; Socher, R.; Manning, C. D.; and Ng, A. Y. 2012. Improving word representations via global context and multiple word prototypes. In *The 50th Annual Meeting of the Association for Computational Linguistics*, 873–882.

Jayaraman, D., and Grauman, K. 2014. Zero-shot recognition with unreliable attributes. In *Annual Conference on Neural Information Processing Systems 2014*, 3464–3472.

Kodirov, E.; Xiang, T.; Fu, Z.; and Gong, S. 2015. Unsupervised domain adaptation for zero-shot learning. In *IEEE International Conference on Computer Vision*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, 1106–1114.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. *Tech Report. Univ. of Toronto*.

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 951–958.

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2014. Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. Pattern Anal. Mach. Intell.* 36(3):453–465.

Li, X., and Guo, Y. 2015. Max-margin zero-shot learning for multi-class classification. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*.

Li, X.; Guo, Y.; and Schuurmans, D. 2015. Semi-supervised zero-shot classification with label representation learning. In *IEEE International Conference on Computer Vision*.

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 *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2751–2758.

Romera-Paredes, B., and Torr, P. H. S. 2015. An embarrassingly simple approach to zero-shot learning. In *Proceedings of the 32nd International Conference on Machine Learning*, 2152–2161.

Socher, R.; Ganjoo, M.; Manning, C. D.; and Ng, A. Y. 2013. Zero-shot learning through cross-modal transfer. In *27th Annual Conference on Neural Information Processing Systems 2013*, 935–943.

Spielman, D. A., and Teng, S. 2004. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, 81–90.

Xiao, H.; Xiao, H.; and Eckert, C. 2012. Adversarial label flips attack on support vector machines. In *ECAI 2012 - 20th European Conference on Artificial Intelligence*, 870–875.

Zhang, Z., and Saligrama, V. 2015. Zero-shot learning via semantic similarity embedding. In *IEEE International Conference on Computer Vision*.