Zero-Shot Recognition via Structured Prediction

Ziming Zhang(✉) and Venkatesh Saligrama

Department of Electrical and Computer Engineering,
Boston University, Boston, USA
{zzhang14,srv}@bu.edu

Abstract. We develop a novel method for zero shot learning (ZSL) based on test-time adaptation of similarity functions learned using training data. Existing methods exclusively employ source-domain side information for recognizing unseen classes during test time. We show that for batch-mode applications, accuracy can be significantly improved by adapting these predictors to the observed test-time target-domain ensemble. We develop a novel structured prediction method for maximum a posteriori (MAP) estimation, where parameters account for test-time domain shift from what is predicted primarily using source domain information. We propose a Gaussian parameterization for the MAP problem and derive an efficient structure prediction algorithm. Empirically we test our method on four popular benchmark image datasets for ZSL, and show significant improvement over the state-of-the-art, on average, by 11.50 % and 30.12 % in terms of accuracy for recognition and mean average precision (mAP) for retrieval, respectively.

Keywords: Zero-shot learning/recognition/retrieval · Structured prediction · Maximum likelihood estimation

1 Introduction

Zero-shot recognition (ZSR) is the problem of recognizing data instances from *unseen* classes (*i.e.* no training data for these classes) during test time. The motivation for ZSR stems from the need for solutions to diverse research problems ranging from poorly annotated big data collections [1] to the problem of extreme classification [2]. In this paper we consider the classical ZSL setting. Namely, we are given two sources of data the so called *source domain* and *target domain*, respectively. In the source domain, each class is represented by a *single* vector of side information such as attributes [3–7], language words/phrases [8–10], or even learned classifiers [11]. In target domain, each class is represented by a collection of data instances (*e.g.* images or videos). During training some known classes with data are given as *seen classes*, while during testing some other unknown classes are revealed as unseen classes. The goal of ZSL is to learn suitable models using seen class training data so that in ZSR the class labels of arbitrary target domain data instances from unseen classes during testing can be predicted.

B. Leibe et al. (Eds.): ECCV 2016, Part VII, LNCS 9911, pp. 533–548, 2016.
DOI: 10.1007/978-3-319-46478-7_33

Key Insight: In batch mode we are given the ensemble of target domain data. Our main idea is that even though labels for target-domain data are unknown, subtle shifts in the data distributions can be inferred and these shifts can in turn be utilized to better adapt the learned classifiers for test-time use.

Intuitively, our insight is justified by noting that target domain data instances could form compact and disjoint clusters in their latent space embeddings. These clusters can be reliably separated into different seen or unseen classes. Nevertheless, the predicted locations of clusters based on source domain data are somewhat inaccurate, resulting in large errors. Consequently, we can improve accuracy by adapting to target domain ensemble distribution in test time.

Another perspective on this issue can be gleaned from Fig. 1, which depicts the CNN feature distribution for the 12 "unseen" classes in the aPascal & aYahoo (aP&Y) dataset [3]. As we see there exist clear gaps between most of the class pairs, indicating that CNN features are sufficiently reliable to recognize these classes. Indeed a linear multi-class support vector machine (SVM) would suffice if we were given even a few instances from the unseen dataset. By using half of unseen data for training the recognition performance on the remaining data is as high as 97 %. Nevertheless, the best known result in the ZSL literature is around 50 % [13]. This huge performance difference, *i.e.* 97 % − 50 % = 47 %, suggests that the estimated unseen class classifiers are inaccurate to some extent compared with the supervised classifiers.

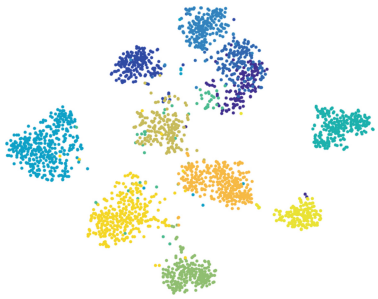

Fig. 1. t-SNE [12] visualization of CNN features for the 12 unseen classes in aP&Y dataset, one color per class. (Color figure online)

Obviously, there are many reasons for the significant performance degradation. First and foremost is that we have no access to the labels for unseen classes and obviously no training instances for them. In addition this difference can also stem from inaccurate source domain attribute vectors, noisy data in target domain during training, imbalanced data distributions, *etc.* Among them one of plausible reasons could be the *projection domain shift* problem, which has been investigated recently [14,15]. The major argument here is that the test-time data distributions in the projection/latent space could be different from the estimation based on training data, and as a result the learned ZSL classifiers for unseen classes cannot work well. This leads us to question that is the focus of this paper, namely, *is it possible to improve the recognition performance of the estimated classifiers for unseen classes if we posit that the unseen class target data forms nice clusters?*

In this paper, we propose a structured prediction approach for ZSR, by assuming that the unseen data can be visually clustered but that the predicted locations from the training data can be somewhat inaccurate. Our idea arises from the following two perspectives: The first perspective is from unsupervised

data clustering, where we attempt to capture the correct underlying distribution in the latent space for each unseen class[1]. Given clustered features such as CNN, it is reasonable to assume that data instances in each cluster should have the same class label as in label propagation (*e.g.* [16]). The second perspective is based on data assignment, which in our case is a bipartite graph matching problem with vertices representing clusters and unseen classes on each side, respectively. The edge weights between these vertices represent the (weighted) average similarities between the data instances in each cluster and unseen class classifiers. This perspective suggests that rather than predicting class label individually we seek to recognize the class label at the cluster level, a viewpoint closely related to multiple instance learning (MIL) [17]. Both aspects aim to globally predict a suitable data structure for unseen classes and utilize it to improve the recognition performance in an unsupervised manner.

Our approach is based on a novel structured prediction method, which in essence is equivalent to maximum a posteriori (MAP) estimation. Further we propose a Gaussian parameterization for batch-mode ZSR, and accordingly derive an efficient algorithm for ZSR. The parameters accounting for test-time shift are adaptive to test data based on the learned associations between source domain attribute vectors and target domain images. Empirically we test our method on four popular benchmark image datasets for ZSL, namely, aPascal & aYahoo (aP&Y) [3], Animals with Attributes (AwA) [18], Caltech-UCSD Birds-200-2011 (CUB) [19], and SUN Attribute (SUN) [20], and achieve the state-of-the-art.

1.1 Related Work

A significant number of works for zero-shot learning are based on learning attribute classifiers that map target domain instances to those in source domain [4,11,21–27]. More recently methods based on similarity learning using linear [9,10,28–32] or nonlinear kernels [13,14,33,34] on source and target domain embeddings have been proposed. There also exist other approaches such as transfer learning [35], multimodel learning [36], multi-view learning [37], multi-domain and multi-task learning [38], that have been applied to zero-shot learning [38]. In general these learning methods can suffer from data noise (*e.g.* intra-class variability, inter-class similarity, noisy ground-truth attribute vectors, *etc.*) leading to performance degradation during test-time recognition of unseen classes.

Recently researchers have begun to incorporate test-time unseen class data into ZSL as unlabeled data to handle the projection domain shift problem [14,15]. In [14] an unsupervised domain adaption was proposed, where the target domain class label projections are utilized as regularization in a sparse coding framework to learn the target domain projection. A separate classifier such as nearest neighbor or semi-supervised label propagation is used as a post-step for recognition with the learned target domain projection. In [15] an approach

[1] For simplicity, in this paper we assume that there is only one cluster per class. With slight modification our method can also work in the cases where multiple clusters could correspond to one unseen class.

based on transductive multi-class and multi-label ZSL is proposed. The idea there is to align the unlabeled data in the feature space with multiple semantic views through multi-view canonical correlation analysis and then recognize these data instances using label propagation. Underlying these methods is the need to account for target domain unseen class data structure in the learning procedure. This has led to improvement in ZSL performance.

In contrast to these previous ZSL approaches which cannot accept trained classifiers as inputs, our method specifically focuses on the recognition task for batch-mode test time processing. Potentially our method can be used in conjunction with any similarity learning procedure trained on seen-class data and can score similarity between unseen classes and target domain data instances. We pursue our goal by formulating ZSR as a bipartite graph matching structured prediction problem. Our aim is to find the best assignment matrix between data instances and unseen classes.

While label propagation (*e.g.* [16]) and certain multi-class classification methods (*e.g.* CoConut [39]) are closely related to ours, they do not incorporate data/domain shift, which is fundamental. We account for domain shift by proposing a novel *joint structured prediction* problem in test time that accounts for unseen-class data structure (*i.e.* clustering) and label assignment. This is the first such work that like CoConut utilizes existing trained classifiers for scoring prior similarity but in addition deals with data-shift arising in ZSR.

Also our method is different from active learning, which can select data samples and acquire labels to learn classifiers. In contrast in ZSR labels cannot be acquired. In addition our method does no learning. It is a structured prediction test time method for labelling unseen unlabelled instances.

2 Zero-Shot Recognition via Structured Prediction

2.1 Problem Setting

(i) ZSL in training: Our method for training predictors using seen class training data resembles many past approaches (see [13]). Let $\{\mathbf{x}_c^{(s)}\}_{c=1,\cdots,C}$ and $\{\mathbf{x}_i^{(t)}, y_i\}_{i=1,\cdots,N}$ denote the training data for source and target domains, respectively. Here $\mathbf{x}_c^{(s)} \in \mathbb{R}^{d_s}, \forall\, c \in [C]$ is the d_s-dim attribute vector for class c; $\mathbf{x}_i^{(t)} \in \mathbb{R}^{d_t}, \forall i$ is a d_t-dim data instance with class label $y_i \in [C]$ for $i \in [N]$. We learn two projection functions $\phi_s : \mathbb{R}^{d_s} \rightarrow \mathbb{R}^{D_s}$ and $\phi_t : \mathbb{R}^{d_t} \rightarrow \mathbb{R}^{D_t}$ for source and target domains, respectively, to minimize the binary prediction loss:

$$\min_{\kappa \in \mathcal{K}, \phi_s \in \Phi_s, \phi_t \in \Phi_t} \sum_{c=1}^{C} \sum_{i=1}^{N} \ell \left(\kappa(\phi_s(\mathbf{x}_c^{(s)}), \phi_t(\mathbf{x}_i^{(t)})), \mathbf{1}_{\{c=y_i\}} \right), \tag{1}$$

where $\mathcal{K}, \Phi_s, \Phi_t$ denote the corresponding feasible functional spaces, $\kappa : \mathbb{R}^{D_s} \times \mathbb{R}^{D_t} \rightarrow \mathbb{R}$ denotes a similarity function, $\mathbf{1}_{\{c=y_i\}}$ denotes an indicator function returning 1 if the condition $c = y_i$ holds, otherwise -1, and $\ell : \mathbb{R} \times \{-1, +1\} \rightarrow \mathbb{R}$ denotes a loss function (*e.g.* hinge loss).

(ii) Online-mode ZSR in testing: We briefly describe this mode in order to contrast our batch-mode setup of ZSR. As is the convention, in this mode, we are given C' source domain unseen class attribute vectors $\bar{\mathcal{X}}^{(s)} = \{\mathbf{x}_{c'}^{(s)}\}_{c'=1,\cdots,C'}$ and a single data instance, $\mathbf{x}_{i'}^{(t)}$, chosen uniformly at random from a collection of N' target domain unseen class data instances $\bar{\mathcal{X}}^{(t)} = \{\mathbf{x}_{i'}^{(t)}\}_{i'=1,\cdots,N'}$. The goal is to match this instance to one of the C' unseen source-domain descriptions. Given the learned similarity kernel κ and the source and target domain embedding functions, the problem reduces to a multi-class classification rule.

$$y_{i'} = \underset{c' \in \{1,\cdots,C'\}}{\arg\max}\ P_\theta(c'|\mathbf{x}_{c'}^{(s)}, \mathbf{x}_{i'}^{(t)}) \equiv \underset{c' \in \{1,\cdots,C'\}}{\arg\max}\ \kappa(\phi_s(\mathbf{x}_{c'}^{(s)}), \phi_t(\mathbf{x}_{i'}^{(t)})), \tag{2}$$

As depicted above we can view the similarity kernel as a probability functional. P_θ denotes the probability of being labeled as c' given the data pair parameterized by θ.

(iii) Batch-mode ZSR in testing: In contrast, our method is based on batch-mode processing. Here during test-time all the N' target-domain unseen class instances are revealed and our task is to match these N' target domain instances to C' source domain descriptions. Our goal is thus to predict a good global structure, $\bar{\mathcal{Y}}$, among the predicted labels simultaneously by exploring useful data dependencies for unseen classes in both source and target domains, rather than in isolation as in the online-mode. We can view this problem probabilistically as attempting to jointly label all instances conditioned on combined (but unassociated) source/target test data:

$$\bar{\mathcal{Y}} = \underset{\omega \in \Omega}{\arg\max}\, P_\theta(\omega|\bar{\mathcal{X}}^{(s)}, \bar{\mathcal{X}}^{(t)}), \tag{3}$$

where $\omega \in \Omega$ denotes a feasible assignment solution between target data and source attribute vectors (and hence unseen class labels). If one were to utilize primarily the similarity function learned on seen class training data the problem reduces to the standard bipartite matching problem. In any case this approach is infeasible due to lack of knowledge of the number of instances corresponding to each class. Regardless we hope to do better by utilizing target-domain batch data (although unlabeled/unassociated) to improve these assignments. Note that the prediction functions described in [14,15] that use unseen class data as unlabeled data can be abstractly represented in this way.

2.2 Structured Prediction in Testing

We propose a structured prediction method for batch-mode ZSR. Intuitively a good labeling structure for target domain unseen class data instances should result in *smooth* label assignments in the latent space. Namely, two close data points tend to have the same class label. To predict smooth labeling structures we consider an approach based on fusing information obtained from cross-domain similarities with empirically observed target domain data distribution.

(i) Maximum a posteriori (MAP) estimation: We will develop a generative parameterized probabilistic model for recognizing test-time target data and describe an approach based on MAP. Using Bayes' rule we can further expand the batch-mode decision rule in Eq. 3 as follows:

$$\bar{\mathcal{Y}} = \arg\max_{\omega\in\Omega} \sum_{c'=1}^{C'}\sum_{i'=1}^{N'} P_\theta(\omega_{c',i'}|\bar{\mathcal{X}}^{(s)},\bar{\mathcal{X}}^{(t)}) = \arg\max_{\omega\in\Omega} \sum_{c'=1}^{C'}\sum_{i'=1}^{N'} P_\theta(\omega_{c',i'},\bar{\mathcal{X}}^{(s)},\bar{\mathcal{X}}^{(t)})$$
$$= \arg\max_{\omega\in\Omega} \sum_{c'=1}^{C'}\sum_{i'=1}^{N'} P_\theta(\omega_{c',i'}) P_\theta(\bar{\mathcal{X}}^{(s)}|\omega_{c',i'}) P_\theta(\bar{\mathcal{X}}^{(t)}|\omega_{c',i'}), \tag{4}$$

where $\omega_{c',i'}$ denotes data $\mathbf{x}_{i'}^{(t)}$ being labeled as unseen class c', $P_\theta(\omega_{c',i'})$ denotes the prior distribution, and $P_\theta(\bar{\mathcal{X}}^{(s)}|\omega_{c',i'}), P_\theta(\bar{\mathcal{X}}^{(t)}|\omega_{c',i'})$ denote the likelihoods of generating data sets $\bar{\mathcal{X}}^{(s)}, \bar{\mathcal{X}}^{(t)}$ given the assignment and parameter θ, respectively. Note that our MAP formulation corresponds to the online-mode ZSR if we remove $P_\theta(\bar{\mathcal{X}}^{(t)}|\omega_{c',i'})$ from Eq. 4 and assume ω is a one-to-one assignment function.

We view $P_\theta(\omega_{c',i'},\bar{\mathcal{X}}^{(s)},\bar{\mathcal{X}}^{(t)})$ as a generative model that models the likelihood of labeling data $\mathbf{x}_{i'}^{(t)}$ as unseen class c' in the context of source and target data $\bar{\mathcal{X}}^{(s)},\bar{\mathcal{X}}^{(t)})$. We posit that the data generation process for source and target domains is conditionally independent given the assignment variable. Consequently, we can factorize the likelihood as the last line in Eq. 4.

Empirically we would like to maximize the log-likelihood as many Bayesian methods [40] do. Therefore, rather than optimizing Eq. 4 directly we prefer optimizing the lower bound of the log-likelihood for structured prediction:

$$\bar{\mathcal{Y}} = \arg\max_{\omega\in\Omega} \sum_{c'=1}^{C'}\sum_{i'=1}^{N'} P_\theta(\omega_{c',i'}) \left[\log P_\theta(\bar{\mathcal{X}}^{(s)}|\omega_{c',i'}) + \log P_\theta(\bar{\mathcal{X}}^{(t)}|\omega_{c',i'})\right]. \tag{5}$$

(ii) Parameterization: We parameterize the log-likelihoods in Eq. 5 with Gaussian models. For source domain, we directly utilize the similarity between data $\mathbf{x}_{i'}^{(t)}$ and unseen class c' with learned functions κ, ϕ_s, ϕ_t as follows:

$$\log P_\theta(\bar{\mathcal{X}}^{(s)}|\omega_{c',i'}) \stackrel{\text{def}}{=} \lambda_s \kappa(\phi_s(\mathbf{x}_{c'}^{(s)}), \phi_t(\mathbf{x}_{i'}^{(t)})), \tag{6}$$

with predefined parameter $\lambda_s \geq 0$. For target domain, we utilize the distance between the projected data $\phi_t(\mathbf{x}_{i'}^{(t)})$ and the *empirical* mean vector $\boldsymbol{\mu}_{c'}^{(t)}$ for unseen class c' in the same latent space by setting parameter $\theta = \{\boldsymbol{\mu}_{c'}^{(t)}\}$. That is,

$$\log P_\theta(\bar{\mathcal{X}}^{(t)}|\omega_{c',i'}) \stackrel{\text{def}}{=} -\lambda_t \|\phi_t(\mathbf{x}_{i'}^{(t)}) - \boldsymbol{\mu}_{c'}^{(t)}\|_2^2, \tag{7}$$

with another predefined parameter $\lambda_t \geq 0$ and $\|\cdot\|_2$ denoting the ℓ_2 norm operator of a vector.

(iii) Initial model for estimating ω and θ: In order to account for target data distribution efficiently, we initialize θ as a set of cluster centers generated

from K-means with $K = C'$. Then we identify one-to-one matches between the clusters and unseen classes so that we can label the data instances in each cluster using the matched class label as the initialization of parameter ω.

To identify the matches, we solve the following binary assignment problem:

$$\max_{\{\bar{B}_{c',k'}\}} \sum_{c'=1}^{C'} \sum_{k'=1}^{C'} \bar{S}_{c',k'} \bar{B}_{c',k'}, \text{ s.t. } \forall c', \forall k', \sum_{c'} \bar{B}_{c',k'} = 1, \sum_{k'} \bar{B}_{c',k'} = 1, \tag{8}$$

where $\bar{B}_{c',k'} \in \{0,1\}, \forall c', \forall k'$, denotes the binary assignment variable, and $\bar{S}_{c',k'}$ denotes the average similarities between unseen class c' and data in cluster k'. This problem can be efficiently solved using linear programming (LP).

(iv) Complete model: In fact each parameter $\boldsymbol{\mu}_{c'}^{(t)}$ in Eq. 7 can be estimated as the weighted means of all the projected target domain features in the latent space for class c'. Importantly this estimation is coupled with parameter ω, as ω describes the relationship between target data and unseen classes.

We denote as $\mathbf{S} \in \mathbb{R}^{C' \times N'}$ the test-time source-target data similarity matrix where $S_{c',i'} = \kappa(\phi_s(\mathbf{x}_{c'}^{(s)}), \phi_t(\mathbf{x}_{i'}^{(t)})), \forall c', \forall i'$ is the (c', i')-th entry in matrix $\mathbf{S}$. We denote as $\bar{\boldsymbol{\Phi}}_t \overset{\text{def}}{=} [\phi_t(\mathbf{x}_{i'}^{(t)})]_{i'=1,\cdots,N'} \in \mathbb{R}^{d_t \times N'}$ the target domain data matrix consisting of each instance $\phi_t(\mathbf{x}_{i'}^{(t)}), \forall i'$, as a column. We denote as $P_\theta(\omega) \overset{\text{def}}{=} \mathbf{H} \in \mathbb{R}^{C' \times N'}$ the source-target assignment weighting matrix. We denote as $\mathbf{S}_{c'} \in \mathbb{R}^{1 \times N'}, \mathbf{H}_{c'} \in \mathbb{R}^{1 \times N'}, \forall c'$, the c'-th rows in $\mathbf{S}$ and $\mathbf{H}$, respectively. Then by substituting Eqs. 6 and 7 into Eq. 5, we can write down our regularized structured prediction objective for ZSR as follows:

$$\min_{\mathbf{H}, \{\boldsymbol{\mu}_{c'}^{(t)} = \bar{\boldsymbol{\Phi}}_t \mathbf{H}_{c'}^T\}} \frac{1}{2}\|\mathbf{H}\|_F^2 - \lambda_s \sum_{c'=1}^{C'} \mathbf{S}_{c'} \mathbf{H}_{c'}^T + \lambda_t \sum_{i'=1}^{N'} \sum_{c'=1}^{C'} H_{c',i'} \|\phi_t(\mathbf{x}_{i'}^{(t)}) - \boldsymbol{\mu}_{c'}^{(t)}\|_2^2 \tag{9}$$

$$\text{s.t.} \forall i', \forall c', H_{c',i'} \geq 0, \sum_{c'=1}^{C'} H_{c',i'} \neq 0, \sum_{i'=1}^{N'} H_{c',i'} = 1, \forall c'_m \neq c'_n, \sum_{i'=1}^{N'} H_{c'_m,i'} H_{c'_n,i'} = 0,$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, and $(\cdot)^T$ denotes the matrix transpose operator. Here the constraints guarantee that: (1) Every instance is assigned to at least one unseen class, and for each unseen class, each row in $\mathbf{H}$ represents a probability distribution over all the instances (on a simplex); (2) The additional orthogonality constraints ensure that every instance is assigned to only one unseen class.

Note that all the assignment constraints for minimizing Eq. 9 are chosen to reflect the fact that we know a priori that in test time every instance must belongs to a single unseen class. Nevertheless, our method can be extended to handle missing matches between source and target domain data by suitably modifying the bipartite graph matching constraints. In reality these missing-match scenarios in ZSR may be more interesting and important, but they are outside the scope of this paper.

(v) Optimization: Solving Eq. 9 is nontrivial as it is highly non-convex. In Algorithm 1 we propose an efficient alternating optimization algorithm to solve

Algorithm 1. Structured prediction in test time for ZSR

Input : cross-domain similarity matrix $\mathbf{S}$, predefined parameters $\lambda_s \geq 0, \lambda_t \geq 0$
Output: Source-target domain binary assignment matrix $\mathbf{B}$

Initialize matrix $\mathbf{B}$ using K-means and find the cluster-class matches using Eq. 8;
repeat
  foreach c' **do**
    Update the c'-th row in matrix $\mathbf{Z}$ by solving Eq. 10;
  end
  $\mathbf{H} \leftarrow \mathbf{B} \circ \mathbf{Z}$; $\forall c', \boldsymbol{\mu}_{c'}^{(t)} = \bar{\boldsymbol{\Phi}}_t \mathbf{H}_{c'}^T$;
  foreach i' **do**
    Update the i'-th column in matrix $\mathbf{B}$ by solving Eq. 11;
  end
until *Certain stop criterion is satisfied*;
return $\mathbf{B}$;

Eq. 9 sub-optimally. The idea here is to decompose $\mathbf{H} = \mathbf{B} \circ \mathbf{Z}$, where $\circ$ denotes the entry-wise multiplication operator between two matrices, $\mathbf{B} \in \{0,1\}^{C' \times N'}$ is a binary matrix indicating the assignments and $\mathbf{Z} \in \mathbb{R}^{C' \times N'}$ is a weighting matrix for the corresponding assignments. When $\mathbf{B}$ is learned and fixed, we can solve a weighting problem for each unseen class using quadratic programming (QP). Letting $\mathcal{J}_{c'}$ be the index set where $\forall j' \in \mathcal{J}_{c'}$ the (c', j')-th entry in $\mathbf{B}$ is 1, we can optimize Eq. 9 as follows: $\forall c'$,

$$\min_{\{Z_{c',j'}\}} \frac{1}{2} \sum_{j' \in \mathcal{J}_{c'}} Z_{c',j'}^2 - \lambda_s \sum_{j' \in \mathcal{J}_{c'}} S_{c',j'} Z_{c',j'} + \lambda_t \sum_{j' \in \mathcal{J}_{c'}} Z_{c',j'} \|\phi_t(\mathbf{x}_{j'}^{(t)}) - \boldsymbol{\mu}_{c'}^{(t)}\|_2^2$$
$$\text{s.t.} \quad \forall j' \in \mathcal{J}_{c'}, Z_{c',j'} \geq 0, \sum_{j'} Z_{c',j'} = 1, \tag{10}$$

where $Z_{c',j'}$ is the (c', j')-th entry in $\mathbf{Z}$. This leads to a sub-optimal solution $\mathbf{H} = \mathbf{B} \circ \mathbf{Z}$. Next, we estimate the binary assignment variable $\mathbf{B}$ only based on the distance term in Eq. 9, which is equivalent to a nearest neighbor problem as shown below:

$$\forall i', \; B_{y',i'} = \begin{cases} 1, \text{if } y' = \arg\min_{c'} \|\phi_t(\mathbf{x}_{i'}^{(t)}) - \boldsymbol{\mu}_{c'}^{(t)}\|_2^2, \\ 0, \text{otherwise}, \end{cases} \tag{11}$$

where $B_{y',i'}$ is the (y', i')-th entry in $\mathbf{B}$. This step guarantees the orthogonality constraints in Eq. 9. We repeat this procedure until certain stop criterion is satisfied (*i.e.* number of iterations). Empirically our algorithm works well even with very few iterations, although there are no guarantees of convergence.

Note that Eq. 11 is also utilized as the recognition decision function.

2.3 Similarity Learning in Training

Our structured prediction method can be applied in test time for ZSR as long as the similarity matrix $\mathbf{S}$ in Eq. 9 can be calculated. Therefore our method is very

flexible, and can be incorporated with other ZSL methods such as [13,28,29,34] for the purpose of recognition. Inspired by the success of semantic embedding, we learn the following similarity function κ with embedding functions ϕ_s, ϕ_t:

$$\kappa(\phi_s(\mathbf{x}_c^{(s)}), \phi_t(\mathbf{x}_i^{(t)})) \stackrel{\text{def}}{=} \phi_s(\mathbf{x}_c^{(s)})^T \phi_t(\mathbf{x}_i^{(t)}) = \phi_s(\mathbf{x}_c^{(s)})^T \mathbf{W} \mathbf{x}_i^{(t)}, \tag{12}$$

where $\phi_t(\mathbf{x}_i^{(t)}) \stackrel{\text{def}}{=} \mathbf{W}\mathbf{x}_i^{(t)}$ is a linear embedding function. Specifically we propose independent learning of embedding functions for source and target domains, respectively, as follows:

(i) Source domain semantic embedding based on mixture models: We simplify the embedding function in [34] and propose using the following optimization problem to define embedding function ϕ_s:

$$\phi_s(\mathbf{x}_y^{(s)}) = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{x}_y^{(s)} - \mathbf{X}_s \boldsymbol{\alpha}\|_2^2, \text{ s.t. } \boldsymbol{\alpha} \geq 0, \mathbf{e}^T \boldsymbol{\alpha} = 1, \tag{13}$$

where $\mathbf{x}_y^{(s)}$ denotes an arbitrary seen or unseen class attribute vector, $\mathbf{X}_s = [\mathbf{x}_c^{(s)}]_{c=1,\cdots,C} \in \mathbb{R}^{d_s \times C}$ denotes the matrix consisting of *all* the seen class attribute vectors as its columns, and $\mathbf{e}$ denotes a vector consisting of all 1's. Clearly the source domain mapping function ϕ_s projects an arbitrary attribute vector onto a $(C-1)$-simplex and represents it as a mixture of seen class attribute vectors. As a result all the C seen class attribute vectors are mapped to the C unique vertices of the simplex accordingly. In test time we use QP to solve Eq. 13 so that all the unseen class attribute vectors can be mapped to unique points on the simplex due to the convexity of Eq. 13.

(ii) Target domain semantic embedding based on multi-class classification: With function ϕ_s in Eq. 13, the learning of linear embedding approaches such as [28,29] can be simplified to the training problem of multi-class SVMs, because in each source domain seen class semantic embedding, there exists only one bin that is not 0 and equal to 1. Consequently we utilize the following optimization to learn the target domain semantic embedding function ϕ_t:

$$\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{W}\|_F^2 + \rho \sum_{i=1}^{N} \sum_{c=1}^{C} \max\left\{0, \mathbf{1}_{\{c=y_i\}} \mathbf{W}_c \mathbf{x}_i^{(t)}\right\}, \tag{14}$$

where $\mathbf{W} \in \mathbb{R}^{C \times d_t}$ denotes the multi-class classifier, $\forall c, \mathbf{W}_c \in \mathbb{R}^{1 \times d_t}$ denotes the c-th row in $\mathbf{W}$ for predicting the similarities between data instances and seen class c, and $\rho \geq$ is a predefined regularization parameter. We utilize existing linear SVM solver such as LIBLINEAR [41] to solve Eq. 14.

Note that this learning approach above is essentially a (source domain) *denoising* version of [29]. For simplicity in our experiments latter we denote this learning approach as ***BL-ZSL***, namely the baseline approach for ZSL.

2.4 Cross-Validation on Predefined Parameters

As in [13,34], we utilize cross-validation to determine suitable values for training-time SVM regularization parameter ρ in Eq. 14 and test-time structured prediction regularization parameters λ_s, λ_t in Eq. 9. Precisely we randomly select two

held-out seen classes for validation purpose to tune λ_s, λ_t, and use the remaining data to tune ρ for training SVMs. We repeat this procedure for several times and choose the parameter combination which returns the best average ZSR performance. The easiest way to set these predefined parameters for our Algorithm 1 is $\lambda_s \gg \lambda_t \gg 1$. Then Eq. 10 is simplified as

$$\forall c', \max_{\{Z_{c',j'}\}} \sum_{j' \in \mathcal{J}_{c'}} S_{c',j'} Z_{c',j'}, \text{ s.t. } \forall j' \in \mathcal{J}_{c'}, Z_{c',i'} \geq 0, \sum_{j'} Z_{c',j'} = 1, \qquad (15)$$

which can be solved efficiently using LP. In practice we find that this simplified version of Algorithm 1 achieves similar performance to the complete one but offers significant computational improvement.

Table 1. Zero-shot recognition accuracy comparison (%) using CNN features in the form of "mean±standard deviation". Here numbers for the comparative methods are cited from the original papers, and "-" means no repeated result available yet.

Method	aP&Y	AwA	CUB	SUN	Ave.
Akata *et al.* [29]	-	61.9	40.3	-	-
Lampert *et al.* [4]	38.16	57.23	-	72.00	-
Fu *et al.* [15]	-	***80.5***	***47.9***	-	-
Kodirov *et al.* [14]	-	75.6	40.2	-	-
Romera-Paredes & Torr [27]	24.22 ± 2.89	75.32 ± 2.28	-	82.10 ± 0.32	-
Zhang & Saligrama [34]	46.23 ± 0.53	76.33 ± 0.83	30.41 ± 0.20	82.50 ± 1.32	58.87
Zhang & Saligrama [13]	***50.35 ± 2.97***	79.12 ± 0.53	41.78 ± 0.52	***83.83 ± 0.29***	***63.77***
BL-ZSL (*i.e.* Denoising version of [29])	39.45	70.45	39.58	84.00	58.37
[13] + Label Propagation [16]	58.7	82.6	50.2	84.0	68.9
[27] + SP-ZSR	37.5	84.3	-	***89.5***	-
[13] + SP-ZSR	62.19 ± 4.65	***92.08 ± 0.14***	***55.34 ± 0.77***	86.12 ± 0.99	73.93
BL-ZSL + SP-ZSR	***69.74 ± 3.47***	92.06 ± 0.18	53.26 ± 1.04	86.01 ± 1.32	***75.27***

3 Experiments

We test our method with predefined attributes for ZSR on aP&Y, AwA, CUB, and SUN. In our experiments we utilize the same experimental settings, including the CNN features and data preprocessing, as [13,34]. We denote by ***SP-ZSR*** our batch-mode ZSR method, and report our results averaged over 100 trials. To overcome the randomness in Algorithm 1, in each trial we run Algorithm 1 for another 100 times and record the average as probabilities over unseen classes per target data. We predict class labels and report our performance based on this assignment probability matrix in each trial.

The computational complexity of our method SP-ZSR scales as O(#target-data * #unseen-classes). Our implementation is based on unoptimized MATLAB code[2] with multi-thread computation, and potentially any ZSL method. In terms of running time, for instance, on aP&Y with [13,34] we can finish prediction within 5 min for 1 trial with 100 runs of Algorithm 1 on a common PC.

[2] Our demo code is available at https://zimingzhang.wordpress.com/publications/.

3.1 Zero-Shot Recognition

For this task we are only interested in whether or not the predicted class label for a target data instance is correct. Therefore, we measure the overall recognition performance by accuracy, while for each individual class we measure the performance by precision and recall (equivalence to accuracy per class).

We summarize the benchmark comparison results against recently proposed methods in Table 1. Overall our method outperforms the state-of-the-art by large margins. Our SP-ZSR significantly improves upon the accuracy of state-of-art ZSL methods using traditional online mode such as [13] by more than 10 %. Compared against related methods such as [14,15] which both benefit from exploring data structures like ours, our method significantly outperforms these methods by 11.58 % on AwA and 7.44 % on CUB, respectively. Also our SP-ZSR outperforms label propagation methods such as [16] by 5.03 %. These observations indicate that our method is more effective in accounting for test-time data shifts

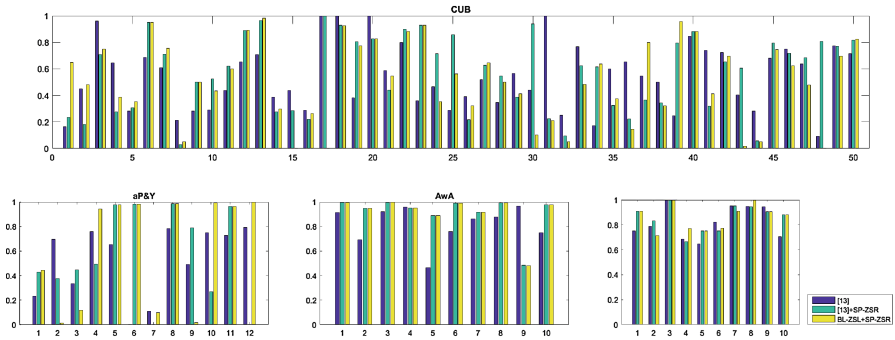

Fig. 2. Class-level recognition precision comparison, where y-axis denotes precision and x-axis denotes the indexes of unseen classes in the corresponding datasets.

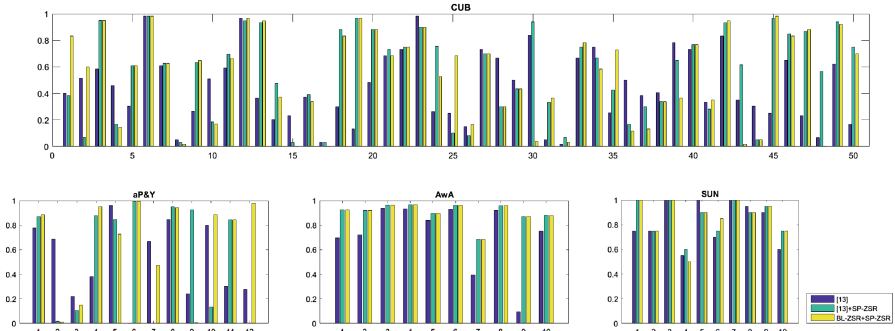

Fig. 3. Class-level recognition recall comparison, where y-axis denotes recall and x-axis denotes the indexes of unseen classes in the corresponding datasets.

Table 2. Average precision and recall comparison (%) for recognition.

Precision	aP&Y	AwA	CUB	SUN	Ave.
Zhang & Saligrama [13]	52.70 ± 27.33	81.70 ± 14.67	54.06 ± 24.13	82.51 ± 12.24	67.74
[13] + SP-ZSR	55.96 ± 35.72	$\boldsymbol{91.37 \pm 14.75}$	$\boldsymbol{57.09 \pm 27.91}$	85.96 ± 10.15	72.59
BL-ZSL + SP-ZSR	$\boldsymbol{62.80 \pm 42.67}$	91.37 ± 14.83	51.10 ± 29.66	$\boldsymbol{86.12 \pm 9.78}$	**72.84**
Recall (*i.e.* , class accuracy)					
Zhang & Saligrama [13]	51.34 ± 29.69	72.14 ± 26.29	45.05 ± 26.16	82.00 ± 16.31	62.63
[13] + SP-ZSR	54.66 ± 42.27	$\boldsymbol{90.28 \pm 8.08}$	$\boldsymbol{55.73 \pm 31.80}$	$\boldsymbol{86.00 \pm 13.19}$	71.67
BL-ZSL + SP-ZSR	$\boldsymbol{65.36 \pm 37.29}$	90.25 ± 8.09	53.30 ± 33.39	86.00 ± 14.97	**73.73**

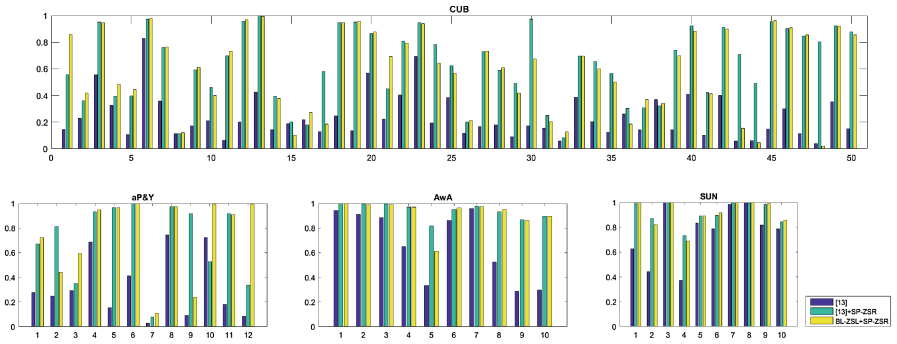

Fig. 4. Class-level average precision (AP) comparison for retrieval, where y-axis denotes AP and x-axis denotes the indexes of unseen classes in the corresponding datasets.

as opposed to methods [14,15] which directly seek to associate test data distribution with training data.

To better analyze the performance of our SP-ZSR for recognition, we also tabulate the class level precision and recall comparison in Figs. 2 and 3, respectively. Here (and in the following experiments) we only consider [13] as the baseline comparative approach because it achieves the state-of-the-art over the four datasets on average. In general SP-ZSR helps improve the performance on individual class when its distribution can be separated from others. In some cases, SP-ZSR decreases precision (or recall), but increases recall (or precision). In few cases, however, we observe that recognition with estimation of data distributions deteriorate the performance on both measures, such as class 2 in aP&Y and class 8 in CUB. More details can be seen from the class distributions in Fig. 5.

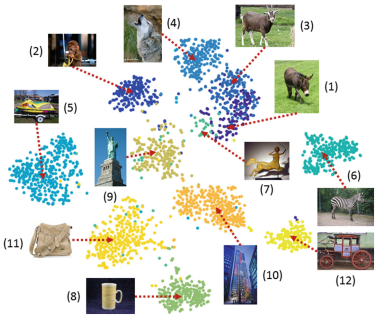

Fig. 5. Visualization of class distributions for aP&Y using CNN features.

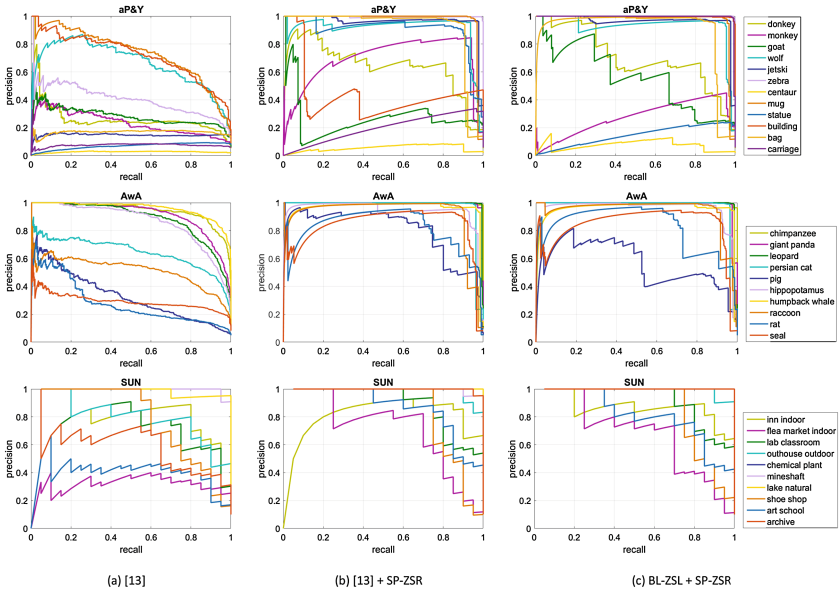

Fig. 6. Precision-recall curve comparison for retrieval on aP&Y, AwA, and SUN. The class names in each legend correspond to the indexes along x-axis in Figs. 2, 3, and 4, respectively.

To summarize the precision and recall comparison, we list the average numbers over all unseen classes in each dataset in Table 2. Overall SP-ZSR does help improve both precision and recall by, at least, 5.10 % and 9.04 %, respectively. Though the learning methods [13] and BL-ZSL are different, our SP-ZSR leads to similar performance. The large standard deviation implies that the performance for individual class has large variability. We will explore this issue further in our future work.

3.2 Zero-Shot Retrieval

In zero-shot retrieval we rank the assignment probabilities per unseen class and measure the retrieval performance by average precision (AP) and precision-recall curve per class. In this way we hope to explore performance of ZSR methods from the perspective of retrieval.

Table 3. mAP comparison (%) for zero-shot retrieval.

Method	aP&Y	AwA	CUB	SUN	Ave.
Zhang & Saligrama [13]	32.69	66.56	23.93	76.48	49.92
[13] + SP-ZSR	70.70	***94.03***	***63.25***	***92.17***	***80.04***
BL-ZSL + SP-ZSR	***74.11***	92.05	58.76	91.68	79.15

As an overview, we first summarize the mean average precision (mAP) comparison in Table 3. SP-ZSR appears to improve upon the retrieval performance of [13] significantly by 30.12 %. Again with different learning approaches, SP-ZSR

works equally well. These results suggest that for retrieval exploring test-time data structures is much more useful than for recognition.

Similar to recognition, we also show the class-level AP performance in Fig. 4. Overall SP-ZSR helps improve the retrieval performance on individual class by taking data structure into account. Unlike recognition, there are a few cases (*i.e.* class 10 in aP&Y, and classes 16, 38 in CUB) where our method leads to small degradation over using only cross-domain similarities. Possible reasons for deterioration could be the tuning parameters or the learned similarity matrix. Note that if the initial predicted similarities for certain class are not distinguishing and its corresponding distribution is not separable as well, just like class 7, "centaur", in aP&Y, our SP-ZSR cannot be expected to work well.

Next we analyze our retrieval performance from the perspective of precision-recall curve as shown in Fig. 6. We do not display the figures for CUB dataset to avoid unnecessary clutter in our illustrations. Note that larger areas under the precision-recall curves again demonstrate the superior performance with our structured prediction method.

4 Conclusion

The focus of this paper is on improving the recognition and retrieval performance of learned classifiers for unseen classes under the supposition that target domain data forms clusters in a suitable embedded space. To deal with the problems such as domain shift in ZSL, we propose a novel structured prediction approach to seek a globally well-matched assignment structure between clusters and unseen classes in test time. Our idea is motivated by the fact that there is a substantial performance gap between supervised learning and current state-of-art ZSL. The key difference between the two approaches is that the former approach benefits from utilizing test data distribution during training. With this as justification we propose classifying unseen target data by taking into consideration not only the learned similarities but also empirical distribution of unlabelled target data. In particular we introduce an unsupervised clustering subroutine into the assignment procedure so that target data structures in both clustering and assignment can be updated iteratively. Empirically we demonstrate significant improvement consistently over state-of-the-art in both zero-shot recognition and retrieval on the four popular benchmark datasets for ZSR.

Acknowledgement. We thank the anonymous reviewers for their very useful comments. This material is based upon work supported in part by the U.S. Department of Homeland Security, Science and Technology Directorate, Office of University Programs, under Grant Award 2013-ST-061-ED0001, by ONR Grant N00014-13-C-0288 and US AF contract FA8650-14-C-1728. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the social policies, either expressed or implied, of the U.S. DHS, ONR or AF.

References

1. Antol, S., Zitnick, C.L., Parikh, D.: Zero-shot learning via visual abstraction. In: ECCV, pp. 401–416 (2014)
2. Bhatia, K., Jain, H., Kar, P., Varma, M., Jain, P.: Sparse local embeddings for extreme multi-label classification. In: NIPS (2015)
3. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR, pp. 1778–1785 (2009)
4. Lampert, C.H., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. PAMI **36**(3), 453–465 (2014)
5. Mensink, T., Verbeek, J., Perronnin, F., Csurka, G.: Metric learning for large scale image classification: generalizing to new classes at near-zero cost. In: ECCV, pp. 488–501 (2012)
6. Parikh, D., Grauman, K.: Interactively building a discriminative vocabulary of nameable attributes. In: CVPR, pp. 1681–1688 (2011)
7. Rohrbach, M., Stark, M., Schiele, B.: Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In: CVPR, pp. 1641–1648 (2011)
8. Berg, T.L., Berg, A.C., Shih, J.: Automatic attribute discovery and characterization from noisy web data. In: ECCV, pp. 663–676 (2010)
9. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M.A., Mikolov, T.: Devise: a deep visual-semantic embedding model. In: NIPS, pp. 2121–2129 (2013)
10. Socher, R., Ganjoo, M., Manning, C.D., Ng, A.: Zero-shot learning through cross-modal transfer. In: NIPS, pp. 935–943 (2013)
11. Yu, F.X., Cao, L., Feris, R.S., Smith, J.R., Chang, S.F.: Designing category-level attributes for discriminative visual recognition. In: CVPR, pp. 771–778 (2013)
12. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. JMLR **9**(2579–2605), 85 (2008)
13. Zhang, Z., Saligrama, V.: Zero-shot learning via joint latent similarity embedding. In: CVPR (2016)
14. Kodirov, E., Xiang, T., Fu, Z., Gong, S.: Unsupervised domain adaptation for zero-shot learning. In: ICCV (2015)
15. Fu, Y., Hospedales, T.M., Xiang, T., Gong, S.: Transductive multi-view zero-shot learning. PAMI **37**(11), 2332–2345 (2015)
16. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. IEEE Trans. Knowl. Data Eng. **20**(1), 55–67 (2008)
17. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: NIPS, pp. 561–568 (2002)
18. Krizhevsky, A.: Learning multiple layers of features from tiny images. Master's thesis (2009)
19. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 dataset. Technical report (2011)
20. Patterson, G., Xu, C., Su, H., Hays, J.: The sun attribute database: beyond categories for deeper scene understanding. IJCV **108**(1–2), 59–81 (2014)
21. Palatucci, M., Pomerleau, D., Hinton, G.E., Mitchell, T.M.: Zero-shot learning with semantic output codes. In: NIPS, pp. 1410–1418 (2009)
22. Mahajan, D., Sellamanickam, S., Nair, V.: A joint learning framework for attribute models and object descriptions. In: ICCV, pp. 1227–1234 (2011)
23. Wang, X., Ji, Q.: A unified probabilistic approach modeling relationships between attributes and objects. In: ICCV, pp. 2120–2127 (2013)

24. Yu, X., Aloimonos, Y.: Attribute-based transfer learning for object categorization with zero/one training example. In: ECCV, pp. 127–140 (2010)
25. Mensink, T., Gavves, E., Snoek, C.G.M.: Costa: co-occurrence statistics for zero-shot classification. In: CVPR, pp. 2441–2448, June 2014
26. Hariharan, B., Vishwanathan, S., Varma, M.: Efficient max-margin multi-label classification with applications to zero-shot learning. Mach. Learn. **88**(1–2), 127–155 (2012)
27. Romera-Paredes, B., Torr, P.H.S.: An embarrassingly simple approach to zero-shot learning. In: ICML (2015)
28. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for attribute-based classification. In: CVPR, pp. 819–826 (2013)
29. Akata, Z., Reed, S., Walter, D., Lee, H., Schiele, B.: Evaluation of output embeddings for fine-grained image classification. In: CVPR, June 2015
30. Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G.S., Dean, J.: Zero-shot learning by convex combination of semantic embeddings. In: ICLR (2014)
31. Li, X., Guo, Y.: Max-margin zero-shot learning for multi-class classification. In: AISTATS (2015)
32. Li, X., Guo, Y., Schuurmans, D.: Semi-supervised zero-shot classification with label representation learning. In: ICCV (2015)
33. Ba, J.L., Swersky, K., Fidler, S., Salakhutdinov, R.: Predicting deep zero-shot convolutional neural networks using textual descriptions. arXiv preprint arXiv:1506.00511 (2015)
34. Zhang, Z., Saligrama, V.: Zero-shot learning via semantic similarity embedding. In: ICCV (2015)
35. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: ICML, pp. 97–105 (2015)
36. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: ICML, pp. 689–696 (2011)
37. Wang, W., Arora, R., Livescu, K., Bilmes, J.: On deep multi-view representation learning. In: ICML, pp. 1083–1092 (2015)
38. Yang, Y., Hospedales, T.M.: A unified perspective on multi-domain and multi-task learning. arXiv preprint arXiv:1412.7489 (2014)
39. Khamis, S., Lampert, C.H.: Coconut: co-classification with output space regularization. In: BMVC (2014)
40. Jaakkola, T.S.: Tutorial on variational approximation methods. In: Opper, M., Saad, D. (eds.) Advanced Mean Field Methods: Theory and Practice, p. 129. MIT Press, Cambridge (2001). Kindly check and confirm the edit made in Ref. [40]
41. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)