Pattern Recognition Letters
journal homepage: www.elsevier.com

Generic Compact Representation Through Visual-Semantic Ambiguity Removal

Yang Long[a,**], Yu Guan[a], Ling Shao[b]

[a] *Open Lab, School of Computing, Newcastle University, Newcastle upon Tyne, UK*
[b] *JD Artificial Intelligence Research (JDAIR), Beijing, P.R.China*

ABSTRACT

Zero-Shot Hashing (ZSH) aims to learn compact binary codes that can preserve semantic contents of the images from unseen categories. Conventional approaches project visual features to a semantic space that is shared by both seen and unseen categories. However, we observe that such a one-way paradigm suffers from the *visual-semantic ambiguity* problem. Namely, the semantic concepts (e.g. attributes) cannot explicitly correspond to visual patterns, and vice versa. Such a problem can lead to a huge variance in the visual features for each attribute. In this paper, we investigate how to remove such semantic ambiguity based on the observed visual appearances. In particular, we propose (1) a novel latent attribute space to mitigate the gap between visual appearances and semantic expressions; (2) a dual-graph regularised embedding algorithm called ***V**isual-**S**emantic **A**mbiguity **R**emoval* (VSAR) that can simultaneously extract the shared components between visual and semantic information and mutually align the data distribution based on the intrinsic local structures of both spaces; (3) a new zero-shot hashing framework that can deal with both instance-level and category-level tasks. We validate our method on four popular benchmarks. Extensive experiments demonstrate that our proposed approach significantly performs the state-of-the-art methods.

1. Introduction

Due to the decreasing cost of digital devices, it gas become a challenge to accurately organise and retrieve those daily generated large-scale images and videos. Scalable Content-based Image Retrieval relies on learning compact binary codes that can preserve the semantic contents so as to achieve high-quality efficient retrieval. Conventional hashing algorithms Yu et al. (2016a); Liu and Shao (2016) mainly explore the intrinsic data structure and aims to preserve such information in the learnt codes using an unsupervised manner. Recent approaches embrace the deep network architecture and takes the supervised label information in to account and achieve extraordinary performance Zhang et al. (2018). However, the representation is only sensitive to training categories. Due to the high cost of human annotation, it is intractable to collect sufficient training images for the ever growing number of categories.

Zero-Shot Hashing (ZSH) aims to learn compact binary codes for unseen categories. The fundamental idea of ZSH is to train a closed-set of human knowledge models that can generalise to an ever growing set of classes without collecting new training images. Such a scenario effectively alleviates the cost of data collection and also provides a feasible solution for recognising inaccessible objects, such as an ancient species that only has text records. Because of these attractive properties, Zero-shot Learning (ZSL) has aroused increasing research interests in the computer vision and machine learning community Yu et al. (2013); Fu et al. (2014b); Palatucci et al. (2009).

Conventional ZSL methods Lampert et al. (2009); Yu et al. (2013); Fu et al. (2015); Farhadi et al. (2009); Socher et al. (2013) rely on directly mapping the visual features to a human-interpretable semantic space and the labels are inferred through human knowledge. However, an inevitable issue of using semantic information is the *ambiguity* problem. In linguistics, a concept is considered ambiguous if its extension is deemed lacking in clarity. It is the uncertainty about which objects belong to the concept or which exhibit characteristics that have this predicate. In the context of ZSL, **Visual-Semantic Ambiguity** refers to the situation that a semantic concept (e.g. an attribute) cannot clearly correspond to a certain pattern of visual data, and vice versa. Therefore, the paradox is how different of

**Corresponding author: Tel.: +44-746-235-5329;
e-mail: yang.long@ieee.org (Yang Long)

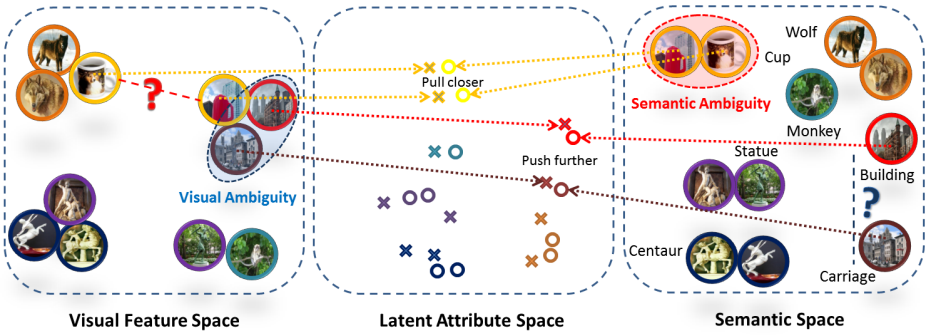

Fig. 1. An intuitive illustration of VSAR (best viewed in colour). Visual Ambiguity (in blue oval): the image of a carriage is taken with a building background. It cannot recover the semantic distance (blue question mark) to the building category. Semantic Ambiguity (in red oval): the cup printed with a wolf and the cup-like building share the same semantic expression which can lead to a large visual variance (the red question mark). After embedding to the latent attribute space using VSAR, such ambiguity is mitigated.

the visual patterns can we tolerate for each semantic concept? Alternatively, should we split the concept into sub-concepts to fit the visual data? This is known as the Sorites Paradox that can lead to two extreme solutions. (1) We can accept all instances as if they have the same attribute. Jayaraman and Grauman Jayaraman and Grauman (2014) also study this problem. They provide an extreme example that the concept 'bumpy' is assigned to both 'bumpy road' and 'bumpy rash' which can lead to unreasonable classification results. Unfortunately, most of the existing methods accept this solution. (2) We refuse any ambiguity and give every seen instance a unique attribute. For example, compared to 'smile', 'Mona Lisa's smile' is clearly referring to a unique visual pattern with no ambiguity. However, it is infeasible to treat everything as unique and assign a new concept to it.

Instead of debating on what is common or unique, in this paper, we propose a latent attribute space to mitigate the visual-semantic ambiguity using a novel algorithm named ***Visual-Semantic Ambiguity Removal*** (VSAR). We measure the visual-semantic ambiguity by the reconstruction error and correct it in the latent attribute space. Intuitively, if a semantic concept refers to multiple variations of visual features, it should be split into different regions in the latent attribute space. In the visual aspect, if two close feature points are labelled by different attributes, we should find lower-dimensional subspaces so that they can be discriminated after embedding. Specifically, we develop a graph regularised embedding function that can minimise the reconstruction errors in both visual and semantic spaces. Meanwhile, the regularisation can preserve the discriminative information for recognising unseen categories. We illustrate this idea in Fig. 1.

This paper aims to systematically the characteristics of the proposed VSAR and its benefits to learn generic compact representation for unseen classes. Our contribution is summarised as follows.

- We propose a novel VSAR algorithm that can simultaneously remove the ambiguity between visual and semantic information.
- Our approaches are extensively evaluated on both ZSL and ZSH benchmarks. Results suggest the important role of visual-semantic ambiguity to the performance improvement.
- We introduce a unified framework that can deal with both category-level and instance-level zero-shot learning tasks without adjusting the paradigm.

To the best of our knowledge, the visual-semantic ambiguity issue has not been well studied yet. Thus, in the following, we only review related ZSL approaches.

2. Related Work

Since learning visual attributes Ferrari and Zisserman (2007) is proposed, extensive studies Lampert et al. (2009); Palatucci et al. (2009); Socher et al. (2013); Kankuekul et al. (2012) have been conducted on how to use attributes as an intermediate representation for ZSL tasks. One interesting direction is to investigate the properties of attributes, such as the label co-occurrence property Mensink et al. (2014), the relativeness Parikh and Grauman (2011), the unreliability Jayaraman and Grauman (2014), and the correlation problem Jayaraman et al. (2014) of human-nameable attributes. All of these are semantic properties and therefore suffer from the semantic-visual ambiguity problem. Due to this problem, some work turns to abandon human-nameable attributes and discovers data-driven attributes Yu et al. (2013); Kumar et al. (2009). However, for ZSL, these methods cannot exploit existing attribute ontologies. Hence, the applicable area is limited. Another trend is based on the embedding framework Akata et al. (2013); Romera-Paredes and Torr (2015); Mahajan et al. (2011); Zhang and Saligrama (2015); Al-Halah et al. (2014); Liu et al. (2015); Cai et al. (2015). All these methods follow the restricted one-way paradigm that suffers from the ambiguity between low-level instances and high-level semantic concepts and labels. Recently, a new direction of ZSL is using the transductive model Guo et al. (2016); Li et al. (2015); Kodirov et al. (2015); Fu et al. (2014a,b); Shao et al. (2016); Long et al. (2016); Yu et al. (2016b). Unlabelled target domain data is collected for learning a transfer function. However, this setting slightly differs from the original ZSL purpose because the target domain may be strictly inaccessible. In contrast, our method can exploit the

extensive existing attribute ontology while also stressing the existence of visual-semantic ambiguity and removing it through a learning process.

3. Visual-Semantic Ambiguity Removal

Problem setup: The training data is in N 3-tuples of 'seen' samples, attributes, and category labels: $(x_1, a_1, y_1), ..., (x_N, a_N, y_N) \subseteq \mathcal{X}_s \times \mathcal{A}_s \times \mathcal{Y}_s$, where $\mathcal{X}_s$ is a D-dimensional feature space $\mathcal{X}_s = [x_{dn}] \in \mathbb{R}^{D \times N}$, $\mathcal{A}_s$ is a M-dimensional attribute space $\mathcal{A}_s = [a_{mn}] \in \mathbb{R}^{M \times N}$, and $y_n \in \{1, ..., C\}$ consists of C discrete categories. The Calligraphic typeface indicates a space. We use subscript u to denote information of 'unseen' space and *hat* denotes 'unseen' samples. During testing, the preliminary knowledge is in $\hat{C}$ pairs of 'unseen' category-level attributes and labels: $(\hat{a}_1, \hat{y}_1), ..., (\hat{a}_{\hat{C}}, \hat{y}_{\hat{C}}) \subseteq \mathcal{A}_u \times \mathcal{Y}_u$, $\mathcal{Y} \cap \mathcal{Y}_u = \varnothing$, $\mathcal{A}_u = [a_{m\hat{c}}] \in \mathbb{R}^{M \times \hat{C}}$. The goal is to learn a classifier, $f : \mathcal{X}_u \rightarrow \mathcal{Y}_u$, where the samples in $\mathcal{X}_u$ are completely unavailable during training. Such a problem is known as zero-shot learning.

Latent Attribute Embedding: We aim to discover a latent attribute embedding space $\mathcal{V}$ shared by both visual and semantic spaces $\mathcal{X}$ and $\mathcal{A}$ to mitigate the visual-semantic ambiguity. During testing, both $\mathcal{X}_u$ and $\mathcal{A}_u$ can be embedded into $\mathcal{V}$.

Zero-shot Recognition: Instead of typical two-step prediction $\mathcal{X}_u \rightarrow \mathcal{A}_u \rightarrow \mathcal{Y}_u$, our embedding is two-way from $\mathcal{X}_u$ and $\mathcal{A}_u$. Because attribute space $\mathcal{A}_u$ and label space $\mathcal{Y}_u$ are in pairs, we can firstly embed the known $\mathcal{A}_u$ to $\mathcal{V}$ as a knowledge domain. During testing, an unseen image $\hat{x}$ is also embedded to $\mathcal{V}$ so that we can compute the index, i.e., $\mathcal{X}_u \rightarrow \mathcal{V} \leftarrow \mathcal{A}_u \leftarrow \mathcal{Y}_u$.

3.1. Latent Attribute Embedding

This is the core component to deal with the visual-semantic ambiguity. We require $\mathcal{X}_s$ and $\mathcal{A}_s$ to compute $\mathcal{V}$. *In the following, we drop the subscript s for convenience, i.e. we replace* $\{\mathcal{X}_s, \mathcal{A}_s, \mathcal{Y}_s\}$ *by* $\{\mathcal{X}, \mathcal{A}, \mathcal{Y}\}$. Typically, each dimension a_m denotes a human-nameable concept, where $M \ll D$. The attribute notions here are instance-level. For the category-level, we can simply set the same attribute vectors to the instances within the same class. For embedding, many previous works are based on a forward matrix transformation, i.e. $\mathcal{X}$ to $\mathcal{A}$. However, because of the visual-semantic ambiguity, the variance in $\mathcal{X}$ is large. Therefore, the forward embedding is difficult to be reconstructed by a backward inverse matrix transformation from $\mathcal{A}$. Therefore, we insert an intermediate latent attribute space $\mathcal{V}$ between $\mathcal{X}$ and $\mathcal{A}$, where $\mathcal{V} = [v_{kn}] \in \mathbb{R}^{K \times N}$. K is the dimension of the embedding space. A straightforward setting is $M \leqslant K \leqslant A$. However, we stress that K can be any positive whole number. Specifically, we introduce our loss function as:

$$J = \|\mathcal{X} - U_1\mathcal{V}\|_F^2 + \alpha\|\mathcal{A} - U_2\mathcal{V}\|_F^2, \tag{1}$$

where $\|.\|_F$ is the Frobenius norm of a matrix, which estimates the Euclidean distance between two matrices. The shared embedding space $\mathcal{V}$ is decomposed from both $\mathcal{X}$ and $\mathcal{A}$, where $U_1 = [u_{1_{dk}}] \in \mathbb{R}^{D \times K}$ and $U_2 = [u_{2_{mk}}] \in \mathbb{R}^{M \times K}$ are the basis matrices of the visual feature and attribute space, respectively.

Using Eq. 1, it becomes easier to understand the properties of the latent attribute space and how it could mitigate the visual-semantic ambiguity. Optimising Eq. 1 aims to minimise the reconstruction errors that are from $\mathcal{V}$ to $\mathcal{X}$ and from $\mathcal{V}$ to $\mathcal{A}$, respectively. To achieve the optimal solution, U_1 and U_2 should preserve the principal components between $\mathcal{X}$ and $\mathcal{A}$. This differs from unsupervised methods, such as PCA, that only analyse the data structure in a single domain. Our Eq. 1 can reduce the variance of the embedded data that comes from both visual and semantic domains. α is a reliability parameter that can balance the strengths of the two terms. In practice, if the attribute space is known as unreliable in prior, e.g. extended from category-level attributes, we can reduce α so that the proposed embedding can focus more on the visual feature space and remove more ambiguity from the attribute space.

3.2. Dual-graph Regularisation

The above Eq. 1 can reduce the difference between the data structures of $\mathcal{X}$ and $\mathcal{A}$. However, it cannot preserve the discriminative information. For instance, if the gap between x_n and a_n is too large, their corresponding weights tend to be minimised to very small values. As a result, the learnt latent attributes are the principal components that are shared by all of the categories. For the purpose of zero-shot recognition, we have to preserve the intrinsic geometrical structure so that the learnt representation is discriminative.

We achieve this goal by taking the local invariance assumption and model the problem through a spectral graph approach named *Dual-graph Regularisation*. In particular, this is a combination of two supervised graphs that model the relationship between $\mathcal{X}$ and $\mathcal{Y}$, and $\mathcal{A}$ and $\mathcal{Y}$. The main criteria is to preserve the local structures. Therefore, we need the two graphs to simultaneously estimate the data structures of both spaces. Each graph has N vertices that correspond to N data points in the training set. As mentioned earlier, our method can effectively handle ZSL tasks for both instance-level and category-level attribute scenarios. In particular, for *instance-level attributes*, we put an edge between each data point x_n or a_n and its p nearest neighbours. For each pair of the vertices s_i and s_j in the weight matrix, $w_{ij} = 1$ if and only if s_i and s_j are connected by an edge, otherwise, $w_{ij} = 0$. As a result, we can separately compute two weight matrices $W_{\mathcal{X}}$ and $W_{\mathcal{A}}$.

It is noteworthy that for *category-level attributes*, $W_{\mathcal{A}}$ is computed slightly different. Every vertex in the same category are connected by a normalised edge, i.e. $w_{ij} = p/n_c$, if and only if a_i and a_j are from the same category c, where n_c is the size of category c.

In the embedding space $\mathcal{V}$, we expect that if the s_i and s_j in both graphs are connected, each pair of embedded points v_i and v_j are also closed to each other. However, for the *visual-semantic ambiguity* problem, $W_{\mathcal{X}}$ and $W_{\mathcal{A}}$ usually give contradictory results. To compromise such conflict, we use the same

reliability parameter α in Eq. 1 to linearly combine the two graphs, i.e. $W_{ij} = W_{\mathcal{X}_{ij}} + \alpha W_{\mathcal{A}_{ij}}$. The resulted regularisation is:

$$\begin{aligned}\mathcal{R} &= \frac{1}{2}\sum_{i,j=1}^{N}\|v_i - v_j\|^2 w_{ij} \\ &= Tr(\mathcal{V}D\mathcal{V}^T) - Tr(\mathcal{V}W\mathcal{V}^T) = Tr(\mathcal{V}L\mathcal{V}^T),\end{aligned} \tag{2}$$

where D is the degree matrix of W, $D_{ii} = \sum_i w_{ij}$. L is known as graph Laplacian matrix $L = D - W$ and $Tr(.)$ computes the trace of a matrix. We combine Eq. 1 and 2 using a regularisation parameter λ to control the balance between reconstruction error and local structure preservation. The final goal is to optimise the following equation:

$$J = \|\mathcal{X} - U_1\mathcal{V}\|_F^2 + \alpha\|\mathcal{A} - U_2\mathcal{V}\|_F^2 + \lambda Tr(\mathcal{V}L\mathcal{V}^T), \tag{3}$$

3.3. Optimisation Strategy

Each term of the above Eq. 3 is convex, but the combined expression of $U_1, U_2, \mathcal{V}$ is non-convex. To our best knowledge, there is no direct solution to find the global optima. Instead, we adopt an alternating optimisation strategy to find the local minima for each term separately as a relaxed solution. Specifically, the whole task is in turn separated into three sub-problems.

1. sub-problem U_1: Suppose we compute the partial derivative of the overall loss function J with respect to U_1, U_2 and $\mathcal{V}$ are fixed as constants. It then becomes a standard least squares problem. Let the partial derivative equal to zero, we have the closed form solution:

$$\begin{aligned}\frac{\partial J}{\partial U_1} &= -2\mathcal{X}\mathcal{V}^T + 2U_1\mathcal{V}\mathcal{V}^T = 0 \\ U_1 &= \mathcal{X}\mathcal{V}^T\left(\mathcal{V}\mathcal{V}^T\right)^{-1}.\end{aligned} \tag{4}$$

2. sub-problem U_2: Similar to the sub-problem 1, we can fix U_1 and $\mathcal{V}$, and compute the partial derivative of J with respect to U_2. The corresponding solution is:

$$U_2 = \mathcal{A}\mathcal{V}^T\left(\mathcal{V}\mathcal{V}^T\right)^{-1}. \tag{5}$$

Since we do not expect any prior bias from the unnormalised magnitudes of the training data, the basis vectors in the matrices should be normalised to unit vectors via:

$$u_{1_{dk}} \leftarrow \frac{u_{1_{dk}}}{\sqrt{\sum_d u_{1_{dk}}^2}} \quad u_{2_{mk}} \leftarrow \frac{u_{1_{mk}}}{\sqrt{\sum_m u_{2_{mk}}^2}}.$$

3. sub-problem $\mathcal{V}$: Fix U_1 and U_2, we can then update $\mathcal{V}$. Applying the matrix properties $Tr(AB) = Tr(BA)$ and $Tr(A^T) = Tr(A)$, and we set the partial derivative respect to $\mathcal{V}$ to zero:

$$\frac{\partial J}{\partial \mathcal{V}} = 2\left(\left(U_1^T U_1 + \alpha U_2^T U_2\right)\mathcal{V} + \mathcal{V}(\lambda L) - \left(U_1^T\mathcal{X} + \alpha U_2^T\mathcal{A}\right)\right) = 0. \tag{6}$$

Since space U_1, U_2 and L are disjointed, this forms a typical Sylvester equation that has the unique solution for $\mathcal{V}$. We use the *lyap*() function in MATLAB to solve this problem.

Batch sampling scheme: In practice, the computational complexity of solving the Eq. 6 is $O(N^3)$. To improve the efficiency, we adopt a batch sampling scheme like the deep learning strategy. The whole training set is divided into t batches by randomly sampling training instances from each categories. The size of each batch roughly equals to $\frac{N}{t}$. As a result, the computational complexity is reduced to $O\left(t\left(\frac{N}{t}\right)^3\right)$, where $(\frac{N}{t})^3 \ll N^3$. Each batch is in turn used to optimise the loss function in Eq. 3. We turn to the next batch until it converges on the previous batch. The whole learning procedure is summarised in Algorithm 1.

Algorithm 1 : Visual-Semantic Ambiguity Removal

Require: $\{\mathcal{X}, \mathcal{A}, \mathcal{Y}\}$, α, λ, K, p, number of batch t.
Ensure: $\mathcal{V}, U_1,$ and U_2.

```
Initialisation: random batch sampling {X_1, A_1, Y_1}...{X_i, A_i, Y_i},
  random initial matrix V.
for each batch do
  Compute the graph Laplacian matrix L using Eq. 2;
  while Eq. 3 is not converged do
    Update U_1 by Eq. 4, then normalise U_1 by u_{1_dk} ← u_{1_dk} / sqrt(Σ_d u_{1_dk}^2);
    Update U_2 by Eq. 5, then normalise U_2 by u_{2_mk} ← u_{1_mk} / sqrt(Σ_m u_{2_mk}^2);
    Update V by Eq. 6;
  end while
end for
return V, U_1, and U_2;
```

3.4. Zero-shot Recognition and Retrieval

Once we obtain the latent attribute embedding $\mathcal{V}$ of the seen data, performing zero-shot recognition is straightforward via *least-square approximation* between $\mathcal{V}$ and $\{\mathcal{A}, \mathcal{X}\}$. During the test, the given informations are the unseen category names and their attributes in pairs: $\{\mathcal{Y}_u, \mathcal{A}_u\}$. We firstly embed all unseen attributes $\mathcal{A}_u$ into the latent embedding space as references: $\mathcal{V}_u = \mathcal{V}\mathcal{A}^T(\mathcal{A}\mathcal{A}^T)^{-1}\mathcal{A}_u$. Given a test unseen instance $\hat{x}$, its embedded latent attribute representation is: $\hat{v} = \mathcal{V}\mathcal{X}^T(\mathcal{X}\mathcal{X}^T)^{-1}\hat{x}$. Finally, we adopt a simple NN classifier to predict the category label $\hat{c}$:

$$\hat{c} = \arg\min_c \|\hat{v} - v_c\|^2, \text{ where } v_c \in \mathcal{V}_u. \tag{7}$$

The above equation can estimate the probability to which class a test instance belongs to. However, the representation is in real numbers. In this paper, we adopt the typical sign function to convert the learnt generic representation in $\mathcal{V}$ into binary codes: $sgn(v)$ which are then readily for scalable image classification and retrieval.

4. Experiments

We provide a comprehensive evaluation of our model for both ZSL and ZSH tasks on four popular datasets. AwA is one

Table 1. Key statistics of the four datasets.

Dataset	# of attributes	Attribute Type	Annotation Level	# of Seen Classes	# of unseen classes	# of total images
AwA	85	Both	per class	40	10	30475
CUB	312	Binary	Both	150	50	11788
aPY	64	Binary	per image	20	12	15339
SUN	102	Continues	per image	707	10	14340

Table 2. Compare with the published state-of-the-art methods.

Method	aPascal&aYahoo	Animals with Attributes
Farhadi et al. (2009)	32.5	-
Mahajan et al. (2011)	37.93	-
Akata et al. (2013)	-	43.5
Jayaraman et al. (2014)	-	47.1
Lampert et al. (2009)	19.1	40.5
Jayaraman and Grauman (2014)	26.02 ± 0.05	43.01 ± 0.07
Romera-Paredes and Torr (2015)	27.27 ± 1.62	49.30 ± 0.21
our VSAR	**39.42± 0.27**	**51.75± 0.43**

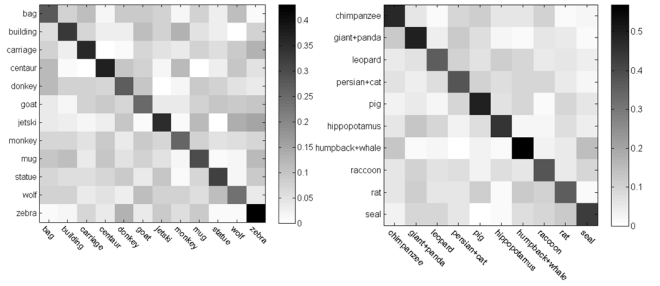

Fig. 2. Confusion matrix of ZSL performance on aPY (left) and AwA (right).

of the earliest work that particularly proposed for ZSL tasks. Many published results are based on this dataset. Each animal category in AwA is labelled by an attribute signature. aPY is an instance-level attribute dataset that each image has a unique attribute signature. In contrast to AwA, aPY covers a more various range of categories, including human, artificial objects, buildings, as well as animals. For comparison reason, we adopt the base features that are provided by the datasets. We carefully follow the standard settings on both of the datasets. In particular, the training/test splits are 40/10 and 20/12 on AwA and aPY dataset, respectively. The optimal reliability parameter α for each dataset is selected from one of $\{0.1, ..., 0.5, ..., 0.9\}$ with the step of 0.1 which yields the best performance by 10-fold cross-validation on the training data. For λ and p, cross-validation is still deployed and finally fixed as $\lambda = 0.03$ and $p = 10$. Furthermore, we evaluate our approach for hashing tasks with deep features Long et al. (2017). The statistics of the adopted datasets are summarised in Table 1.

4.1. Comparison with the state-of-the-arts

We summarise our comparison in Table 2, where the hyphen indicates the existing method has not tested on the datasets in their original publication. Our method significantly outperforms the previous published results and can achieve state-of-the-art performance comparing to most recent papers. From the confusion matrices in Fig. 2 we can see that the recognition rate to each category tends to be averaged. Such a result indicates the performance of our proposed method is stable and reliable. It is also worth noting that, due to the attributes of the two datasets are not both category-level or instance-level, all of the compared methods have to adjust the framework to fit such different settings. In comparison, our VSAR approach can deal with both of the situations.

4.2. Algorithm analysis

Effects of terms in VSAR. To understand the success of our VSAR algorithm, the first important question is how does each terms in our VSAR algorithm work for ZSL. Thus, we separately strip-down each term in Eq. 3 into three baseline models. The first model is referred as *X-to-A*, in which we remove the second term of Eq. 3 and let the visual space $\mathcal{X}$ directly map to the semantic space, i.e. $\mathcal{V} = \mathcal{A}$. This is exactly a DAP procedure that, during the test, the image is firstly mapped to the semantic space and then classified to the label space. The second model is referred as *A-to-X*. This is an interesting scenario that investigates whether we could regenerate the original visual features given just the semantic representations. Specifically, we train the model by setting $\mathcal{V} = \mathcal{X}$ and remove the first term in Eq. 3. During the test, we firstly project all attributes of the unseen categories/instances to $\mathcal{X}$. A test image is then classified in this embedding space using Eq. 7. In the third model that is denoted as *No-Graph*, we explore the importance of our dual-graph regularisations. Specifically, we train the model by setting $\lambda = 0$.

In Fig. 3. it can be seen that our full model significantly outperforms all of the baseline methods. In addition, we find the performance of the third model is roughly equal to random guess. Such a failure case matches our previous expectation that, without regularisation, Eq. 1 tend to discover the principle components rather than discriminating the categories. It is also noticeable that the *A-to-X* method gets better result on the aPY dataset than that on AwA. We ascribe this to the instance-level attributes. Such a result implies that it is feasible to generate visual features of each image from its semantic representations in future work.

Number of latent attributes. Another important issue is how many latent attributes K are required for the embedding space. Does a larger number of K always give better results? To investigate this question, we gradually increase K from 50, 85, 500, 1000, and 1000 per further step. We show the result in Fig. 3 (left). Generally speaking, a larger K tends to benefit the performance. However, we point out that there is an optimal K that gives the peak result. After that, the performance gradually degrades while we further increase K. This problem is severer on AwA than that on aPY. This is because when K goes too large, this can be viewed as an

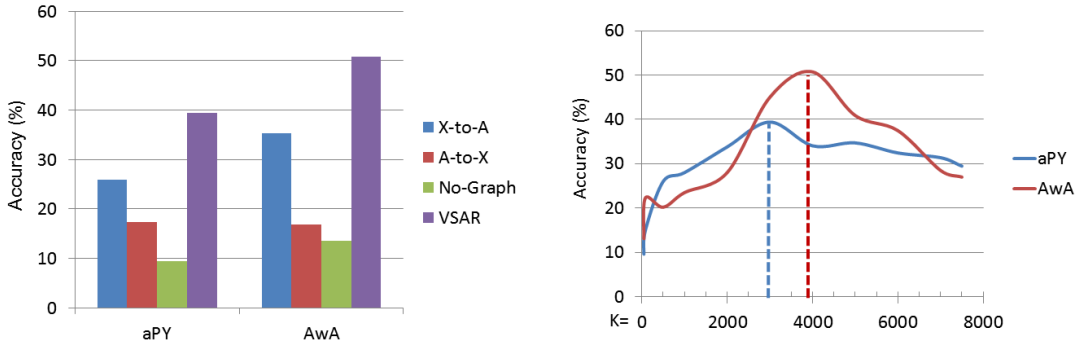

Fig. 3. Evaluating each term of the loss function in Eq. 3 (left) and the performance curve respects to the dimension K of the latent attribute space (right).

spectral over-fitting problem Zhao and Liu (2007). Since the attributes of AwA is category-level, the variance of its semantic space is much smaller than its visual space, which results in that the model on the AwA is more likely to over-fitting. For the whole experiments, we fix $K = 3000$ and 4000 for aPY and AwA, respectively.

Efficiency Our implementation is conducted in Matlab 2014a environment that is installed on a 12-core Linux system with 400G memory. The test time is done within a second. The training process takes roughly half an hour (i.e. number of batches $t = 15$) to get a converged model. Most of the time is used for solving the Eq. 6. We stress our contribution of using the batch sampling scheme, whereas directly solving the Eq. 7 without the batch sampling scheme can take up to 10 hours.

4.3. Visual-semantic ambiguity removal

In this section, we investigate what kinds of visual-semantic ambiguity are removed using our algorithm. This question can be considered from two aspects. Firstly, we consider the semantic ambiguity between different categories. On aPY dataset, we find such a semantic ambiguity problem is very severe. We use the provided "ground truth" attribute labels as the representation for each image. We then search the nearest neighbour for each image like an 1-*NN* classification. We find that only 67.17% of the nearest neighbours can match their original categories. Such a result implies that even if the conventional attribute classifiers can give perfect predictions, the overall recognition rate is only 67.17%. In Fig. 4 (Upper), we show that our VSAR is able to remove some of the semantic ambiguities. For example, in the second columns, the test image 'donkey' is misclassified as a 'bag' because the material and the logo of the bag possesses the same attributes to the donkey. However, in the visual space, such two instances are very distinctive. Therefore, using VSAR, our method successfully removes the ambiguity and gives the correct nearest neighbour. On the AwA dataset, the semantic ambiguity does not exist because all of the images in one category share the same attributes. Therefore, we consider the problem of visual ambiguity, i.e. the extracted low-level features from different categories are confused to each other. Specifically, we compare our method with the DAP framework using the *X*-to-*A* model. In Fig. 4 (Lower), we show some prediction errors in DAP can

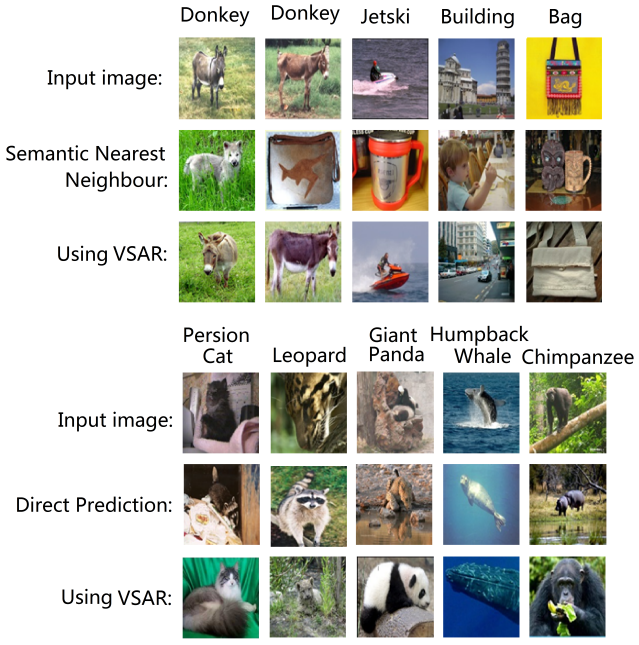

Fig. 4. Examples of successful semantic ambiguity removal on aPY (Upper) and the visual ambiguity removal on AwA (Lower).

be corrected using VSAR. Such an ability contributes to the remarkable performance improvement (39.42% to 51.75%) in Fig. 3.

4.4. Zero-shot Hashing Extensions

After confirming the effectiveness of our approach on ZSL tasks, we aim to compare it with recent state-of-the-art ZSL and hashing methods. Since there are too many aspects can affect the performance, we adopt a fair protocol propose by Xian et al. (2017). We strictly follow their experimental setting, in terms of visual features, attributes, seen/unseen splits, etc.

Our comparison is shown in Table 5. The proposed method steadily outperform the state-of-the-art results on all of the four datasets. Moreover, when we convert the representation into binary codes, we observe that the performance degradation is not significant. Without loss of generality, we use the same code length (64-bit) for all of the four datasets. As discussed above,

Table 3. Time (in sec) taken by each method to produce hash codes for images in the training dataset.

Method	AwA Dataset			SUN Dataset		
	24-bit	32-bit	40-bit	32-bit	64-bit	128-bit
tSNE-ZSH	1.271×10^{-5}	1.213×10^{-5}	1.195×10^{-5}	2.157×10^{-5}	2.184×10^{-5}	2.129×10^{-5}
NCA-ZSH	1.056×10^{-5}	1.193×10^{-5}	1.148×10^{-5}	3.680×10^{-5}	3.536×10^{-5}	3.671×10^{-5}
Isomap-ZSH	9.062×10^{-6}	9.651×10^{-6}	1.102×10^{-5}	8.476×10^{-5}	8.268×10^{-5}	8.360×10^{-5}
GPLVM-ZSH	1.334×10^{-5}	1.143×10^{-5}	1.178×10^{-5}	9.053×10^{-5}	9.297×10^{-5}	9.104×10^{-5}
Kernel PCA-ZSH	1.099×10^{-5}	1.466×10^{-5}	1.078×10^{-5}	8.452×10^{-5}	8.532×10^{-5}	8.051×10^{-5}
LLE-ZSH	1.472×10^{-5}	1.152×10^{-5}	1.288×10^{-5}	3.299×10^{-5}	3.138×10^{-5}	3.287×10^{-5}
Ours	8.273×10^{-6}	8.021×10^{-6}	9.985×10^{-6}	1.976×10^{-6}	2.032×10^{-5}	1.928×10^{-5}

Table 4. Time (in sec) taken by each method to produce hash codes for images in the test dataset.

Method	AwA Dataset			SUN Dataset		
	24-bit	32-bit	40-bit	32-bit	64-bit	128-bit
tSNE-ZSH	2.108×10^{-4}	2.118×10^{-4}	2.122×10^{-4}	7.017×10^{-4}	7.031×10^{-4}	7.019×10^{-4}
NCA-ZSH	2.196×10^{-4}	2.149×10^{-4}	2.223×10^{-4}	7.918×10^{-4}	7.682×10^{-4}	7.533×10^{-4}
Isomap-ZSH	3.299×10^{-4}	3.209×10^{-4}	3.255×10^{-4}	6.867×10^{-4}	6.857×10^{-4}	6.832×10^{-4}
GPLVM-ZSH	3.483×10^{-4}	3.418×10^{-4}	3.380×10^{-4}	8.368×10^{-4}	8.431×10^{-4}	9.284×10^{-4}
Kernel PCA-ZSH	4.178×10^{-4}	4.031×10^{-4}	3.696×10^{-4}	8.443×10^{-4}	8.273×10^{-4}	8.760×10^{-4}
LLE-ZSH	2.124×10^{-5}	2.092×10^{-5}	2.106×10^{-5}	8.698×10^{-5}	8.554×10^{-5}	8.756×10^{-5}
Ours	1.152×10^{-5}	1.193×10^{-5}	1.376×10^{-5}	3.285×10^{-5}	3.336×10^{-5}	3.927×10^{-5}

such a lower dimension is not the best for our method. As a supplementary check, we examine the performance when we use 512-dimension representation and the results are increased by 5% in average. However, due to the focus of this paper is scalable retrieval, the over long codes will increase the computational cost. Therefore, we have not include these results in this paper.

Efficiency Finally, we measure the time computed by our algorithm for the embedding and producing hash code for each instance in the training (Table 3) and test set (Table 4). We compare to six types of dimensionality reduction techniques in the metrics of Pachori and Raman (2016): Local Linear Embedding (LLE), t-Distributed Stochastic Neighbour Embedding (t-SNE), Isomap, Kernel PCA, Gaussian Process Latent Variable Model (GPLVM), and Neighbourhood componets analysis (NCA). Our method is shown more efficient than these benchmark approaches, especially at the test time.

Table 5. Zero-shot Hashing Performance (64-bit).

Method	**SUN**	**CUB**	**AWA**	**aPY**
DAP	39.9	40.0	44.1	33.8
IAP	19.4	24.0	35.9	36.6
CONSE	38.8	34.3	45.6	26.9
CMT	39.9	34.6	39.5	28.0
SSE	51.5	43.9	60.1	34.0
LATEM	55.3	49.3	55.1	35.2
ALE	58.1	54.9	59.9	39.7
DEVISE	56.5	52.0	54.2	39.8
SJE	53.7	53.9	65.6	32.9
ESZSL	54.5	53.9	58.2	38.3
SYNC	56.3	55.6	54.0	23.9
SAE	53.4	42.0	58.1	32.9
Ours (Real Numbers)	59.5	56.2	70.8	53.5
Ours (Binary Codes)	56.9	53.1	66.5	50.9

5. Conclusion and future work

We introduce that the visual-semantic ambiguity is a common issue in ZSL tasks with extensions using deep features and ZSH applications. Our results on all of the datasets support that ambiguity removal can significantly benefit the recognition performance. The proposed VSAR is an unified framework that can deal with various semantic inputs, such as category-level and instance-level attributes. Instead of regarding ZSL as a multi-label classification task, we adopt an embedding approach without struggling with the effectiveness of each attribute concept. Due to this property, our method can be simply applied to various existing intermediate semantic representations, such as data-driven attribute Yu et al. (2013) or word-vector Socher et al. (2013).

Due to our method can effectively preserve the intrinsic structure of the data, the learnt intermediate embedding is readily to be converted into generic compact representation for hashing purposes. As discussed earlier, when substitute the low-level base visual features into deep features, the method is no longer sensitive to the dimension of latent embedding. Even though we compact the codes to 64-bit, the performance loss is in an acceptable range. Moreover, our approach is shown more efficient than most of conventional manifold learning-based approaches.

References

Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C., 2013. Label-embedding for attribute-based classification, in: CVPR.

Al-Halah, Z., Gehrig, T., Stiefelhagen, R., 2014. Learning semantic attributes via a common latent space, in: VISAPP.

Cai, Z., Liu, L., Yu, M., Shao, L., 2015. Latent structure preserving hashing, in: BMVC.

Farhadi, A., Endres, I., Hoiem, D., Forsyth, D., 2009. Describing objects by their attributes, in: CVPR.

Ferrari, V., Zisserman, A., 2007. Learning visual attributes, in: NIPS.

Fu, Y., Hospedales, T.M., Xiang, T., Fu, Z., Gong, S., 2014a. Transductive multi-view embedding for zero-shot recognition and annotation, in: ECCV.

Fu, Y., Hospedales, T.M., Xiang, T., Gong, S., 2014b. Learning multimodal latent attributes. Pattern Analysis and Machine Intelligence, IEEE Transactions on 36, 303–316.

Fu, Z., Xiang, T., Kodirov, E., Gong, S., 2015. Zero-shot object recognition by semantic manifold distance, in: CVPR.

Guo, Y., Ding, G., Jin, X., Wang, J., 2016. Transductive zero-shot recognition via shared model space learning, in: AAAI.

Jayaraman, D., Grauman, K., 2014. Zero-shot recognition with unreliable attributes, in: NIPS.

Jayaraman, D., Sha, F., Grauman, K., 2014. Decorrelating semantic visual attributes by resisting the urge to share, in: CVPR.

Kankuekul, P., Kawewong, A., Tangruamsub, S., Hasegawa, O., 2012. Online incremental attribute-based zero-shot learning, in: CVPR.

Kodirov, E., Xiang, T., Fu, Z., Gong, S., 2015. Unsupervised domain adaptation for zero-shot learning, in: ICCV.

Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K., 2009. Attribute and simile classifiers for face verification, in: ICCV.

Lampert, C.H., Nickisch, H., Harmeling, S., 2009. Learning to detect unseen object classes by between-class attribute transfer, in: CVPR.

Li, X., Guo, Y., Schuurmans, D., 2015. Semi-supervised zero-shot classification with label representation learning, in: ICCV.

Liu, L., Shao, L., 2016. Sequential compact code learning for unsupervised image hashing. IEEE transactions on neural networks and learning systems 27, 2526–2536.

Liu, L., Yu, M., Shao, L., 2015. Projection bank: From high-dimensional data to medium-length binary codes, in: ICCV.

Long, Y., Liu, L., Shen, F., Shao, L., Li, X., 2017. Zero-shot learning using synthesised unseen visual data with diffusion regularisation. IEEE Transactions on Pattern Analysis and Machine Intelligence .

Long, Y., Zhu, F., Shao, L., 2016. Recognising occluded multi-view actions using local nearest neighbour embedding. Computer Vision and Image Understanding 144, 36–45.

Mahajan, D., Sellamanickam, S., Nair, V., 2011. A joint learning framework for attribute models and object descriptions, in: ICCV.

Mensink, T., Gavves, E., Snoek, C., 2014. Costa: Co-occurrence statistics for zero-shot classification, in: CVPR.

Pachori, S., Raman, S., 2016. Zero shot hashing. ACMMM .

Palatucci, M., Pomerleau, D., Hinton, G.E., Mitchell, T.M., 2009. Zero-shot learning with semantic output codes, in: NIPS.

Parikh, D., Grauman, K., 2011. Relative attributes, in: ICCV.

Romera-Paredes, B., Torr, P., 2015. An embarrassingly simple approach to zero-shot learning, in: ICML.

Shao, L., Liu, L., Yu, M., 2016. Kernelized multiview projection for robust action recognition. International Journal of Computer Vision 118, 115–129.

Socher, R., Ganjoo, M., Manning, C.D., Ng, A., 2013. Zero-shot learning through cross-modal transfer, in: NIPS.

Xian, Y., Schiele, B., Akata, Z., 2017. Zero-shot learning-the good, the bad and the ugly. CVPR .

Yu, F., Cao, L., Feris, R., Smith, J., Chang, S.F., 2013. Designing category-level attributes for discriminative visual recognition, in: CVPR.

Yu, M., Liu, L., Shao, L., 2016a. Structure-preserving binary representations for rgb-d action recognition. IEEE transactions on pattern analysis and machine intelligence 38, 1651–1664.

Yu, M., Liu, L., Shao, L., 2016b. Structure-preserving binary representations for rgb-d action recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on 38, 1651–1664.

Zhang, H., Liu, L., Long, Y., Shao, L., 2018. Unsupervised deep hashing with pseudo labels for scalable image retrieval. IEEE Transactions on Image Processing 27, 1626–1638.

Zhang, Z., Saligrama, V., 2015. Zero-shot learning via semantic similarity embedding, in: ICCV.

Zhao, Z., Liu, H., 2007. Spectral feature selection for supervised and unsupervised learning, in: ICML.