Chapter 2
An Embarrassingly Simple Approach
to Zero-Shot Learning

Bernardino Romera-Paredes and Philip H. S. Torr

Abstract Zero-shot learning concerns learning how to recognise new classes from just a description of them. Many sophisticated approaches have been proposed to address the challenges this problem comprises. Here we describe a zero-shot learning approach that can be implemented in just one line of code, yet it is able to outperform state-of-the-art approaches on standard datasets. The approach is based on a more general framework which models the relationships between features, attributes, and classes as a network with two linear layers, where the weights of the top layer are not learned but are given by the environment. We further provide a learning bound on the generalisation error of this kind of approaches, by casting them as domain adaptation methods. In experiments carried out on three standard real datasets, we found that our approach is able to perform significantly better than the state of the art on all of them.

2.1 Introduction

Zero-shot learning (ZSL) is a relatively recent machine learning paradigm that was introduced in the works [21, 28], and quoting the latter, it aims to tackle the following question:

> Given a semantic encoding of a large set of concept classes, can we build a classifier to recognise classes that were omitted from the training set?

That is, ZSL consists in recognising new categories of instances without training examples, by providing a high-level description of the new categories that relate them to categories previously learned by the machine. This can be done by means of

B. Romera-Paredes (✉) · P.H.S. Torr
Department of Engineering Science, University of Oxford, Parks Road,
Oxford OX1 3PJ, UK
e-mail: bernard@robots.ox.ac.uk; bernardino.romeraparedes@eng.ox.ac.uk

P.H.S. Torr
e-mail: philip.torr@eng.ox.ac.uk

learning an intermediate encoding describing each class, referred to as attributes. In words of [1]:

> Attributes correspond to high-level properties of the objects which are shared across multiple classes, which can be detected by machines and which can be understood by humans.

One recurrent example that we mention in this chapter is the use of attributes such as *white*, *strong*, *furry*, and *quadrupedal*, to describe and learn classes of animals.

Zero-shot learning has attracted considerable attention due to both its wide applicability to many real- world situations and the singular challenges it presents. An example of ZSL happens when dealing with an ever growing set of classes, such as detecting new species of living beings, using attributes such as the ones mentioned in the previous example. Another scenario occurs when the granularity of the description of the categories to be distinguished makes it infeasible to obtain training instances for each of them, e.g. when a user wants to recognise a particular type of shoe (we refer to Chap. 9 for more on this topic). The main challenge ZSL poses is to design a model able to exploit the relations between features, attributes, and classes, so that the knowledge learned at the training stage can be transferred to the inference stage, in a similar way as human beings are able to understand a new concept, if it is described as a combination of previously known attributes or concepts [27]. Hereafter, we use the term signature to refer to this attribute description of a class.

Zero-shot learning is inherently a two-stage process: training and inference. In the training stage, knowledge about the attributes is captured, and in the inference stage this knowledge is used to categorise instances among a previously unseen set of classes. Many efforts have been made to improve the training stage [10, 15, 17], whereas the inference stage has received little attention [16]. For example many approaches blindly assume that all attributes convey the same amount of information, and can be predicted with the same accuracy, thus, they are evenly utilised in the inference rule. However these assumptions rarely hold true in real world cases.

We study a framework that is able to integrate both stages, overcoming the need to make strong and unrealistic assumptions, as the ones previously described. This framework, introduced in [1], is based on modelling the relationship between features, attributes, and classes as a (linear) model composed of two layers. The first layer contains the weights that describe the relationship between the features and the attributes, and is learned at the training stage. The second layer models the relationship between the attributes and the classes and is fixed using the prescribed attribute signatures of the classes. Given that the seen classes and the unseen classes are different, this second layer is interchangeable.

The main contributions of this work are:

- Given the framework in [1], we derive a principled choice of the regularizer, which has three nice properties:

  1. It performs comparably or better than the state of the art.
  2. It is efficient both at the training and at the inference stages.
  3. It is extremely easy to implement: one line of code for training and another one for inference (without calling any external functions).

- We provide a bound on the generalisation error of the approaches comprised in this framework. This is done by bridging the gap between zero-shot learning and domain adaptation, and making use of previous results in the latter [4, 5].

The remainder of the chapter is organised as follows. In Sect. 2.2 we briefly review methods proposed to deal with zero-shot learning. In Sect. 2.3 we describe the above ZSL framework, and present our method. In Sect. 2.4 we analyse its learning capabilities. In Sect. 2.5 we report the results of our experiments on one synthetic and three standard real datasets. Finally in Sect. 2.6 we discuss the main contributions of this work and propose several research lines that can be explored.

2.2 Related Work

Zero-shot learning relies on learning how to recognise several properties or attributes from objects, so that these learned attributes can be harnessed when used in the description of new, unseen classes. Indeed, it is attributes learning that drives the possibility of learning unseen classes based only on their description [27]. Within the context of machine learning, an antecedent of the notion of attribute learning can be found in [9] in the form of binary descriptors. The aim was using these binary descriptors as error-correcting codes, although these did not convey any semantic meaning. Recently, there has been an increasing interest in attributes learning, partially due to the availability of data containing tags or meta-information. This has proved to be particularly useful for images [10, 11, 21], as well as videos [13, 24].

Many papers focus on attributes learning, namely the training stage in zero-shot learning methods, putting special emphasis on the need to disentangle the correlations between attributes at the training stage, because these properties may not be present in the target data [17]. For example in [10] the authors focus on the feature extraction process with the aim of avoiding confusion in the learning process of attributes that often appear together in the training set instances.

With regard to the inference stage in which the predicted attributes are combined to infer a class, many approaches are variants of 1-nearest neighbour, or probabilistic frameworks. Approaches that resemble 1-nearest neighbour consist in looking in the attribute space for the closest unseen class signature to the predicted attribute signature of the input instance. It is used in [10], and in [28] the authors study risk bounds of this approach when using the Hamming distances between the predicted signature and the signatures of the unseen classes. Whereas 1-nearest neighbour is an intuitive way for inferring classes from the attributes, it presents several drawbacks. Namely, it treats equally all dimensions of the attribute space, which may be suboptimal, as some attributes are more important than others for discriminating between classes, and metrics such as Hamming distance ignore quantitative information in the prediction of the attributes.

In [21, 22] the authors propose a two-stage probabilistic framework in which the predictions obtained in the first stage can be combined to determine the most likely unseen class. Within this framework two approaches are proposed: directed attribute prediction (DAP), and indirect attribute prediction (IAP). In DAP a probabilistic classifier (e.g. logistic regression model) is learned at the training stage for each attribute. At the inference stage, the previous estimators are used to infer among the unseen classes provided their attributes signatures. In IAP one probabilistic classifier is learned for each seen class, whereas at the inference stage the predictions are combined accounting for the signatures of both seen and unseen classes. The DAP approach has been widely used by many other methods. In [35] the authors extend DAP by weighting the importance of each attribute, based on its frequency of appearance. These probabilistic approaches bring a principled way of combining the attribute predictions of a new instance in order to infer its class. However, in addition to being unable to estimate the reliability of the predicted attributes, they introduce a set of independence assumptions that hardly ever hold in real world, for example, when describing animals the attributes "terrestrial" and "farm" are dependent, but are treated as independent in these approaches.

Very recently, the authors of [16] proposed an approach that acknowledges uncertainty in the prediction of attributes, having mechanisms to deal with it. The approach is based on random forests that classify attribute signatures into the unseen classes, using a validation partition from the training set. The resultant model empirically proves to be superior to previous inference methods, such as DAP, and it obtains state-of-the-art results in the benchmark datasets. One of the limitations of this model is the need to have the attribute signatures of the unseen classes at the training stage. In other words, the model learned at the training stage is tailored to work with a predefined set of unseen classes.

The approach we describe in Sect. 2.3 bypasses the limitations of these methods by expressing a model based on an optimisation problem which relates features, attributes and classes. There are some works which follow a similar strategy. A relevant approach is the one described in [1], where the authors propose a model that implicitly learns the instances and the attributes embeddings onto a common space where the compatibility between any pair of them can be measured. The approach we describe here is based on the same principle, however we use a different loss function and regularizer which not only makes the whole process simpler and efficient, but also leads to much better results. Another related approach is proposed in [14], where the authors use the information regarding the correlations between attributes in both training and test instances. The main differences are that they focus on attribute prediction, and they employ a max-margin formulation that leads to a more complex approach. These approaches [1, 14], as well as the one we propose, can be seen as particular instances of the general framework described in [37], which unifies a wide range of multitask learning and multi-domain learning methods.

Other approaches consider the attributes as latent variables to be learned. For example in [36], an explicit feature map is designed to model the relationships

between features, attributes and classes. Other approaches, such as [24, 26], consider different schemes where attributes representations are to be learned.

The approach we describe is grounded on the machine learning areas of transfer learning and domain adaptation. The term transfer learning encompasses several machine learning problems, and has received several names, such as learning to learn [23] or inductive transfer [7, 31, 33]. Here, we refer to transfer learning in the lifelong learning sense, that is, the aim is to extract knowledge from a set of source tasks, so that it can be applied to learn future tasks more efficiently. Zero-shot learning problems share the necessity to extrapolate the knowledge gained previously to tackle a new learning scenario. The main difference is that in transfer learning the information about the new tasks is given as a set of labelled instances, whereas in zero-shot learning this information takes the form of descriptions of the unseen classes. An extensive review of transfer learning methods can be found in [29].

The aim of domain adaptation is to learn a function from data in one domain, so that it can be successfully applied to data from a different domain [4, 8, 19]. It resembles transfer learning but there are important differences to note. In transfer learning the marginal input distribution (domain) in both source and target tasks is supposed to be the same, whereas each task comprises a different objective predictive function. For example, given a set of journal documents sampled from a fixed marginal distribution, a source task may consist in classifying documents between different topics, and the target task could be about classifying each document in terms of its author. Domain adaptation makes the reverse assumption, that is, the objective predictive function is the same but the marginal distributions for source and target tasks are different. Following the previous example, now we have a common function to learn: classifying documents in terms of different topics. However the source and target tasks receive documents from two different journals, that is, from two different marginal distributions. The link between our approach and domain adaptation becomes clear in Sect. 2.4.1.

2.3 Embarrassingly Simple ZSL

In order to explain our approach, we start by describing a standard linear supervised learning method, and then extend that model to tackle the ZSL scenario. In the following, we adopt the convention of using lower-case letters to denote scalars, lower-cases bold letters to denote vectors, and higher-case bold letters to denote matrices.

Supervised linear model

Let us denote by $\mathbf{X} \in \mathbb{R}^{d \times m}$ the instances available at the training stage, where d is the dimensionality of the data, and m is the number of instances. Similarly we use

$\mathbf{Y} \in \{0, 1\}^{m \times z}$ to denote the ground truth labels of each training instance belonging to any of the z classes. In most cases, each row of $\mathbf{Y}$ contains only one positive entry indicating the class it belongs to. Nevertheless, the present framework allows an instance to belong to several classes simultaneously.

If we were interested in learning a linear predictor for z classes, we would optimise the following problem:

$$\underset{W \in \mathbb{R}^{d \times z}}{\text{minimise}} \ L\left(\mathbf{X}^\top \mathbf{W}, \mathbf{Y}\right) + \Omega\left(\mathbf{W}\right), \tag{2.1}$$

where $\mathbf{W}$ contains the parameters to be learned, L is a convex loss function, and Ω a convex regularizer. Problem (2.1) encompasses several approaches, depending on the choice of L and Ω. For example if L is the sum of hinge losses, and Ω is the Frobenius norm, this would lead to a standard support vector machine (SVM), but one can consider other loss functions such as logistic loss, and other regularizers, such as the trace norm, leading to multitask learning methods [2, 32].

ZSL model

Quoting [21], the formal definition of the ZSL problem can be described as follows:

Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m) \subset \mathcal{X} \times \mathcal{Y}$ be training samples where $\mathcal{X}$ is an arbitrary feature space and $\mathcal{Y}$ consists of z discrete classes. The task is to learn a classifier $f : \mathcal{X} \longrightarrow \mathcal{Y}'$ for a label set $\mathcal{Y}'$ of z' classes, that is disjoint from $\mathcal{Y}$.

In order to accomplish that, we are given the attributes of all classes as additional information. We assume that each class is described by a known signature composed of a attributes. We can represent the training signatures in a matrix $\mathbf{S} \in [0, 1]^{a \times z}$. This matrix may contain boolean entries, when the description of classes is defined as a list of attributes, or more generally, it may contain for each attribute any value in [0, 1] providing a soft link between attributes and classes. Together matrices $\mathbf{Y}$ and $\mathbf{S}$ provide enough information so that one can obtain the ground truth attributes for each instance.

In problem (2.1) the attributes are not used, and therefore, there is no way to perform knowledge transfer from this set of classes to new classes. One can introduce the given information about the attributes, $\mathbf{S}$, by introducing a mapping from the attributes to the feature space, $\mathbf{V}$, such that $\mathbf{W} = \mathbf{VS}$, where $\mathbf{V} \in \mathbb{R}^{d \times a}$. That leads to the following problem, similar to the one proposed in [1]:

$$\underset{\mathbf{V} \in \mathbb{R}^{d \times a}}{\text{minimise}} \ L\left(\mathbf{X}^\top \mathbf{VS}, \mathbf{Y}\right) + \Omega\left(\mathbf{VS}\right). \tag{2.2}$$

At the inference stage, given the features of an instance, $\mathbf{x} \in \mathbb{R}^d$, we wish to determine to which class it belongs to, among a new set of z' unseen classes, $\mathcal{Y}'$, disjoint from

the set of seen classes, $\mathcal{Y}$. To do so, we are provided with their attributes signatures, $\mathbf{S}' \in [0, 1]^{a \times z'}$. The prediction is then given by

$$\underset{i \in [1, \dots, z']}{\operatorname{argmax}} \ \mathbf{x}^{\top} \mathbf{V} \mathbf{s}'_i, \tag{2.3}$$

where $\mathbf{s}'_i \in [0, 1]^a$ denotes the i-th column of matrix $\mathbf{S}'$.

One interpretation of this model is provided in [1]. There, each class is represented in the attribute space by means of its signature. Thus, the learning weights, $\mathbf{V}$, map any input instance, $\mathbf{x}$, into this attribute space. Given that both classes and instances are mapped into a common space, one can estimate the *compatibility* between them. Thus, at the inference stage, the model predicts the class in $\mathcal{Y}'$ that is most compatible with the input instance, by making use of (2.3). Note that if all given signatures are normalised, $\left\| \mathbf{s}'_1 \right\|_2 = \left\| \mathbf{s}'_2 \right\|_2 = \dots \left\| \mathbf{s}'_{z'} \right\|_2$, then the notion of maximum compatibility among the signatures corresponds to finding the minimal Euclidean distance with respect to $\mathbf{V}^{\top} \mathbf{x}$ in the attribute space.

It is important to note the advantage of this model with respect to typical ZSL approaches reviewed in Sect. 2.2. Recall that these approaches were based on first estimating the attributes of a given instance, and then finding the class that best matches the predicted attributes, using some probabilistic or distance measure. In this way, all attributes are assumed to convey the same amount of information, an assumption that is likely detrimental, as often some attributes have more discriminative power than others. On the other hand, the approach in (2.2) is able to learn and exploit the relative importance of each of the attributes for discriminating between classes. For example, if the i-th attribute has less discriminative powers than the others, then the i-th column of the learned weights $\mathbf{V}$ should have a smaller norm than the others, so that it has a smaller contribution in the classification decision.

The method above makes the implicit assumption that for each attribute, its reliability to discriminate between seen classes is similar to its reliability to distinguish between unseen classes. In order to explain why this assumption is reasonable, let us recall the example of animals classification, and let us assume that we are given the attributes *it has teeth*, and *is white*. The former attribute may be more difficult to recognise than the latter, given that some instances of animals may not show the mouth, whereas the colour of an animal is easy to infer. Hence the importance of the attribute *it has teeth* for the final classification decision should be low, independently of the classes at hand, given that it is more difficult to learn a reliable predictor for that attribute. This assumption is relevant whenever the reliability on estimating the attributes remain constant, regardless of the classes considered. The key point of this framework is that it does not try to minimise explicitly the classification error of the attributes, which are an intermediate layer that we are not directly interested in. Instead, it minimises the multiclass error of the final classes, by both learning implicitly how to recognise attributes, and also pondering the importance of each of them in the decision of the class.

There are several points to note from problem (2.2). First, if the regularizer Ω is of the form $\Omega(\mathbf{B}) = \Psi\left(\mathbf{B}^\top\mathbf{B}\right)$ for an appropriate choice of the function Ψ, then by using the representer theorem [3], this leads to a kernel version of the problem, where only inner products between instances are used:

$$\underset{\mathbf{A}\in\mathbb{R}^{m\times a}}{\text{minimise}}\ L\left(\mathbf{KAS}, \mathbf{Y}\right) + \Psi\left(\mathbf{S}^\top\mathbf{A}^\top\mathbf{KAS}\right), \tag{2.4}$$

where $\mathbf{K} \in \mathbb{R}^{m\times m}$ is the Gram matrix, $k_{i,j} = \langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle$, being $\phi(\mathbf{x})$ the representation of $\mathbf{x}$ in a given feature space. Secondly, problem (2.2) and its equivalent problem (2.4) are convex, thus its globally optimal solution can be found.

A scheme of this framework is shown in Fig. 2.1. This framework is utilised in its linear form (Eq. 2.2) in [1], for a particular choice of the loss function (based on the hinge loss function), and the regularizer (based on the Frobenius norm of the learning weights). In the following, we describe and justify a different choice for those elements, which leads to a more efficient and effective training model.

2.3.1 Regularisation and Loss Function Choices

We now come to the first contribution of this chapter. The framework described above comprises several approaches, which vary depending on their regularizers and loss functions. Here we design a regularizer which accomplishes the following desiderata:

- Given any (training) attribute signature, $\mathbf{s}_i \in [0, 1]^a$ for some $i \in [1, \ldots, z]$, its mapping to the d-dimensional feature space is given by $\mathbf{V}\mathbf{s}_i \in \mathbb{R}^d$. This representation must be controlled so that ideally the mapping of all signatures on the feature space have a similar Euclidean norm. This allows fair comparisons between signatures, and prevents problems that stem from highly unbalanced training sets.
- Conversely, the mapping of each training instance $\mathbf{x}_i$, for $i \in [1, \ldots, m]$, into the a-dimensional attribute space is given by $\mathbf{V}^\top\mathbf{x}_i \in \mathbb{R}^a$. Similarly to the previous point, it would be interesting to bound the Euclidean norm of that term. The aim here is to map all instances to a common region in the attribute space. In this way, we can encourage the generalisation of the model to test instances, if their representation into the attribute space fall into the same region where the training instances lie.

A regularizer that accomplishes the previous points can be written as follows:

$$\Omega\left(\mathbf{V}; \mathbf{S}, \mathbf{X}\right) = \gamma\left\|\mathbf{V}\mathbf{S}\right\|_{\text{Fro}}^2 + \lambda\left\|\mathbf{X}^\top\mathbf{V}\right\|_{\text{Fro}}^2 + \beta\left\|\mathbf{V}\right\|_{\text{Fro}}^2, \tag{2.5}$$

where the scalars γ, λ, β are the hyper-parameters of this regularizer, and $\|\cdot\|_{\text{Fro}}$ denotes the Frobenius norm. The first two terms account for the above points, and we have added one further term consisting in a standard weight decay penalising the Frobenius norm of the matrix to be learned.
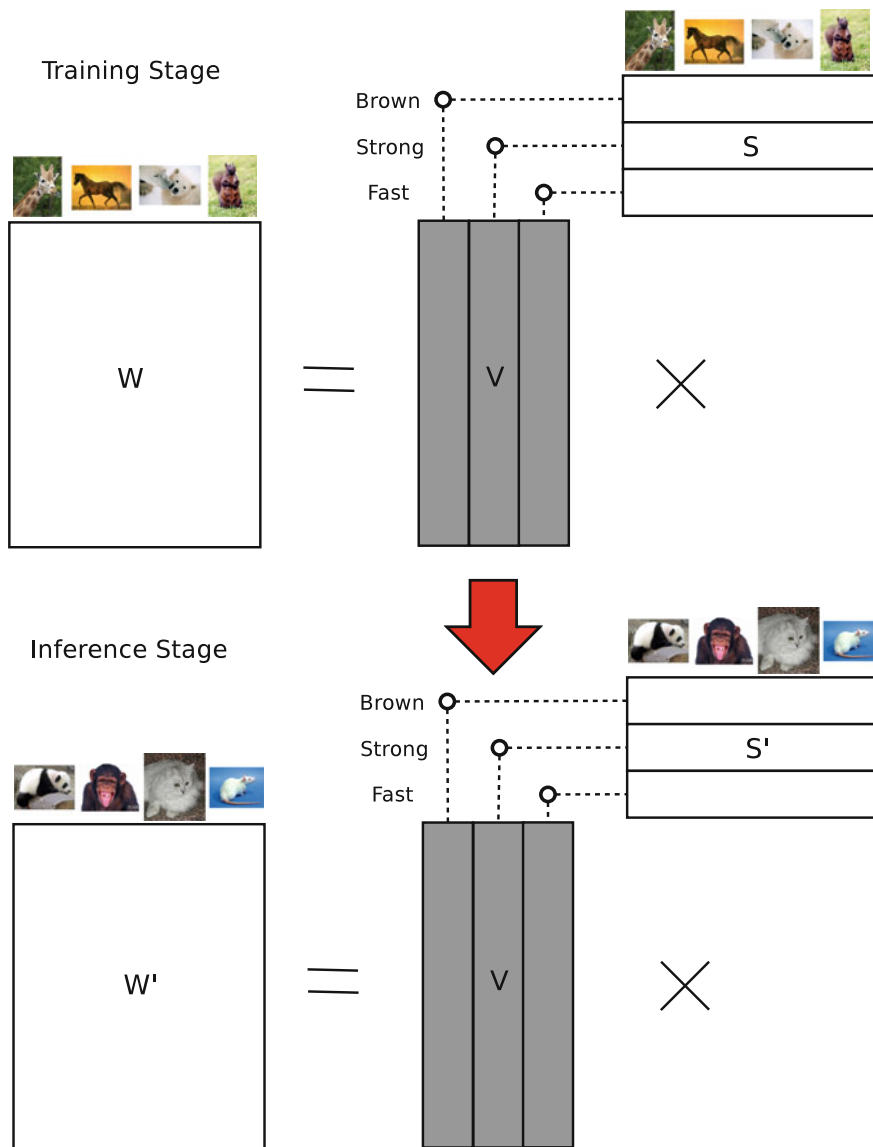
Fig. 2.1 Summary of the framework described in Sect. 2.3. At the training stage, we use the matrix of signatures **S** together with the training instances to learn the matrix **V** (in *grey*) which maps from the feature space to the attribute space. At the inference stage, we use that matrix **V**, together with the signatures of the unseen classes, **S′**, to obtain the final linear model **W′**

Having made these choices, we note that if:

- $L\left(\mathbf{P}, \mathbf{Y}\right) = \|\mathbf{P} - \mathbf{Y}\|_{\text{Fro}}^2$.
- $\beta = \gamma\lambda$

then the solution to problem (2.2) can be expressed in closed form:

$$\mathbf{V} = \left(\mathbf{X}\mathbf{X}^\top + \gamma\mathbf{I}\right)^{-1}\mathbf{X}\mathbf{Y}\mathbf{S}^\top\left(\mathbf{S}\mathbf{S}^\top + \lambda\mathbf{I}\right)^{-1}. \tag{2.6}$$

This, and the corresponding kernel version that can be derived from (2.4), are the one-line-of-code solutions we mentioned in the introduction.

2.4 Risk Bounds

In this section we provide some theoretical guarantees about our approach, bounding the expected error on the inference stage with respect to the training error. In order to do so, we first transform our problem into a domain adaptation one.

2.4.1 Simple ZSL as a Domain Adaptation Problem

Let us assume that problem (2.2) can be expressed in the following way:

$$\underset{\mathbf{V}\in\mathbb{R}^{d\times a}}{\text{minimise}} \sum_{i=1}^{m}\sum_{t=1}^{z}\ell\left(\mathbf{x}_i^\top\mathbf{V}\mathbf{s}_t^\top, y_{t,i}\right) + \Omega\left(\mathbf{V}\right), \tag{2.7}$$

where $\ell\left(\cdot, \cdot\right): \mathbb{R}\times\{-1, 1\} \longrightarrow [0, 1]$. That implies that one instance may be classified to belong to zero, one, or more than one classes. Such an assumption may be realistic in some cases, for example when there are some instances in the training set that do not belong to any seen class. Then, problem (2.7) can be expressed in a more conventional form:

$$\underset{\mathbf{v}\in\mathbb{R}^{da}}{\text{minimise}} \sum_{i=1}^{m}\sum_{t=1}^{T}\ell\left(\widetilde{\mathbf{x}}_{t,i}^\top\mathbf{v}, y_{t,i}\right) + \Omega\left(\mathbf{v}\right), \tag{2.8}$$

where

$$\widetilde{\mathbf{x}}_{t,i} = \text{vec}\left(\mathbf{x}_i\mathbf{s}_t^\top\right) \in \mathbb{R}^{da}. \tag{2.9}$$

Note that at the inference time, given a new instance, $\mathbf{x}$, the predicted confidence of it belonging to an unseen class t with attribute signature $\mathbf{s}_t$, is given by $\widetilde{\mathbf{x}}_t^\top\mathbf{v} = \mathbf{v}^\top\text{vec}\left(\mathbf{x}\mathbf{s}_t^\top\right)$. Therefore, even if the original test instances $\mathbf{x}$ were sampled from the same distribution as the training instances, the transformation of them

using attributes signatures makes the training and test instances come from different distributions. Note also that in the current settings, we are learning a unique common function across domains. As a consequence, we are facing a domain adaptation problem.

2.4.2   Risk Bounds for Domain Adaptation

Domain adaptation has been analysed from a theoretical viewpoint in several works [4, 5]. Here we apply these developments to our problem.

In a domain adaptation problem we assume that the training instances are sampled from a source distribution $\mathcal{D}$, and the test instances are sampled from a target distribution $\mathcal{D}'$. Following the definition of [4], a function h is said to be a predictor if it maps from the feature space to $\{0, 1\}$, and f is the ground truth labelling function for both domains, mapping from the feature space to $[0, 1]$. Then the expected error of h with respect to the source distribution is defined as:

$$\varepsilon(h) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \left[|f(\mathbf{x}) - h(\mathbf{x})| \right],$$

and the expected error of h with respect to the target distribution, $\varepsilon'(h)$, is defined accordingly.

Theorem 2 in [4] states that given a hypothesis space $\mathcal{H}$ of VC-dimension $\bar{d}$, and sets $\mathcal{U}, \mathcal{U}'$ of $\bar{m}$ instances sampled i.i.d. from $\mathcal{D}$ and $\mathcal{D}'$, respectively, then with probability at least $1 - \delta$, for every $h \in \mathcal{H}$:

$$\varepsilon'(h) \leq \varepsilon(h) + 4\sqrt{\frac{2\bar{d}}{\bar{m}} \left(\log \frac{2\bar{m}}{\bar{d}} + \log \frac{4}{\delta} \right)} + \alpha + \frac{1}{2} \hat{d}_{\mathcal{H} \Delta \mathcal{H}} \left(\mathcal{U}, \mathcal{U}' \right), \qquad (2.10)$$

where

- α is an upper-bound of $\inf_{h \in \mathcal{H}} \left[\varepsilon(h) + \varepsilon'(h) \right]$. In particular if the ground truth function f is contained in $\mathcal{H}$, then $\alpha = 0$.
- $d_{\mathcal{H}} \left(\mathcal{D}, \mathcal{D}' \right)$ is known as the $\mathcal{A}$-distance between distributions $\mathcal{D}$ and $\mathcal{D}'$ over the subsets defined in $\mathcal{H}$ [20]:

$$d_{\mathcal{H}} \left(\mathcal{D}, \mathcal{D}' \right) = 2 \sup_{h \in \mathcal{H}} |P_{\mathcal{D}}(h) - P_{\mathcal{D}'}(h)|,$$

where $P_{\mathcal{D}}(h)$ denotes the probability of any event in h, under the distribution $\mathcal{D}$. This is equivalent to the expected maximal accuracy achieved by a hypothesis in $\mathcal{H}$ separating the instances generated by the two different distributions $\mathcal{D}$ and $\mathcal{D}'$. In a similar vein, $\hat{d}_{\mathcal{H}} \left(\mathcal{U}_S, \mathcal{U}_T \right)$ is defined as the empirical distance between the samples $\mathcal{U}$ and $\mathcal{U}'$.

- $\mathcal{H} \Delta \mathcal{H}$ is the symmetric difference hypothesis space of $\mathcal{H}$ and it is defined as: $\mathcal{H} \Delta \mathcal{H} = \{h(x) \oplus h'(x) : h, h' \in \mathcal{H}\}$, $\oplus$ being the XOR operator. That is, a hypothesis g is in $\mathcal{H} \Delta \mathcal{H}$, if for a couple of hypothesis h, h' in $\mathcal{H}$, $g(x)$ is positive if and only if $h(x) \neq h'(x)$ for all x.

In our case $\mathcal{H}$ is the hypothesis space composed of all linear classifiers, $\bar{m} = mz$, and $\bar{d} = da + 1$. Let us assume that both train and test instances are sampled from the same distribution, $\mathcal{C}$. When we do the transformation specified in Eq. (2.9) using $\mathbf{S}$ and $\mathbf{S}'$ for the training and test instances, we end up having two different distributions, $\mathcal{D}$, and $\mathcal{D}'$ and we are interested in quantifying the $\mathcal{A}$-distance between them over our symmetric difference hypothesis space, $d_{\mathcal{H} \Delta \mathcal{H}} (\mathcal{D}, \mathcal{D}')$. The assumption about both train and test instances are sampled from the same distribution (before the transformation) may not hold true in many cases, however it can be a fair approximation in the standard case where the contribution of the differences of training and test distributions of the feature spaces is negligible in comparison to the differences between $\mathbf{S}$ and $\mathbf{S}'$ when quantifying the distance between distributions $\mathcal{D}$ and $\mathcal{D}'$.

We observe two extreme cases. The first one contemplates the trivial scenario where $\mathbf{S} = \mathbf{S}'$, so that both distributions are similar and thus the distance is 0. In that case, if $\alpha = 0$, the bound given in Eq. (2.10) becomes equivalent to the Vapnik–Chervonenkis bound on a standard classifier. The second case arises when each attribute signature of the seen classes is orthogonal to each attribute signature of the unseen classes, that is, for each $i \in \{1 \ldots z\}$, $j \in \{1 \ldots z'\}$, $\langle \mathbf{s}_i, \mathbf{s}'_j \rangle = 0$.

To make the explanation of the latter case clearer let us denote by $\mathbf{x} \in \mathbb{R}^d$ any training instance in the original feature space, and similarly let $\mathbf{x}' \in \mathbb{R}^d$ be any test instance. Then, by applying equation (2.9) using the training signature $\mathbf{s}_i$, and test signature $\mathbf{s}'_j$ we have

$$\widetilde{\mathbf{x}}_i = \mathrm{vec} \left(\mathbf{x} \mathbf{s}_i^\top \right) \in \mathbb{R}^{da}$$

$$\widetilde{\mathbf{x}}'_j = \mathrm{vec} \left(\mathbf{x}' \mathbf{s}'^\top_j \right) \in \mathbb{R}^{da}$$

Note that because of the orthogonality assumption between training and test signatures the following holds true:

$$\langle \widetilde{\mathbf{x}}_i, \widetilde{\mathbf{x}}'_j \rangle = \mathrm{trace} \left(\mathbf{x} \mathbf{s}_i^\top \mathbf{s}'_j \mathbf{x}'^\top \right) = 0. \tag{2.11}$$

Equation (2.11) implies that in the new feature space any training instance is orthogonal to any test instance. Because of that, the following lemma becomes useful.

Lemma 1 *Let us consider $\mathcal{H}$ be the hypothesis space composed of all linear classifiers. Then given two orthogonal sets $\mathcal{P}, \mathcal{Q}$, in which the element 0 is not in either of them, there exists a hypothesis $g \in \mathcal{H} \Delta \mathcal{H}$ which separates them.*

Proof Let us consider any couple of points $\mathbf{p} \in \mathcal{P}$, $\mathbf{q} \in \mathcal{Q}$ with the only condition that they are not zero. We define

$h(\mathbf{x}) = \mathrm{sign} \left((\mathbf{p} + \mathbf{q})^\top \mathbf{x} \right)$, and

$h'(\mathbf{x}) = \mathrm{sign} \left((\mathbf{p} - \mathbf{q})^\top \mathbf{x} \right)$.

For any point $\mathbf{p}' \in \mathcal{P}$, $h(\mathbf{p}') = h'(\mathbf{p}')$, given that by definition $\mathbf{p}'$ and $\mathbf{q}$ are orthogonal. Similarly, for any point $\mathbf{q}' \in \mathcal{Q}$, $h(\mathbf{q}') = -h'(\mathbf{q}')$.

Therefore, for any point in $\mathcal{Q}$, $g \in \mathcal{H} \Delta \mathcal{H}$ associated to functions $h, h' \in \mathcal{H}$ will be positive, and for any point in $\mathcal{P}$, the same function g will be negative. $\qquad \square$

As a consequence of Lemma 1, when the orthogonality assumption holds, the right-hand side term in Eq. (2.10) becomes bigger than 1, so that the bound is vacuous. One illustrative instance of this case happens when $\mathbf{S} = [\mathbf{B}, \mathbf{0}^{\mathbf{a},\mathbf{c}},]$, and $\mathbf{S}' = [\mathbf{0}^{\mathbf{a},\mathbf{b}}, \mathbf{C}]$ for some non-zero matrices $\mathbf{B} \in \mathbb{R}^{a \times b}$, $\mathbf{C} \in \mathbb{R}^{a \times c}$. In that case, the set of attributes that describe the seen classes are completely different from the ones describing the unseen classes, thus no transfer can be done.

All real scenarios lay between the previous cases. One interesting question is to characterise the value $d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathcal{D}, \mathcal{D}'\right)$ as a function of solely $\mathbf{S}$ and $\mathbf{S}'$. We leave this question open.

2.5 Experiments

In order to assess our approach and the validity of the statements we made, we conducted a set of experiments on one synthetic and three real datasets, which comprise a standard benchmark of evaluation of zero-shot learning methods.[1]

2.5.1 Synthetic Experiments

First we used synthetically generated data with the aim of both checking the correctness of the described method, which we refer to as ESZSL (embarrassingly simple zero-shot learning), and comparing it with the baseline algorithm DAP on a controlled set-up. All hyper-parameters required by these methods were tuned by a validation process. This process is based on leaving out one subset of validation classes, so that the performance of the model is validated against them. In all cases the range of values tried for the hyper-parameters was 10^b, for $b = -6, -5, \ldots, 5, 6$. This set of values was chosen after performing preliminary experiments which empirically showed that the optimal performance for both approaches is found within this interval.

The data were generated as follows. Initially, we created the signatures for the classes by sampling each element of $\mathbf{S}$ from a Bernoulli distribution with 0.5 mean. We created the ground truth mapping from the attributes to the features, $\mathbf{V}^+ \in \mathbb{R}^{a \times d}$, where we have fixed $a = 100$ and $d = 10$, by sampling every element of it from a Gaussian distribution $\mathcal{G}(0, 1)$. The value of d is intentionally low so that there appear correlations between the attributes, as is usually the case in real data. For each class t,

---

[1]The code can be found at http://romera-paredes.com/zsl.

Fig. 2.2 Multiclass
accuracy obtained by DAP
[21], and ESZSL
(Sect. 2.3.1), when varying
the number of seen classes,
z. *Vertical bars* indicate ± 1
standard deviation
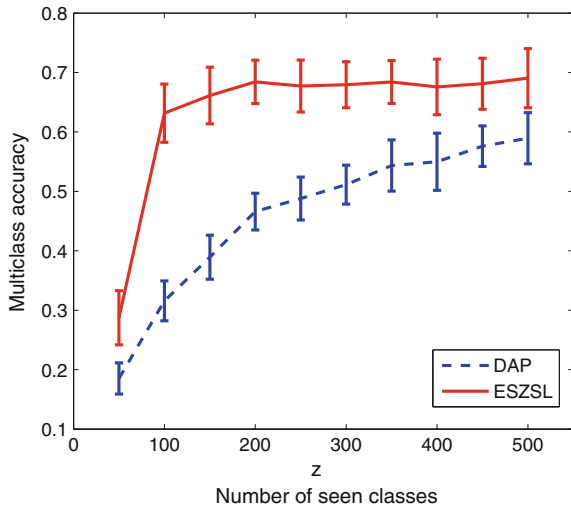
we created 50 instances by first generating their representation in the attribute space
by adding Gaussian noise, $\mathcal{G}(0, 0.1)$ to the attribute signature $\mathbf{S}_t$, then we brought
them back onto the original feature space by using $\mathbf{V}^+$. Following this process, we
generated a training set composed of z seen classes, and a test and validation set
composed of 100 unseen classes each.

In the first experiment, we evaluated how the number of seen classes affected
the performance of the methods on unseen classes. To do so, we varied the number
of seen classes from 50 to 500 in intervals of 50. According to the results shown
in Fig. 2.2, we can see that ESZSL significantly outperforms DAP in all cases. It is
remarkable that the performance of ESZSL with 100 seen classes is superior to the
performance of DAP with 500 seen classes. We also observe that the performance
of ESZSL plateaus when the number of seen classes is above 200, possibly because
there is no further margin of improvement.

In Sect. 2.3 we argue that the described approach should be robust to attributes
having different discriminative capabilities for characterising the classes. In the sec-
ond experiment, we assess how the approaches perform in the extreme case where
some attributes provide no information at all about the classes at hand. The way we
have implemented this is by first, synthesising a dataset just as described above, and
second, by randomly selecting a set of attributes (without replacement) so that their
information in all signatures is corrupted. In particular let us define by $\mathcal{A}$ the set of all
attributes, with cardinality $|\mathcal{A}| = a$. From this set $\mathcal{A}$ we randomly sample ψ mislead-
ing attributes, creating the set $\Psi \subseteq \mathcal{A}$, $|\Psi| = \psi$. The way each of the inputs of the
attributes in Ψ is corrupted is again by sampling from a Bernoulli distribution with
0.5 mean. In this experiment we have tried different values of ψ in the range of 5–45
attributes (out of 100), in intervals of 5. The results, reported in Fig. 2.3, show that our
method significantly outperforms the baseline. For example we observe that when

Fig. 2.3 Multiclass accuracy obtained by DAP [21], and ESZSL (Sect. 2.3.1), when varying the number of corrupted attributes, ψ. *Vertical bars* indicate ± 1 standard deviation
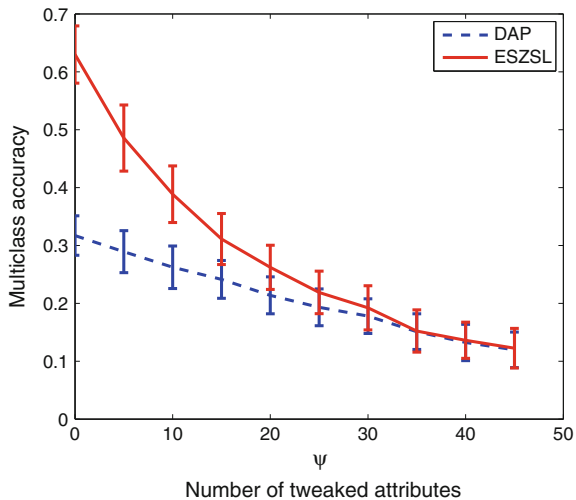
having 15 misleading attributes, our method achieves a comparable performance as the baseline with none misleading attributes.

2.5.2   Real Data Experiments

We have tried the same real datasets as the ones reported in [16] which are the Animals with Attributes dataset (AwA) [21], the SUN scene attributes database (SUN) [30] described in Chap. 11, and the aPascal/aYahoo objects dataset (aPY) [10]. These consist of collections of images comprising a varied set of categories in different scopes: animals, scenes, and objects, respectively. The AwA dataset contains attribute-labelled classes, which we will use as **S** in the model. The datasets aPY and SUN are attribute-labelled instances datasets, so the attribute signature of each class is calculated as the average attribute signature of the instances belonging to that class. The characteristics of each of these datasets are summarised in Table 2.1.

Table 2.1 Summary of the real datasets employed in the experimental section

	AwA	aPY	SUN
Attributes	85	65	102
Seen classes	40	20	707
Unseen classes	10	12	10
Instances	30, 475	15, 339	14, 340

In the following we perform three sets of experiments. In the first one, we compare our approach with alike methods that also belong to the framework described in Fig. 2.1. In the second set of experiments, we compare our approach against the current state of the art. Finally in the last experiment, we compare our approach and a standard classification method, for attributes prediction. The aim here is to assess whether the good results in zero-shot learning come at the expense of attribute prediction performance. In all cases, in order to tune the hyper-parameters of the methods, we use the following validation procedure. We create the validation set by grouping all instances belonging to 20 % of the classes in the training partition, chosen at random (without replacement). Once the hyper-parameters are tuned, we pool the validation set instances together with the training set instances in order to train the final model. We use the range of values, 10^b for $b = -3, -2, \ldots, 2, 3$ to tune all hyper-parameters.

2.5.2.1 Preliminary Experiments

Here we present an experiment comparing our approach to [1]. We used the recently provided VGG network features [34], of the AwA dataset. This dataset also provides both binary and continuous versions of the attributes signatures. Here, we compare these two scenarios. We utilised the best configuration reported on [1], using different training set sizes of 500, 1000, and 2000 instances. The results are shown in Table 2.2. As expected, both approaches perform better when the attributes signatures are continuous. In any case, our approach clearly outperforms [1] in all cases. It is also worth mentioning that the approach in [1] took more than 11 hours to run the scenario with 2000 training instances, whereas ours only took 4.12 s.

2.5.2.2 Comparison with the State of the Art

In order to make our approach easily comparable with the state of the art, we used the set of standard features provided by the authors of the data [16, 21, 30], including SIFT [25], and PHOG [6]. We used combined χ^2-kernels, one for each feature

Table 2.2 Comparison between the approach in [1] and ESZSL, using VGG features extracted from the AwA dataset, utilising binary attributes signatures (Left), and continuous attributes signatures (Right)

Training instances	Binary attributes		Continuous attributes	
	[1]	ESZSL	[1]	ESZSL
500	33.30 %	**33.85 %**	47.31 %	**51.63 %**
1000	39.02 %	**43.16 %**	49.40 %	**53.87 %**
2000	41.02 %	**46.89 %**	54.09 %	**56.99 %**

Table 2.3 Multiclass accuracy obtained by DAP [21], ZSRwUA [16], the method described in Sect. 2.3.1 ESZSL, and its modification ESZSL-AS, on the three real datasets described in Table 2.1

Method/Dataset	AwA	aPY	SUN
DAP	40.50	18.12	52.50
ZSRwUA	43.01 ± 0.07	26.02 ± 0.05	56.18 ± 0.27
ESZSL	**49.30 ± 0.21**	15.11 ± 2.24	**65.75 ± 0.51**
ESZSL-AS	–	**27.27 ± 1.62**	61.53 ± 1.03

channel,[2] following the procedure explained in [16, 21]. In all cases, we used the same attributes signatures, and the same standard partitions between seen and unseen classes, as the ones employed in [16].

In these experiments we compare 4 methods: DAP [21], ZSRwUA [16], ESZSL (Sect. 2.3.1), and a small modification of the latter that we call ESZSL All Signatures (ESZSL-AS).

ESZSL-AS can be applied in attribute-labelled instances datasets (aPY and SUN), and consists in treating each training attribute signature as a class in its own right. That is effectively done by removing Y in Eq. (2.6), where now $S \in \mathbb{R}^{a \times m}$ contains as many signatures as the number of training instances. The inference process remains the same, and the unseen class signatures are used to predict the category.

For each dataset we ran 20 trials, and we report the mean and the standard deviation of the multiclass accuracy in Table 2.3. Overall we notice that the approaches described in Sect. 2.3 significantly outperform the state of the art.

In the AwA dataset, ESZSL achieves an absolute improvement over 6 % over the state of the art. Even more surprising, this performance is better than state-of-the-art approaches applied to discovered (non-semantic) attributes, which according to [16] is 48.7. Let us recall that this dataset contains attribute-labelled classes, and so, ESZSL-AS cannot be applied here.

Regarding the aPY dataset, the standard ESZSL approach has struggled and it is not able to outperform the DAP baseline. One hypothesis is that the small number of classes in comparison to the number of attributes has probably affected negatively the performance. In contrast we see that ESZSL-AS obtains state-of-the-art results, achieving a 1.25 % of improvement over the previous best approach. Its success can be explained by reversing the previous reasoning about why standard ESZSL failed. Indeed, ESZSL-AS effectively considers as many seen classes as the number of training instances.

Finally, in the SUN dataset both ESZSL approaches obtain extremely good results, significantly outperforming the current state of the art. ESZSL leads the table, achieving an improvement of 9.6 %. We note that here the number of seen classes is much bigger than the number of attributes, therefore the advantages obtained by ESZSL-AS in the previous experiment vanish.

---

[2]Available at www.ist.ac.at/chl/ABC.

Table 2.4 Comparison between SVM (Learning attributes directly), and ESZSL, for attributes prediction, using mean average precision as a measure

Mean Average Precision	AwA	aPY	SUN
Learning attributes directly	**56.95** %	**30.78** %	**79.36** %
Using $\mathbf{X}^\top \mathbf{V}$ from ESZSL	50.73 %	29.51 %	68.53 %

2.5.2.3 Attributes Prediction

The focus of our model is on maximising the multiclass accuracy among the classes at hand. However, as a byproduct of the learning process, we can also use V as a way to predict attributes. In this experiment we check whether these attribute predictors are effective, or on the contrary, the gain in zero-shot performance comes at the expense of attribute prediction. In order to do so, we compare the described option with a simple approach that learns an SVM for each attribute directly. The results are reported in Table 2.4.

The gain in ZSL performance comes at the expense of attribute prediction. This may be because our approach tends to neglect the attributes that are unreliable or useless for class prediction, whereas in attribute prediction all are considered equally important. These results are in the same vein as the ones reported in [1].

2.6 Discussion

In this work, we have described an extremely simple approach for ZSL that is able to outperform by a significant margin the current state of the art approaches on a standard collection of ZSL datasets. It combines a linear model together with a principled choice of regularizers that allow for a simple and efficient implementation.

We have also made explicit a connection between ZSL and domain adaptation. In particular, we have expressed the framework described in Sect. 2.3 as a domain adaptation problem. As a consequence, we are able to translate theoretical developments from domain adaptation to ZSL.

Given the simplicity of the approach, there are many different research lines that can be pursued. In this work we focus on semantically meaningful attributes, but the development of similar ideas applied to word embeddings as in [12], is both promising and straightforward within this framework. Another interesting research line is to study the addition of nonlinearities and more layers into the model, leading to a deep neural network where the top layer is fixed and interchangeable, and all the remaining layers are learned. Recent works exploring this direction are [18, 37].

As a concluding comment, we acknowledge that many problems require complex solutions, but that does not mean that simple baselines should be ignored. On the contrary, simple but strong baselines both bring light about which paths to follow in order to build more sophisticated solutions, and also provide a way to measure the quality of these solutions.

References

1. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for attribute-based classification. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
2. Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. Mach. Learn. (ML) **73**(3), 243–272 (2008)
3. Argyriou, A., Micchelli, C.A., Pontil, M.: When is there a representer theorem? vector versus matrix regularizers. J. Mach. Learn. Res. (JMLR) **10**, 2507–2529 (2009)
4. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Mach. Learn. (ML) **79**(1–2), 151–175 (2010)
5. Ben-david, S., Blitzer, J., Crammer, K., Sokolova, P.M.: Analysis of representations for domain adaptation. In: Conference on Neural Information Processing Systems (NIPS) (2006)
6. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: International conference on Image and video retrieval (CIVR) (2007)
7. Croonenborghs, T., Driessens, K., Bruynooghe, M.: Learning relational options for inductive transfer in relational reinforcement learning. In: International conference on Inductive logic programming (ILP) (2008)
8. Daumé III, H.: Frustratingly easy domain adaptation. In: Annual Meeting of the Association of Computational Linguistics (ACL) (2007)
9. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. J. Artif. Intell. Res. **2**(1), 263–286 (1994)
10. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
11. Ferrari, V., Zisserman, A.: Learning visual attributes. In: Conference on Neural Information Processing Systems (NIPS) (2007)
12. Frome, A., Corrado, G., Shlens, J., Bengio, S., Dean, J., Ranzato, M., Mikolov, T.: Devise: a deep visual-semantic embedding model. In: Conference on Neural Information Processing Systems (NIPS) (2013)
13. Fu, Y., Hospedales, T.M., Xiang, T., Gong, S.: Learning multimodal latent attributes. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **36**(2), 303–316 (2014)
14. Hariharan, B., Vishwanathan, S., Varma, M.: Efficient max-margin multi-label classification with applications to zero-shot learning. Mach. Learn. (ML) **88**(1), 127–155 (2011)
15. Hwang, S.J., Sha, F., Grauman, K.: Sharing features between objects and their attributes. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2011)
16. Jayaraman, D., Grauman, K.: Zero-shot recognition with unreliable attributes. In: Conference on Neural Information Processing Systems (NIPS) (2014)
17. Jayaraman, D., Sha, F., Grauman, K.: Decorrelating semantic visual attributes by resisting the urge to share. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
18. Jetley, S., Romera-Paredes, B., Jayasumana, S., Torr, P.H.: Prototypical priors: from improving classification to zero-shot learning. In: British Machine Vision Conference (BMVC) (2015)

19. Jiang, J., Zhai, C.: Instance weighting for domain adaptation in nlp. In: Annual Meeting of the Association of Computational Linguistics (ACL) (2007)
20. Kifer, D., Ben-David, S., Gehrke, J.: Detecting change in data streams. In: International Conference on Very Large Data Bases (VLDB) (2004)
21. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
22. Lampert, C.H., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **36**(3), 453–465 (2014)
23. Lawrence, N.D., Platt, J.C.: Learning to learn with the informative vector machine. In: International Conference on Machine Learning (ICML) (2004)
24. Liu, J., Kuipers, B., Savarese, S.: Recognizing human actions by attributes. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2011)
25. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. (IJCV) **60**(2), 91–110 (2004)
26. Mahajan, D., Sellamanickam, S., Nair, V.: A joint learning framework for attribute models and object descriptions. In: International Conference on Computer Vision (ICCV) (2011)
27. Murphy, G.: The Big Book of Concepts. The MIT Press (2004)
28. Palatucci, M., Hinton, G., Pomerleau, D., Mitchell, T.M.: Zero-shot learning with semantic output codes. In: Conference on Neural Information Processing Systems (NIPS) (2009)
29. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
30. Patterson, G., Hays, J.: Sun attribute database: Discovering, annotating, and recognizing scene attributes. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
31. Raykar, V.C., Krishnapuram, B., Bi, J., Dundar, M., Rao, R.B.: Bayesian multiple instance learning: automatic feature selection and inductive transfer. In: International Conference on Machine Learning (ICML) (2008)
32. Romera-Paredes, B., Aung, H., Bianchi-Berthouze, N., Pontil, M.: Multilinear multitask learning. In: International Conference on Machine Learning (ICML) (2013)
33. Rückert, U., Kramer, S.: Kernel-based inductive transfer. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML–PKDD) (2008)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR) (2015)
35. Suzuki, M., Sato, H., Oyama, S., Kurihara, M.: Transfer learning based on the observation probability of each attribute. In: International Conference on Systems, Man and Cybernetics (2014)
36. Wang, Y., Mori, G.: A discriminative latent model of object classes and attributes. In: European Conference on Computer Vision (ECCV) (2010)
37. Yang, Y., Hospedales, T.M.: A unified perspective on multi-domain and multi-task learning. In: International Conference on Learning Representations (ICLR) (2015)