Unsupervised Domain Adaptation for Zero-Shot Learning

Elyor Kodirov, Tao Xiang, Zhenyong Fu, Shaogang Gong
Queen Mary University of London, London E1 4NS, UK
{e.kodirov, t.xiang, z.fu, s.gong}@qmul.ac.uk

Abstract

Zero-shot learning (ZSL) can be considered as a special case of transfer learning where the source and target domains have different tasks/label spaces and the target domain is unlabelled, providing little guidance for the knowledge transfer. A ZSL method typically assumes that the two domains share a common semantic representation space, where a visual feature vector extracted from an image/video can be projected/embedded using a projection function. Existing approaches learn the projection function from the source domain and apply it without adaptation to the target domain. They are thus based on naive knowledge transfer and the learned projections are prone to the domain shift problem. In this paper a novel ZSL method is proposed based on unsupervised domain adaptation. Specifically, we formulate a novel regularised sparse coding framework which uses the target domain class labels' projections in the semantic space to regularise the learned target domain projection thus effectively overcoming the projection domain shift problem. Extensive experiments on four object and action recognition benchmark datasets show that the proposed ZSL method significantly outperforms the state-of-the-arts.

1. Introduction

Conventional approaches to visual recognition are based on supervised learning. That is, given a large labelled training dataset of a known set of classes (*e.g*. hundreds of instances per class), a classifier is learned to classify each instance in a test dataset into the same set of classes. Collecting large quantities of annotated instances for each class is a bottleneck, especially when visual recognition research is moving towards a finer granularity [3]. For example, naming many fine-grained bird classes (*e.g*. Snowy Egret) is very challenging for most people except bird experts, let alone collecting instances. Inspired by humans' ability to recognise a new object category (class) without ever seeing a visual instance, zero-shot learning (ZSL) has received increasing interests [18, 9, 19, 23, 15, 1, 36, 11, 2]. Given a labelled training dataset containing seen classes, visual recognition by ZSL aims to recognise a visual instance of a new class that has never been seen before (hence zero-shot), which greatly reduces the annotation cost and makes visual recognition more scalable.

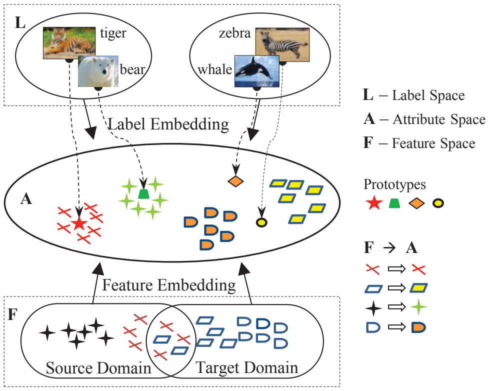


Figure 1: An illustration of the visual feature projection approach and how it suffers from the domain shift problem [28] without domain adaptation. For the two classes in the source domain and the two in the target, both their visual feature vectors and class names are embedded in a semantic space (attribute in this case) shared between the two domains. When the low-level feature projection function is learned from the source and applied without adaptation to the target, the target domain data and their class prototypes are well separated, resulting in poor classification. This is due to domain shift – although both tiger and zebra have the 'has stripe' attribute, their stripes are visually very different.

A machine learning approach to ZSL requires to extract the knowledge from the labelled training (source) dataset and to transfer that knowledge to the test (target) dataset. ZSL can thus be considered as a special case of transfer learning [28]. However it differs from conventional inductive transfer learning [28] such as multi-task learning in that the target domain is unlabelled, and transductive transfer learning such as domain adaptation [16] because the two domains have different tasks and non-overlapping label spaces. This unique setting renders most existing transfer learning methods inapplicable.

Since the target domain has no labelled data, existing ZSL methods adopt a naive transfer learning approach by which a model learned from the labelled source domain is applied to the target domain blindly without any model adaptation. More specifically, existing ZSL methods typically assume that there is a semantic embedding space within which both the feature space and the class label spaces of the source and target domains can be embedded (see Fig. 1). A commonly adopted semantic embedding space is an attribute space [6, 18, 1, 37, 15], *e.g*. the class label 'polar bear' can be represented as a binary attribute vector indicating that it 'is white', 'has fur' and 'eats fish'. Alternatively, such a space could be a semantic word space [24, 10, 11, 2] where the bigram 'polar bear' is represented as a high-dimensional word vector. The embedded label vector in any given semantic space is called a class prototype [10, 11]. Given a semantic embedding space, most existing methods take a *visual feature projection* approach [18, 9, 19, 23, 15, 1, 36, 11, 2]. Specifically, the knowledge extracted from the source data is represented in the form of a projection function that maps each low-level feature vector computed from a source object image (labelled) to its class prototype as an attribute or word vector. After this projection, the learned knowledge (projection function) is applied to the target data to project each target image into the same embedding space. After such projections, the classification of these target images can be simply nearest neighbour distance matching to the target class prototypes in the semantic space [9, 19] or its probabilistic variants [18, 10]. Without adapting the learned projection function to the target domain, existing methods are prone to the projection domain shift problem [10], as illustrated in Fig. 1.

In this paper, we propose to solve the domain shift problem by developing a new unsupervised domain adaptation model. As mentioned earlier, the ZSL problem itself is not a domain adaptation problem because the two domains have different tasks/classes. However, taking a visual feature projection approach, the learning of projection function for the target domain is a standard domain adaptation problem – both domains are projected into the same label space (attribute or word vector) – albeit an unsupervised one as no label is available in the target domain. Uniquely, instead of learning a typical classification/regression function as in most previous works [18, 9, 10, 11, 2], we cast the projection function learning problem as a sparse coding problem: Each dimension of the semantic embedding space corresponds to a dictionary basis vector and the coefficients/sparse code of each visual feature vector is its projection in the semantic embedding space. To learn a meaningful dictionary/projection function for the target data, we introduce two important regularisation terms in the cost function making our framework a regularised sparse coding model designed specifically for unsupervised domain adaptation. The first term controls the adaptation strength from the source domain to the target domain, whilst the second term rectifies explicitly the domain shift problem in ZSL, requiring the embedded target data to be near to the unseen class prototypes.

Our contributions are twofold: (1) For the first time, ZSL is formulated as an unsupervised domain adaptation problem, tailor-made for solving the challenging ZSL problem. (2) A regularised sparse coding based unsupervised domain adaptation framework is proposed to solve the domain shift problem suffered by existing ZSL methods. Extensive experiments on four challenging object and action benchmark datasets [18, 34, 32] validate the advantages of the proposed model, and demonstrate that the proposed model outperforms the state-of-the-arts naive transfer learning based models [4, 18, 1, 10, 37, 15, 19, 20, 11, 2] when applied to ZSL, often by a big margin.

2. Related Work

Semantic Embedding Space. All ZSL methods exploit semantic embedding spaces as the bridge for knowledge transfer. The semantic embedding spaces considered by most early works are attribute spaces [18, 19, 23, 15, 1, 36]. However, to represent an object class in an attribute space, an attribute ontology has to be defined manually (*e.g*. what attributes are needed to describe different types of birds) and each class needs be annotated by an attribute vector (*e.g*. a bird expert needs to define various attributes of a Snowy Egret). Such requirements hinder the scalability of an attribute space based ZSL method. To overcome this, more recent works explore the semantic word vector space [9, 10, 26, 11, 2], which is learned using large corpus of unannotated text for natural language processing tasks such as sentence completion [24]. The text corpus is so big that any class label or textual description of the class [5] can be embedded in this space, effectively mitigating the scalability issue. Both types of spaces are exploited simultaneously in our framework.

Visual Feature Projection vs. Visual-Semantic Similarity Matching. As mentioned earlier, most existing ZSL methods are *visual feature projection* based. Alternatively, a *visual-semantic similarity matching* approach can be adopted [18, 7]. Taking this approach, the knowledge from the source data is learned and represented in the form of a n-way probabilistic seen class classifier. This knowledge is then applied to the target data by computing a visual similarity between a target image and each of the source classes in the visual feature space. Finally, a target data point is classified to a target class if the visual similarity relationships between the data point and all the source classes match with the corresponding semantic similarity relationships (profile) between this target class and all the source classes in the semantic space. Our regularised sparse coding based domain adaptation framework combines the visual feature projection and visual-semantic similarity matching based approaches in a single formulation. Specifically, visual-semantic similarity matching is integrated seamlessly as part a regularisation term in our sparse coding based projection function learning model. We demonstrate empirically through experiments that fusing the two types of approaches benefits the ZSL task.

Projection Domain Shift. The problem of domain shift under the ZSL setting was first reported in [10] and known as the projection domain shift problem. The solution offered in [10] is a heuristic one-step self-training strategy to pull the prototype towards its closet data points (not necessary from the same class) followed by a multi-view embedding based on canonical correlation analysis (CCA) to align different semantic spaces with the low-level feature space. Our method differs from [10] in that (1) We consider this as unsupervised domain adaptation problem and develop a single-step learning process to adapt the projection function from the source to the target domain. In contrast, in [10] it is learned by two separate steps: first learned from the source data then adapted to the target data using CCA embedding. Any missing information from the first step cannot be recovered in the second step. (2) Our model rectifies the domain shift problem by visual-semantic similarity matching using regularisation in sparse coding learning, rather than a heuristic preprocessing step based on one-step self-training. Our experiments show that this more principled approach is superior to the heuristic approach of [10].

Unsupervised Domain Adaptation. Despite the fact that ZSL as a whole is not a domain adaptation problem, a key step of its solution – learning a target domain projection function is, because both the source and target data have the same task of projection to the same semantic space. A large variety of unsupervised domain adaptation approaches have been proposed [22] ranging from covariate shift, self-labelling, feature representation adaptation, to clustering based approaches. Most of them are designed for text document analysis. However, recently a number of methods are proposed for visual recognition [30]. Among them the most relevant are the unsupervised subspace alignment based approaches [13, 12, 25, 8], as sparse coding can also be considered as learning a projection to a subspace (*i.e.* the semantic space in ZSL). However, there are key differences: First, we do not aim to align data distributions of the source and target domains. This is because although they can be described by the same set of attributes, it is a multi-label problem (*e.g.* each image is described by multiple attributes), which is a setting not considered in [13, 12, 25, 8]. Second, since the target class labels can be embedded in the same space, they are exploited to regularise the learned target domain projection to explicitly tackle the domain shift problem. In contrast, the subspace learned in [13, 12, 25, 8] does not have semantic meaning thus cannot exploit the semantic relatedness between target and source data classes. Note that our unsupervised domain adaptation method is also transductive. This is because of the unique nature of zero-shot learning: there is no separate training data in the target domain.

3. Methodology

3.1. Problem Formulation

Suppose there are c_s source classes with n_s labelled instances $S = \{X_s, Y_s, \mathbf{z}_s\}$ and c_t target classes with n_t unlabelled instances $T = \{X_t, Y_t, \mathbf{z}_t\}$. Each instance is represented using a $d-$dimensional visual feature vector. We thus have $X_s \in \mathbb{R}^{n_s \times d}$ and $X_t \in \mathbb{R}^{n_t \times d}$, and $X_s = [\mathbf{x}_1, \ldots, \mathbf{x}_{n_s}]$ and $X_t = [\mathbf{x}_1, \ldots, \mathbf{x}_{n_t}]$ where $\mathbf{x}_i \in \mathbb{R}^d$. $\mathbf{z}_s \in \mathbb{R}^{n_s}$ and $\mathbf{z}_t \in \mathbb{R}^{n_t}$ are class label vectors for the source and target data respectively. We assume that the source and target classes are disjoint: $\mathbf{z}_s \cap \mathbf{z}_t = \varnothing$. Given a semantic embedding space, Y_s and Y_t are the $m-$dimensional semantic representation of each class label z in the source and target datasets respectively (*e.g.* m-dimensional binary attribute vectors). Therefore, $Y_s \in \mathbb{R}^{n_s \times m}$ and $Y_t \in \mathbb{R}^{n_t \times m}$; $Y_s = [\mathbf{y}_1, \ldots, \mathbf{y}_{n_s}]$ and $Y_t = [\mathbf{y}_1, \ldots, \mathbf{y}_{n_t}]$, where $\mathbf{y}_i \in \mathbb{R}^m$. For the source dataset, Y_s is given because each visual instance $\mathbf{x}_i$ of the source data is labelled by either a binary attribute vector or a continuous word vector representing its corresponding class label z_s^i. In contrast, Y_t has to be estimated because the target dataset is unlabelled. The problem of zero shot learning (ZSL) is thus to estimate Y_t and $\mathbf{z}_t$ given X_t.

3.2. Sparse Coding for Projection Learning

We aim to learn a projection function to map each $d-$dimensional visual feature vector $\mathbf{x}_i$ in X_s or X_t to a $m-$dimensional semantic embedding vector $\mathbf{y}_i$. We typically have $m < d$, *i.e.* we seek a lower dimensional subspace to project $\mathbf{x}_i$ into. In the context of ZSL, the subspace is a semantic space (attribute or word). In this work, the learning of the visual space to semantic space projection is formulated as a dictionary learning and sparse coding problem. Sparse coding aims to represent a data vector as a sparse linear combination of basis elements, which are atoms of a learned dictionary. Taking attribute space as an example, to project a data point from the visual feature space (higher dimensional) to an attribute space (lower dimensional), we consider that each basis element (atom) corresponds to an attribute (or an axis in the attribute space). For example, to represent the attribute of 'has fur' in an image of an animal, a corresponding sparse coding coefficient of the image is the weight of that basis element in the image which represents how much fur is present in that image[1]. Denote the dictionary as $D \in \mathbb{R}^{d \times m}$, a visual feature vector $\mathbf{x}_i$ can be reconstructed as $D\,\mathbf{y}_i$ using its coefficient vector/projection $\mathbf{y}_i$ and the dictionary/projection matrix D. Dictionary learning is thus to learn D and $\mathbf{y}_i$ to

[1] This is related to the concept of relative attributes [29].

minimise the reconstruction error. In our definition, each dictionary basis has clear semantic meaning therefore we call the learned dictionary *semantic dictionary*.

Next we shall formulate the dictionary learning problem separately for the source and target domains respectively, and highlight the difference in their formulations. First, in the source domain the sparse coding coefficient vector for each visual instance is known: For each $\mathbf{x}_i$, its corresponding $\mathbf{y}_i$ is the embedding (attribute or word vector) of its class label z_s^i in the semantic space. This is very different from the conventional dictionary learning by sparse coding whereby $\mathbf{y}_i$ needs to be estimated together with D. Let us denote the source domain semantic dictionary as D_s, the dictionary learning problem can be solved by quadratic optimisation:

$$D_s = \min_{D_s} \|X_s - D_s Y_s\|_F^2, \;\; s.t. \;\; ||d_i||_2^2 \leq 1, \qquad (1)$$

where $\|.\|_F$ is the Frobenius norm of a matrix. It is a standard least squares minimisation problem with a closed form solution. To avoid trivial solutions, a regularisation term is added to favour a solution of smaller norm. Eq. (1) thus becomes:

$$D_s = \min_{D_s} ||X_s - D_s Y_s\|_F^2 + \lambda ||D_s||_F^2 \;\; s.t. \;\; ||d_i||_2^2 \leq 1, \; (2)$$

where λ controls the strength of the regularisation term. This is known as a ridge regression problem, also with a closed form solution [14].

Second, contrary to the source domain, the formulation for dictionary/project function learning by sparse coding in the target domain requires the conventional sparse coding mechanism as both the dictionary and the coefficient vectors are unknown and need to be learned from data:

$$\{D_t, Y_t\} = \min_{D_t, Y_t} \|X_t - D_t Y_t\|_F^2 + \lambda ||Y_t||_1 \;\; s.t. \;\; ||d_i||_2^2 \leq 1, \qquad (3)$$

where $||Y_t||_1 = \sum_{i=1}^{n_t} ||\mathbf{y}_i||_1$. In this formulation, the model is essentially learning a sparse representation of the data *unsupervised*. Since both D_t and Y_t are unknown and unconstrained (apart from enforcing Y_t to be sparse), there is no guarantee that the learned representation captures a suitable semantic embedding space so that D_t is the correct projection function for X_t and the projection Y_t in the semantic embedding space is meaningful for ZSL. Indeed, we found through experiments that without any regularisation, the unsupervised learned D_t has no use for ZSL.

Such a regularisation could come from the labelled source domain. Following the conventional naive transfer/non-adaptation ZSL approach, D_s is learned from the source domain (Eq. (1)) and then applied directly to the target data. This method, which forces $D_t = D_s$ rather than allowing D_t to be adapted from D_s, is prone to the domain shift problem illustrated in Fig. 1. To overcome this domain shift problem, we propose to use both D_s and the target domain class prototypes to regularise the learning of D_t. This results in a novel unsupervised domain adaptation method for ZSL based on regularised sparse coding.

3.3. Unsupervised Domain Adaptation by Regularised Sparse Coding

Now, we introduce two critical regularisation terms into Eq. (3) to impose (a) an *adaptation regularisation constraint*: The D_t learned from the unlabelled target data should be similar to D_s learned from the labelled source data; and (b) an *visual-semantic similarity constraint*: The "closeness" of the interpretations of target data ($\mathbf{y}_i$) to their true class labels in the semantic embedding space (*i.e.* unseen class prototypes denoted as $\mathbf{p}_i^t$ and $i \in \{1, \ldots, c_t\}$). This defines the new regularised sparse coding framework:

$$\{D_t, Y_t\} = \min_{D_t, Y_t} \|X_t - D_t Y_t\|_F^2 + \lambda_1 \|D_t - D_s\|_F^2 +$$
$$\lambda_2 \sum_{i,j} w_{ij} \|\mathbf{y}_i - \mathbf{p}_j^t\|_2^2 + \lambda_3 ||Y_t||_1 \;\; s.t. \;\; ||d_i||_2^2 \leq 1. \qquad (4)$$

Adaptation regularisation constraint (AR). Compared to Eq. (3), the new regularisation term $\|D_t - D_s\|_F^2$ in Eq. (4) regularises the amount of adaptation (closeness) of the learned dictionary D_t to the supervised learned dictionary D_s. This term makes sure that the learned D_t is also a semantic dictionary that projects a target data point from the feature space to the same semantic space as D_s. In this regard, D_s is treated as a basis for learning the dictionary D_t so that D_t is not deviated freely from D_s. Without this regularisation, D_s could be adapted towards a trivial solution especially when $n_t > n_s$, *i.e.* the target data outnumbers the source data.

Visual-semantic similarity constraint (VSS). The second new regularisation term $\sum_{i,j} w_{ij} \|\mathbf{y}_i - \mathbf{p}_j^t\|_2^2$ in Eq. (4) enforces the visual-semantic similarity constraint. This is used to ensure that the learned coefficient vector $\mathbf{y}_i$ for each target data (its projection in the semantic embedding space) is close to its true class label z_i^t, embedded in the semantic embedding space as $\mathbf{p}_i^t$. Since z_i^t is unknown, we obtain an estimate by visual-semantic similarity matching using the indirect attribute prediction (IAP) method [18], where a probability is computed for $\mathbf{x}_i$ being labelled as z_j^t which defines the closeness of $\mathbf{y}_i$ to $\mathbf{p}_j^t$. Formally, the probability of $\mathbf{x}_i$ being the j-th unseen class is used as weight w_{ij} to enforce a closeness in the distance between the projection $\mathbf{y}_i$ and the j-th unseen class prototype $\mathbf{p}_j^t$, resulting in this regularisation term defined as $\sum_{i,j} w_{ij} \|\mathbf{y}_i - \mathbf{p}_j^t\|_2^2$. Note that this constraint utilises visual-semantic similarity matching whilst the sparse-coding dictionary aims to estimate the optimal visual feature projection. Our model therefore exploits simultaneously both ZSL strategies (see Sec. 2) in our unified regularised sparse coding framework.

Algorithm 1: Unsupervised domain adaptation for ZSL

```
Input: $\{X_s, Y_s\}$, $X_t$, $D_s$, $\{\mathbf{p}_1^t, ..., \mathbf{p}_{c_t}^t\}$, $\lambda_1$, $\lambda_2$ and $\lambda_3$.
Output: $Y_t$ the coefficients for unseen class data, and $D_t$
        the dictionary.
Initialise: $Y_t$ and $w_{ij}$ by Eq. (3) and IAP respectively
while not converge do
    Update $D_t$ by Eq. (6);
    Update $Y_t$ by Eq. (7);
end
```

3.4. Optimisation

It is important to point out that Eq. (4) is not convex for D_t and Y_t simultaneously, although it is convex for each of them separately. We thus deploy an alternating optimisation method to solve it. In particular, we alternate between the following two subproblems:

(1) Fix Y_t, update D_t

$$D_t^* = \underset{D_t}{\operatorname{argmin}} \|X_t - D_t Y_t\|_F^2 + \lambda_1 \|D_t - D_s\|_F^2 \tag{5}$$

This is a standard least squares problem and we have the closed form solution:

$$D_t^* = (X_t Y_t^T + \lambda_1 D_s)(Y_t Y_t^T + \lambda_1 I)^{-1}. \tag{6}$$

(2) Fix D_t, update Y_t

$$Y_t^* = \underset{Y_t}{\operatorname{argmin}} \|X_t - D_t Y_t\|_F^2 + \lambda_2 \sum_{i,j} w_{ij} \|\mathbf{y}_i - \mathbf{p}_j^t\|_2^2 + \lambda_3 ||Y_t||_1 \tag{7}$$

In this equation, the first two terms can be combined into a single quadratic form and it becomes a conventional sparse coding problem. To solve it we use the Lasso [33] solver from the SPAMS toolbox [21]. The iterations will terminate when the objective function in Eq. (4) converges or after a fixed number of iteration. Note that we put a positive constraint on coefficients if the attribute space is used because it does not make sense to have a negative attribute value. For semantic word space, this constraint is removed. The complete algorithm is summarised in Algorithm 1. Our algorithm always converges within 5 iterations in our experiments.

3.5. Zero-Shot Classification

Single Semantic Space. Once the dictionary coefficients Y_t is estimated, zero-shot classification can be performed. In this work, two classification strategies are considered: 1) a nearest neighbour (NN) classifier and 2) a semi-supervised label propagation (LP) framework. For the NN classifier, given a target data $\mathbf{x}_i$, its coefficients y_t^i is directly used to compare with the unseen prototypes. It is then labelled as the nearest unseen class. For the LP classifier, the method in [10] is adopted. Specifically, we exploit the unseen data and the unseen prototypes (as the labelled data) to set up a graph, then the label information is propagated from the unseen prototypes to each unseen data. We report the performance of our algorithm on both strategies.

Combining Multiple Spaces. Multiple semantic spaces can be easily combined in our framework to exploit their complementarity. For example, after estimating Y_t^A and Y_t^W for attribute and word space, respectively, we can combine the similarity matrices from these two spaces by a simple strategy: For the NN classifier, the distances to neighbours are averaged; for the LP classifier, the graph similarity matrix are averaged before label propagation.

4. Experiments

4.1. Datasets and Settings

Datasets. Four datasets are used in our experiments. (a) **AwA dataset** [18] consists of 30,475 animal images belonging to 50 classes. An 85D attribute vector is provided for each class. (b) **Caltech-UCSD Birds 2011 (CUB)** [34] is a fine-grained dataset with attributes. It contains 200 different bird classes, with 11,788 images in total. The class level attribute annotations are given with 312 visual attributes (*e.g*. color, part pattern). (c) **aPascal-aYahoo** [6] consists of two attribute datasets: aPascal is a 12.695 images subset of the PASCAL VOC 2008 dataset and aYahoo has 2,644 images. A 64D attribute vector is provided for each image. There are 20 object classes for aPascal, and 12 for aYahoo and they are disjoint. aPascal-aYahoo is used for cross-dataset ZSL (see Sec. 4.4). (d) **UCF101 dataset** [32] is one of the largest datasets for action recognition with 101 classes, containing 13,320 video clips and 27 hours of video data in total. The videos are collected from YouTube with large camera motions and cluttered background making them particularly challenging. Each class is annotated with 115 hand crafted attributes (*e.g*. body posture, body part motion). To our best knowledge, there has been no zero-shot action classification experiment reported on this dataset.

Features. We use two types of features: Deep Convoluational Neural Network (CNN) features and traditional low-level features. For the AwA dataset, we extract the OverFeat implementation of the CNN features (4,096D) [31]. For the low-level features, the dataset-provided low-level features are used, same as in previous works [18, 1, 10]. For the CUB dataset, we extract the same CNN features as for AwA. Since low-level features are not provided with the dataset, we follow [1] to extract 96 color descriptors from regular grids at three scales, and aggregate them into fisher vectors (FVs) using 256 Gaussians. For aPascal-aYahoo, we use the 9,751D low-level features provided by [6]. For the UCF101 dataset, we use the features provided by the THUMOS challenge [17] which contain 4,000D Motion Boundary Histogram (MBH) features [35].

Settings. We use two types of semantic embedding spaces: (a) An attribute space where each class is represented as a binary attribute vector. (b) An 100D semantic word space learned by the skip-gram model [24] using a text corpus containing 4.6M Wikipedia pages. For the source/target class split, we use the standard 40/10 split for the AwA dataset. For the CUB dataset, we use the same 150/50 split as in [1]. For aPascal-aYahoo, we use standard split, that is, aPascal is used for training, and aYahoo for testing. For the UCF101 dataset, two types of split are used: 81/20 and 51/50. In each experiment, we run 10 trails with different random splits, and report the average classification accuracy with standard errors. The four parameters λ, λ_1, λ_2, and λ_3 (Eq. (2) and Eq. (4)) are set empirically and we found that the results are insensitive to the parameter values. Other parameters are set by 5-fold cross validation using the source data. These include the the LP parameters: the number of neighbours to construct similarity and the parameter for balancing the propagation rate [10]. For the zero-shot classifier (see Sec. 3.5), LP is used for the reported results unless stated otherwise.

4.2. Comparison with the State-of-the-art

Comparative models. For AwA, we select 11 most recent and competitive ZSL methods for comparison, as shown in Table 1. These 11 models differ in various aspects: (1) Features: Most reported results on the dataset-provided low-level features, although more recently the CNN features have been used [27, 4, 11, 2]. (2) Side information (SI): This refers to what semantic information extracted from human knowledge is used. In addition to embedding each class label into either an attribute space (A) or word vector space (W), the Wordnet hierarchy (H) is used in [1] and [2]. Some methods [37, 15] also use a different form of human annotation: instead of class attribute annotation, a class similarity (CS) matrix is annotated. (3) Most of them are based on the visual feature projection approach, with the only exception of IAP [18] which uses visual-semantic similarity matching.

In contrast, far fewer studies have been reported on the more challenging CUB dataset (more and finer-grain classes). For the UCF101 action recognition dataset, no results have been reported so far, although Liu *et al.* [19] has tackled a similar zero-shot learning problem for action recognition, albeit using different (much smaller) datasets. The method in [19] is essentially based on learning projection to the attribute space using the source data and then using the projection function to project the target data in the attribute space followed by nearest neighbour based classification. In addition to [19], we also use code in [18] to obtain the results on DAP and IAP. For all the competitors, kernelised SVMs are used to learn the attribute classifiers (projection function to the attribute space) and source class classifiers (for IAP). In contrast, in our model the learned dictionary acts as attributes classifiers.

Performance Comparison.

Method	F	SI	AwA	CUB
IAP [18]	L/C	A	42.2/44.5	5.60/19.5
DAP [18]	L/C	A	41.4/53.2	10.5/31.4
DS [20]	L/C	W/A	35.7/52.7	-
AHLE [1]	L	A+H	43.5	18.0
Deng *et al.* [4]	L/C	A	38.5/44.2	-
TMV-BLP [10]	L	A+W	47.1	-
Yu *et al.* [37]	L	CS	48.3	-
Jayaraman [15]	L	A+CS	48.7	-
Makoto [27]	C	A	62.4	-
Fu *et al.* [11]	C	A+W	66.0	-
Akata *et al.* [2]	L/L+C	A+W+H	42.3/67.8	19.0/**47.1**
Ours	L/C	A	47.5/73.2	26.7/ 39.5
Ours	L/C	A+W	**49.7/75.6**	**28.1**/ 40.6

Table 1: ZSL results on AwA and CUB in classification accuracy on the target data (%). Notations – 'F': features; 'L': low-level features; 'SI': side information; 'C': CNN features; 'A': attribute space; 'W': semantic word vector space; 'H': WordNet hierarchy; 'CS': class similarity. When two results are reported, they correspond to the two types of features used.

AwA and CUB benchmarking – Table 1 shows that overall our method has the best performance on these two image datasets. In particular, it is observed that: (1) On AwA, if the same low-level features are used, our result (49.7%) is the highest. (2) The results reported in [10] give the most competitive alternative to ours in this setting. As discussed in Sec. 2, TMV-BLP [10] and the proposed model are the only two which aim to rectify the projection domain shift problem by utilising the unlabelled target domain data. These results suggest that the proposed new regularised sparse coding based formulation is more effective than the two-step (projection followed by adaptation) approach taken by [10]. (3) When the more powerful CNN features are used, our model gains a significant performance boost, rising from 49.7% to 75.6% and the gap to the best alternative (67.8% [2]) becomes bigger. (4) The same conclusion can be drawn on the CUB dataset – our overall results are superior to the compared methods. Note that [2] obtained better result using the CNN features (47.1% vs. 40.6%) but its result on low-level feature is much weaker than ours (19.0% vs. 28.1%). It is worth pointing out that [2] employ combined low-level and CNN features, and use more than two semantic spaces. In contrast, other methods including ours use only one type of features and no more than two semantic spaces. Richer features and more complementary semantic spaces would certainly help our method as well but were not used to be fair to other compared methods.

UCF101 benchmarking – For this dataset, the results are

Method	SI	51/50 (%)	81/20 (%)
DAP [18]	A	02.2 ± 0.5	06.1 ± 1.5
IAP [18]	A	06.9 ± 1.1	11.1 ± 1.9
Liu *et al*. [19]	A	02.5 ± 1.2	06.2 ± 2.1
Ours	A	13.2 ± 1.9	20.1 ± 3.1
Ours	A+W	**14.0** ± **1.8**	**22.5** ± **3.5**

Table 2: Results on the UCF101 dataset

Method	AwA (%)	CUB (%)	UCF101(%)
GFK [12]	65.2	31.7	16.3
SIDL [25]	64.3	33.2	18.7
SADA [8]	65.7	31.4	17.4
Ours-no-adapt	62.1	34.5	18.1
Ours	**75.6**	**40.6**	**22.5**

Table 3: Evaluations on unsupervised domain adaptation methods. CNN features are used.

shown in Table 2. Comparing Table 2 with Table 1, it is apparent that ZSL for action recognition from videos is a much harder task than object recognition from images. In particular, with 50 target classes in both CUB and UCF101, the same DAP and IAP methods yielded much poorer results, close to the chance level (2%) in the case of DAP. In addition, the following observations can be made: (1) Our model performs much better than the three compared alternatives [18, 19], almost doubling the recognition rates of the best competitor (IAP) under both settings. Note that although we use an additional semantic embedding space (word space), our results with attributes alone is still much better. (2) The very poor results from both DAP and [19] suggest that projection without adaptation fails completely on this dataset. Moreover, it also suggests that using the source data to learn a n-way classifier for measuring the visual similarity is more sensible for video actions given the larger domain shift problem at hand. This explains the better performance of IAP than DAP and [19].

4.3. Comparison with Unsupervised Domain Adaptation Methods

In this experiment, we demonstrate unsupervised domain adaptation helps ZSL, and our regularised sparse coding based adaptation is better than the alternatives.

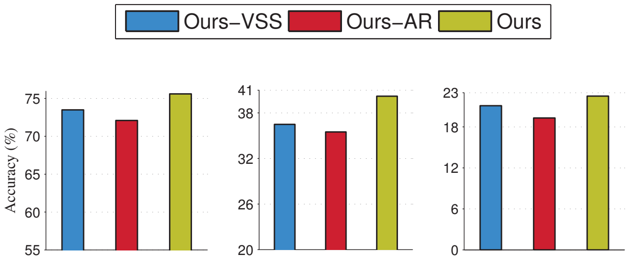
Competitors. For all three datasets, we compare our method with three most recent and relevant subspace alignment based unsupervised domain adaptation methods: 1) Geodesic Flow Kernel (GFK) [12] 2) Subspace Alignment Domain Adaptation (SADA) [8] 3) Subspace Interpolation Dictionary Learning (SIDL) [25]. All three methods attempt to align the data distributions of the two domains. When applied to our ZSL problem, the projection function (based on the same sparse coding model) learned in the source domain can thus be used directly for the target domain after they are aligned. In addition we also compare with our model without adaptation, that is, setting $D_t = D_s$ (see Sec. 3.2), denoted as ours-no-adapt.

Performance Comparison. Table 3 shows the comparative results. We can see that adaptation certainly helps in our framework: comparing ours with ours-no-adapt, a clear improvement can be observed thanks to the adaptation of D_s to D_t using Eq. (4). The results also show that the alternative subspace alignment based adaptation methods are much weaker than ours. The results are slightly better than that without adaptation on AwA; but on the more challenging CUB dataset, their adaptations have an adverse effect. These results thus suggest that existing unsupervised domain adaptation methods are not effective under the ZSL setting. This is because that they are designed for visual recognition problems where each data can only have a single class label. In a multi-label scenario such as ZSL (*e.g*., each AwA image can have dozens of attributes present), the subspace alignment alignment strategy would not be a good strategy. This is particularly true for CUB where all images contain a bird and aligning the distributions of two sets of bird images will have little effect because the two distributions may have already been similar. The alignment thus would not help to solve the more subtle domain shift problem that the beak of a seagull is different from that of a pigeon. In contrast, our model utilises the unseen class prototypes to regularise the learning of target domain projection function which is specifically designed for rectifying the domain shift problem for zero-shot learning.

4.4. Further Analysis

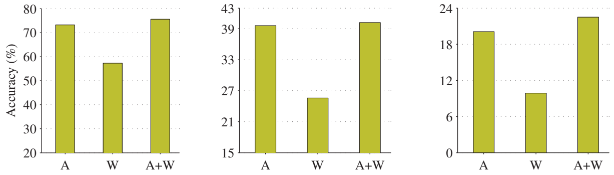
Effects of Regularisations Terms. In Fig. 2, we compare our full model with various stripped-down versions of the model to validate the contributions of the two regularisation terms in Eq. (4). Specifically we compare our full model (Eq. (4)) with our model without the visual-semantic similarity constraint, *i.e*. Eq. (4) without the regularisation term $\sum_{i,j} w_{ij} \|\mathbf{y}_i - \mathbf{p}_j^t\|_2^2$ (denoted ours–VSS, '–' for minus) and our model without the adaptation regularisation constraint (ours–AR). The results in Fig. 2 show clearly that both regularisation terms contribute to the superior performance of our model.

Effects of Combining Multiple Semantic Spaces. In our framework, the attribute and semantic word vector space are combined in our label propagation based zero-shot classification algorithm (see Sec. 3.5). Figure 3 shows the results of our model when only one of the two semantic embedding spaces is used. It can be seen that the model performance is notably improved by utilising both semantic spaces. It is also noted that using just one semantic space, the model already achieves very competitive performance. Moreover, the performance in the attribute space is stronger



(a) AwA (b) CUB (c) UCF101

Figure 2: Evaluation of the contributions of each component of our framework on AwA, CUB and UCF101 (CNN features).



(a) AwA (b) CUB (c) UCF101

Figure 3: Effectiveness of combining multiple semantic embedding spaces (CNN features).

compared to the word vector space because the latter is unsupervised and does not benefit from human annotation. In particular it is noted that the semantic word space is much weaker for UCF101. This is because the action names such as 'apply lipstick' and 'ski-jet' are much more abstract and ambiguous to describe the rich content of the associated actions, compared to the nouns in the image datasets (*e.g.* 'giant panda'). Simply embedding the class names to the semantic word space may not be the best way to explore the word space for ZSL in action recognition especially for subtle and complex actions.

Effects of the classification methods. The results reported so far are obtained using the label propagation (LP) classifier after domain adaptation. Table 4 shows that when the nearest neighbour classifier (NN) with cosine distance is used the performance is only slightly worse, by about 2%.

	AwA(%)	CUB(%)	UCF101(%)
NN	74.1	38.4	20.1
LP	**75.6**	**40.2**	**22.5**

Table 4: Classification methods: nearest neighbour (NN) vs. label propagation (LP).

The effects of the amount of target data used. One of the key differences between our model and the alternatives except [10] is that we exploit unlabelled target data in model learning. This is determined by the nature of our approach – a transfer learning method with any form of adaptation to the target data needs to use the target data. In this experiment, we evaluate how the learned projection is affected by the amount of target data used in model learning. Figure 4 suggests that the impact is very small. The performance on all three datasets only drops slightly with as few as 1% of the target data.

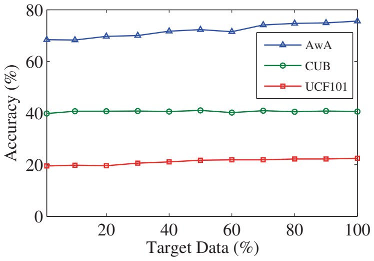


Figure 4: The effect of the amount of target data used

Cross-dataset ZSL. In this experiment, we follow the same setting as in [18] and evaluate our model on using aPascal [6] as source data and aYahoo [6] as target data. Since these datasets have per-image attribute annotations we learn dictionary with per-image labels. A ZSL classification accuracy of 26.5% is obtained by our model with NN as classifier, while with exactly the same features, the DAP and IAP results in [18] are 16.8% and 16.9% respectively - about 10% lower than ours.

Attribute prediction w/ and w/o domain adaptation. The attribute prediction accuracy on the target data of our model with and without domain adaptation is evaluated using the AUC metrics as in [18] . Without domain adaptation, the results on AwA, CUB, aPascal-aYahoo are 65.5%, 54.1%, and 56.7%, respectively, whereas the results with domain adaptation are 69.1%, 57.8%, and 59.2%, respectively. This suggests that domain adaptation leads to better attribute prediction accuracy, which in turn contributes to the better ZSL performance.

5. Conclusions

We have proposed a novel ZSL framework based on regularised sparse coding. Compared with most existing ZSL methods that perform naive transfer, our model is essentially an unsupervised domain adaptation model which learns a projection function from a visual space to a semantic embedding space using both labelled source and unlabelled target data. Extensive comparative evaluations validate the advantages of our model over the state-of-the-arts.

Acknowledgments

This work was funded by the EU FP7 Project SUNNY (grant no. 313243).

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *CVPR*, 2013. 1, 2, 5, 6

[2] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015. 1, 2, 6

[3] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2014. 1

[4] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *Computer Vision–ECCV 2014*, pages 48–64. Springer, 2014. 2, 6

[5] M. Elhoseiny, B. Saleh, and A.Elgammal. Write a classifier: Zero shot learning using purely textual descriptions. In *ICCV*, 2013. 2

[6] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 2, 5, 8

[7] J. Feng, S. Jegelka, S. Yan, and T. Darrell. Learning scalable discriminative dictionary with sample relatedness. In *CVPR*, 2014. 2

[8] B. Fernando, A. H. M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013. 3, 7

[9] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 1, 2

[10] Y. Fu, T. Hospedales, T. Xiang, Z. Fu, and S. Gong. Transductive multi-view embedding for zero-shot recognition and annotation. In *ECCV*, 2014. 2, 3, 5, 6, 8

[11] Z. Fu, T. Xiang, E. Kodirov, and S. Gong. Zero-shot object recognition by semantic manifold distance. In *CVPR*, 2015. 1, 2, 6

[12] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012. 3, 7

[13] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, pages 999–1006, 2011. 3

[14] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. 4

[15] D. Jayaraman and K. Grauman. Zero-shot recognition with unreliable attributes. In *NIPS*, 2014. 1, 2, 6

[16] J. Jiang. A literature survey on domain adaptation of statistical classifiers. 1

[17] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/THUMOS14/`, 2014. 5

[18] C. Lampert, H. Nickisch, and S. Harmeling. Attribute based classification for zero-shot visual object categorization. In *IEEE TPAMI*, 2013. 1, 2, 4, 5, 6, 7, 8

[19] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *CVPR*, 2011. 1, 2, 6, 7

[20] G. S. I. G. M. Rohrbach, M. Stark and B. Schiele. What helps where and why? semantic relatedness for knowledge transfer. In *CVPR*. IEEE, 2010. 2, 6

[21] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. In *Journal of Machine Learning Research*, volume 11, pages 19–60. 2010. 5

[22] A. Margolis. A literature review of domain adaptation with unlabeled data. 2011. 3

[23] T. Mensink, E. Gavves, and C. G. M. Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *CVPR*, 2014. 1, 2

[24] T. Mikolov, K. Chen, and G. Corrado. Efficient estimation of word representation in vector space. In *Proceedings of Workshop at ICLR*, 2013. 2, 6

[25] J. Ni, Q. Qiu, and R. Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *CVPR*, 2013. 3, 7

[26] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Devise: A deep visual-semantic embedding model. In *ICLR*, 2014. 2

[27] M. Ozeki and T. Okatani. Understanding convolutional neural networks in terms of category-level attributes. In *ACCV*, 2014. 6

[28] S. J. Pan and Q. Yang. A survey on transfer learning. *TKDE*, 2010. 1

[29] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011. 3

[30] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE Signal Process. Mag.*, 32(3):53–69, 2015. 3

[31] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *http://arxiv.org/abs/1312.6229*, 2013. 5

[32] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human action classes from videos in the wild. In *CRCV-TR-12-01*, 2012. 2, 5

[33] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 1996. 5

[34] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *ICCV*, 2011. 2, 5

[35] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 5

[36] X. Wang and Q. Ji. A unified probabilistic approach modelling relationships between attributes and objects. In *ICCV*, 2013. 1, 2

[37] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S. Chang. Designing category-level attributes for discriminative visual recognition. In *CVPR*, 2013. 2, 6