Zero Shot Learning via Multi-Scale Manifold Regularization

Shay Deutsch [1,2] Soheil Kolouri [3] Kyungnam Kim [3] Yuri Owechko [3] Stefano Soatto [1]
[1] UCLA Vision Lab, University of California, Los Angeles, CA 90095
[2] Department of Mathematics, University of California Los Angeles [3] HRL Laboratories, LLC, Malibu, CA
{shaydeu@math., soatto@cs.}ucla.edu {skolouri, kkim, yowechko,}@hrl.com

Abstract

We address zero-shot learning using a new manifold alignment framework based on a localized multi-scale transform on graphs. Our inference approach includes a smoothness criterion for a function mapping nodes on a graph (visual representation) onto a linear space (semantic representation), which we optimize using multi-scale graph wavelets. The robustness of the ensuing scheme allows us to operate with automatically generated semantic annotations, resulting in an algorithm that is entirely free of manual supervision, and yet improves the state-of-the-art as measured on benchmark datasets.

1. Introduction

Zero-shot learning (ZSL) aims to enable decisions about unseen (target) classes by assuming a shared intermediate representation learned from a disjoint set of (source) classes [17]. Early methods assumed ground truth (human annotation) was available for intermediate representations such as object attributes that can be inferred from an image. More recently, methods have emerged that automatically infer such an intermediate "semantic" representation. Among these, some have cast the problem as joint alignment of the data using graph structures [10, 5] or directly using regularized sparse representations [23, 15]. Most automatic methods, however, perform at levels insufficient to support practical applications.

We propose a new alignment algorithm based on Spectral Graph Wavelets (SGWs) [13], a multi-scale graph transform that is localized in the vertex and spectral domains. The nodes in our graph are visual features, for example activations of a convolutional neural network or any other "graph signal." Such graph signals are considered an embedding of semantic attributes in a linear space, automatically computed using Word2Vec. Learning is based on the assumption that nearby visual representations should produce similar semantic representations, which translates into a smoothness criterion for the graph signal. Our approach is performed in a transductive setting, using all unlabeled data where the classification and learning process on the transfer data is entirely unsupervised.

While our suggested approach is similar in scope to other "visual-semantic alignment" methods (see [5, 10] and references therein) it is, to the best of our knowledge, the first to use multi-scale localized representations that respect the global (non-flat) geometry of the data space and the fine-scale structure of the local semantic representation. Moreover, learning the relationship between visual features and semantic attributes is unified into a single process, whereas in most ZSL approaches it is divided into a number of independent steps [10].

Our tests show improvements on popular benchmarks such as the Animals With Attributes (AWA) dataset, demonstrating the ability of our approach to perform multi-scale manifold alignment using only automated semantic features. In addition we also demonstrate the robustness of our regularization method on the CUB dataset showing state-of-the-art results.

1.1. Related Work

Currently there are two main approaches in zero shot learning. In the first, following [16], given semantic attributes of a previously-unseen image during the test time, one would like to classify an unseen image when its attributes are given at the test time. In the second, the test attributes are unknown; however, all the test images are available at test time, and one wishes to estimate the attributes and to classify the test datum simultaneously. Among the first, [27, 24, 28, 2] suggest learning a cross-domain matching function to compare attributes and visual features. Among the second, [10] propose a multi-view transductive approach through multi-view alignment using Canonical Correlation Analysis, and the method in [15].

Several approaches for zero shot learning have been recently proposed. These include kernel alignment with unsupervised domain-invariant component analysis [11] which addresses the problem from the multi-source domain generalization point of view. It is an extension of Unsupervised

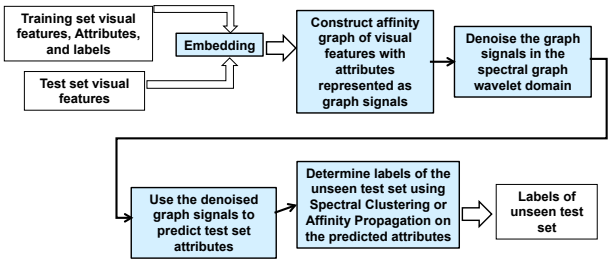

Figure 1. High level overview of our approach for zero shot learning.

Domain-Invariant Component Analysis with centered kernel alignment. Joint Latent Similarity Embedding [30] uses a dictionary-learning based discriminative learning framework to learn class-specific classifiers in both source and target domain jointly. Synthesized Classifiers for zero-shot learning [5] address the problem from the perspective of manifold learning, introducing phantom object classes which live between the low-level feature space and the semantic space, resulting in a better aligned semantic space.

An important component of ZSL is the choice of attributes. There are two types of semantic representations typically used in ZSL: (i) Human-annotated attributes, and (ii) Automatically-generated attributes.
Early methods, starting from [17], used human annotation. Automatic annotation is clearly more practical, but challenging to work with. Typically, the automatically-generated attributes are Word2Vec outputs generated using the skip-gram model trained on English Wikipedia articles [20, 4].

Denoising-based alignment methods include [23], which uses $\ell_{2,1}$ as an objective function. However, even using deep learning features, the performance of automatic methods lags that of methods using manual annotation.

ZSL is closely related to transfer learning and domain adaptation; we refer the reader to [12, 22, 9, 18] and references therein for details.

Our regularization method is inspired by [8, 7], that recently suggested a new framework for unsupervised manifold denoising which is based on SGW [13]. However, our approach is different in two ways from [8]: first, the task in our case is to align the visual and semantic representations, and therefore our graph construction is different since we use a graph attribute signal for the semantic representation, while in [8] the task is to remove noise in the manifold coordinates, which are used as the graph signal, in order to obtain a smooth manifold approximation. Second, we apply a different regularization approach, which is better suited to addressing complex manifolds which are not smooth everywhere. We employ this task by denoising all SGW bands without using thresholds, thus avoiding loss of possibly important information which is discarded in [8, 7].

1.2. Summary of Contributions

We *formalize* ZSL as the problem of learning a map from the data space X to a semantic descriptions Y (Sect. 3), after making the underlying assumptions and the resulting limitations explicit (Sec. 2).

Our first contribution is to *cast* the inference process as imposing a *differentiable structure* on the map $h : X \to Y$, supported on a discrete graph. To address this problem, we use the *multi-scale graph transform* (Sect. 3.1), which allows us to enforce global regularity without sacrificing local structure.

Our second contribution is to perform such inference in an integrated fashion (Sect. 3.2), which allows us to *forgo any manual annotation*, even in the source (training) data space. We avoid the independent steps followed by most ZSL work, some of which require supervision, to arrive an entirely automatic ZSL process.

Despite being entirely automatic, our ZSL approach *achieves state-of-the-art results* on benchmark datasets (Sect. 4.)

2. Problem Formulation and Model Assumptions

We call X the data or *visual* space (*e.g.* visual features), Y the *semantic* or attribute space (*e.g.* words in the English language), and Z_s, Z_t two disjoint class or *label* spaces (*e.g.* different sets of labels). The subscript s denotes the source (or sample, or training) set, and t denotes the target (or transfer, or test) set.

More specifically, let $\{\mathbf{x}_s^i, z_s^i\}_{i=1}^{n_s}$ be a sample, where $\mathbf{x}_s^i \in X, X \subset \mathbb{R}^m$ is measured and $z_s^i \in Z$ is an observed label in a set Z of cardinality c_s, which we indicate with an abuse of notation as $z_s^i \in \{0,1\}^{c_s}$. Also, for each instance i, let $\mathbf{y}_s^i \in Y$ be its *semantic representation*, consisting of D binary attributes $Y = \{0,1\}^D$ (for instance, $\mathbf{y}_s^i$ is the indicator vector of a list of D words describing the datum $\mathbf{x}_s^i$). It is also possible to consider $\mathbf{y}^i$ as a vector of likelihoods,[1] in which case $Y \subset \mathbb{R}_+^D$. The given *training set* consists of the sample $S = \{(\mathbf{x}_s^i, \mathbf{y}_s^i, z_s^i)\}_{i=1}^{n_s}$.

Let $\{\mathbf{x}_t^j\}_{j=1}^{n_t}$ be a *test (or transfer) set* with $\mathbf{x}_t^j \in X$, with unknown semantic attributes $\mathbf{y}_t^j$. We are interested in classifying the test set into a *different set of classes* $z_t^j \in Z_t$, where the set Z_t is of cardinality c_t, and disjoint from Z: $Z \cap Z_t = \emptyset$. We assume that this can be done with a classifier $\phi : X \to Z_t$. However, the labels in the transfer set are not given.

While one could just perform unsupervised learning on X to cluster the test set into c_t classes, ZSL breaks the problem into two: First, use the training set S to learn a map

[1] The k-th component of $\mathbf{y}_s^i$ is the likelihood of attribute k describing the datum $\mathbf{x}_s^i$: $\mathbf{y}_s^i[k] = \ell_{x_s^i}(k) = P(\mathbf{x}_s^i|k)$.

from X to Y, $h : X \to Y$. Then, use the same h to map points in the transfer set to attributes $\mathbf{y}_t^j \doteq h(\mathbf{x}_t^j)$; finally, perform unsupervised learning in Y, rather than X.

ZSL hinges on two assumptions: That the classifier for the training set $\phi : X \to Z; \mathbf{x}_s^i \mapsto z_s^i$, has the composite form $\phi(\mathbf{x}) = g(h(\mathbf{x}))$, where $g : Y \to Z$, and that the same h can be applied to both the source and target sets. One can then discard the component g of the trained classifier, and perform unsupervised learning in the set $\{\mathbf{y}_t^j = h(\mathbf{x}_t^j)\}_{j=1}^{n_t}$.

The assumptions above correspond to having a Markov chain $X \to Y \to \{Z, Z_t\}$, or equivalently that Y is a sufficient statistic of X for Z. In other words, we are assuming that, given the word representation, the image data tell us nothing about the class.

The name of the game, therefore, is to craft the class of functions so that the transferred one, h (sometimes improperly called semantic "projection" or "embedding"), is rich. Ideally, rich enough to yield a *sufficient statistic* of $\mathbf{x}$ for z_s *and* z_t. The simplest choice is for all functions to be linear, $\phi(\mathbf{x}) = W_{\mathbf{x}}\mathbf{x}$, $g(\mathbf{y}) = S\mathbf{y}$ and $h(\mathbf{x}) = \mathbf{V}\mathbf{x}$ for suitable matrices $S, \mathbf{V}$ and $W_{\mathbf{x}} = S\mathbf{V}$, as done in [24], even though the space X is typically non-linear (not a vector space). At the opposite end of the spectrum, specifying the function $h : X \to Y$ requires defining its domain X, range Y and map $X \to Y$ outside the sample set $\{\mathbf{x}_s^i\}_{i=1}^{n_s}$, all of which can be non-linear.

In our approach, we take an intermediate position, whereby we assume that the training attribute space Y is embedded in a linear space, through *Word2Vec* trained on all English Wikipedia articles using the skip-gram neural network model [20]. We then focus on X and h, with the latter learned while acknowledging the intrinsically non-linear nature of X, which is however not modified during the learning procedure. In particular, we assume that X is a smooth (but non-flat) manifold, and h is a smooth function supported on it.

In Sect. 3 we describe how these smoothness assumptions arise, and show how they can be turned into a loss function suitable for *unsupervised* learning.

3. Methodology

The criterion for learning the map h and its support space X is that nearby points in X have similar attributes in Y, and that attributes in Y are sufficient to classify both in Z_s and Z_t. Assuming a differentiable structure for X, this implies smoothness for the map h. Unfortunately, we only have the value of h on a discrete sample of X, so the goal is to find a smooth manifold X and map h supported on it, that are well "aligned" with the training data.

To this end, consider a classifier in the original sample space, ϕ, designed to minimize some loss function $l : Z_s \times Z_s \to \mathbb{R}^+$; $l(\phi(\mathbf{x}_s^i), z_s^i) = l(g(h(\mathbf{x}_s^i)), z_s^i)$. Consider the pre-image of z_s^i via g, that is the set of all semantic attributes that result in the same label: $g^{-1}(z_s^i) = \{\mathbf{y}_+ \mid g(\mathbf{y}) = z_s^i\}$, and a modified loss function $\ell : Y \times Y \to \mathbb{R}$ defined by

$$\ell(h(\mathbf{x}_s^i), g^{-1}(z_s^i)) = l(g(h(\mathbf{x}_s^i), z_s^i)\ , \forall\ i. \tag{1}$$

The goal is to find h and g that minimize the expected loss, with the expectation computed with respect to the variability in X. Since the space X is not linear, and not known, in addition to the usual regularization imposed on the maps h, g, we also need to impose regularization on the space X, and infer it along with said maps. Finally, if we are given *true attributes*, $\mathbf{y}_s^i$, we can replace them in the pre-image $g^{-1}(z_s^i)$ and solve

$$h, X = \arg\min \sum_{i=1}^{n_s} \ell(h(\mathbf{x}_s^i), \mathbf{y}_s^i) + \rho_h + \rho_X \tag{2}$$

where ρ_h and ρ_X are regularizing functionals. However, such attributes could have uncertainty or inconsistencies of their own, for instance if they are obtained by some measurement device rather than an oracle. Assuming that the training data are noisy values computed at samples around the manifold X, this problem can be naturally framed as the alignment of a smooth Y-valued function on X. Samples are represented as an attributed graph, in which the nodes represent visual data $\mathbf{x}^i$'s, with weighted edges based on their similarity, and the graph signal are the attributes $\mathbf{y}^i$'s (semantic representations). We then have a Y-valued map defined on X, which we learn from noisy samples using regularization, adapted to the manifold domain.

Graph signal processing tools [26] are well-suited to addressing this problem: Once the graph is constructed, alignment is performed by regularization applied directly in the Spectral Graph Wavelets domain [8]. This allows us to perform the alignment locally while taking into account the global properties of the space.

The next section provides a brief overview of the graph signal processing machinery needed to develop our approach in the following section.

3.1. Preliminaries

Consider a set of points $\mathbf{x} = \{\mathbf{x}_i\}\ , i = 1, ...N, \mathbf{x}_i \in \mathbb{R}^m$ which are sampled from an unknown manifold M. An undirected, weighted graph $G = (V, E)$ is constructed where V corresponds to the nodes and E to the set of edges on the graph. The adjacency matrix $\mathbf{W} = (w_{ij})$ consists of the weights $w_{i,j}$ between node i and node j. In this work, the weights are chosen using the cosine similarity between the vector observations:

$$W_{ij} = \begin{cases} \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{||\mathbf{x_i}||_2^2 ||\mathbf{x_j}||_2^2} & \text{if } \mathbf{x}_j \in \text{kNN}(\mathbf{x}_i) \\ 0 & \text{else} \end{cases} \tag{3}$$

where $\langle \mathbf{x}_i, \mathbf{x}_j \rangle = \sum_{k=1}^m x_{ik}, x_{jk}$, x_{ik} is the scalar value in the k dimension of the point $\mathbf{x}_i$.

In order to characterize the global smoothness of a function $\mathbf{f}_r \in \mathbf{R}^N$, we define its Graph Laplacian quadratic form with respect to the graph as:

$$|| \bigtriangledown \mathbf{f}_r||^2 = \sum_{V(i,j)} w_{ij}[f_r(i) - f_r(j)]^2 = \mathbf{f}_r^T \mathbf{L} \mathbf{f}_r, \quad (4)$$

where $\mathbf{f}_r$ is the graph signal which correspond to an arbitrary dimension r of the semantic representation $\mathbf{y}$, and $\mathbf{L}$ denotes the combinatorial graph Laplacian, defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, with $\mathbf{D}$ the diagonal degree matrix with entries $d_{ii} = d(i)$. The degree $d(i)$ of vertex i is defined as the sum of weights of edges that are connected to i. The eigenvalues and eigenvectors of $\mathbf{L}$ are $\lambda_1, \ldots, \lambda_N$ and $u_1, \ldots, u_N$, respectively. Note that using the notation from the previous section we have that $h(\mathbf{x_i}) = [f_1(i), f_2(i), ...f_D(i)]$. For a fixed dimension r, the graph signal of a fixed semantic representation (e.g, semantic representation of "tail") is $(h(\mathbf{x}))_r = \mathbf{f}_r$.

The Graph Fourier Transform (GFT) $\hat{f}_r$ is defined as the expansion of f_r in terms of the eigenvectors u of the Graph Laplacian $\hat{f}_r(\lambda_l) = \sum_i f_r(i) u_l(i)$. Spectral graph wavelets (SGWs) [13] define a scaling operator in the Graph Fourier domain, based on the eigenvectors of the graph Laplacian $\mathbf{L}$, which can be thought of as an analog of the Fourier transform for functions on weighted graphs. SGWs are constructed using a kernel function operator $\kappa(\mathbf{L})$ which acts on a graph signal $\mathbf{f}_r$ by modulating each graph Fourier mode $\hat{f}_r(\lambda_l)$ by $\kappa(\lambda_l)$. Scaling is defined in the spectral domain by the operator $\kappa(s\mathbf{L})$. Given a function f_r, the wavelet coefficients take the form $\Psi_{\mathbf{f}_r}(s,n) = \sum_{l=1}^N \kappa(s\lambda_l) \hat{f}_r(\lambda_l) u_l(n)$.

SGWs can be computed with a fast algorithm based on approximating the scaled generating kernels by low order polynomials. The wavelet coefficients at each scale can then be computed as a polynomial of $\mathbf{L}$ applied to the input data. When the graph is sparse, which is typically the case under the manifold learning model, the computational complexity scales linearly with the number of points, leading to a computational complexity of $O(N)$ [13]. Including a scaling function corresponding to a low pass filter operation, SGWs map an input graph signal, a vector of dimension N, to $N(J+1)$ scaling and wavelet coefficients, $c = A_{\mathbf{L_X}} \mathbf{f}_r$ which are computed efficiently using the Chebyshev polynomial approximation. The inverse wavelet transform can be estimated using a pseudo inverse transform of A, denoted as A^*.

3.2. Description of the regularization algorithm

After the graph is constructed using the proposed representation of semantic attributes as graph signals, we compute the SGW transform using low-order polynomials of the Laplacian. This way, the SGW coefficients are localized in the vertex domain, since for any two points i and n on the graph with $d_G(i,n) = K$, where d_G is the shortest distance path between two points on the graph, we have that $\mathbf{L}^K(i,n) = 0$ if $K > J$ [13]. We denote $\mathcal{N}(i,K)$ to be the set of vertex i's neighbors in the graph which are within K hops away from i. Let $\mathbf{W}_{\mathcal{N}(K)}$ and $\mathbf{L}_{\mathcal{N}(K)}$ denote the affinity matrix and its corresponding Laplacian, obtained using (3) and connecting all vertices n on the graph that are $\mathcal{N}(i,K)$ hops apart on $\mathbf{W}$. Note that for $K=1$ we have that $\mathbf{W}_{\mathcal{N}(K=1)} = \mathbf{W}$ and $\mathbf{L}_{\mathcal{N}(K=1)} = \mathbf{L}$.

We retain all scaling coefficients, which correspond to the low frequencies, and apply Tikhonov regularization directly to each of the SGW bands $\mathbf{\Psi}_{\tilde{\mathbf{f}}_r}(s(j))$, $2 \leq j \leq J$, for each of the noisy coordinate semantic representation $(h(\mathbf{x_t}))_r = \tilde{\mathbf{f}}_r$:

$$\min_{\mathbf{\Psi}_{\mathbf{f}_r}(s)} \left\{ ||\mathbf{\Psi}_{\mathbf{f}_r}(s) - \mathbf{\Psi}_{\tilde{\mathbf{f}}_r}(s)||_2^2 + \gamma \mathbf{\Psi}^T{}_{\mathbf{f}_r}(s) \mathbf{L}_{\mathcal{N}(j)} \mathbf{\Psi}_{\mathbf{f}_r}(s) \right\} \quad (5)$$

Using equality (19) in [6] and replacing the graph signal $\tilde{\mathbf{f}}_r$ with the SGW band coefficients $\mathbf{\Psi}_{\tilde{\mathbf{f}}_r}(s)$, it can be shown that the optimal solution to this problem is

$$\Psi^*{}_{\tilde{\mathbf{f}}_r}(s,n) = \sum_{l=1}^N [\frac{1}{1+\gamma\lambda_l^j}] \hat{\tilde{\Psi}}_{\tilde{\mathbf{f}}_r}(s,\lambda_l) u_l(n) \quad (6)$$

where $\hat{\tilde{\Psi}}_{\tilde{\mathbf{f}}_r}(s,\lambda_l)$ is the Graph Fourier transform of $\Psi_{\tilde{\mathbf{f}}_r}(s)$.

To calculate this solution, we use a few steps of a diffusion process on a fixed graph, by solving:

$$\mathbf{\Psi}^*{}_{\tilde{\mathbf{f}}_r}(s) = (\mathbb{I} + \gamma \mathbf{L}_{\mathcal{N}(j)})^{-1} \mathbf{\Psi}_{\tilde{\mathbf{f}}_r}(s) \quad (7)$$

Note that one step of a diffusion process on the graph is equivalent to Tikhonov regularization [14]. Thus our approach is essentially solving a diffusion process on the graph using graph signals which are SGW coefficients themselves that are localized both in the visual and semantic spaces. Note that (5) smooths out the predicted signal $(h(\mathbf{x_t}))_r$ based on the graph underlying connectivity.

After performing regularization, which provides us with the denoised $\hat{A}_{\mathbf{L_X}} \tilde{\mathbf{f}}_r$, we take the pseudo inverse transform A^* and obtain $A^* \hat{A}_{\mathbf{L_X}}(\tilde{\mathbf{f}}_\mathbf{r}) = \mathbf{f}_\mathbf{r}^*$, (Note that $(h(\mathbf{x_t}))_r = \mathbf{f}_r^*$). The proposed algorithm is applied to all semantic representation dimensions, to obtain the full regularized semantic representation $(h(\mathbf{x_t}^i)) = \mathbf{y}_i^*$ for each instance i in the testing set. Using the regularized semantic representation $(h(\mathbf{x_t}^i)) = \mathbf{y}_i^*$ for each instance i in the testing set, we perform clustering into $c_i, i = 1..c_t$ classes by globally partitioning the regularized graph (constructed from $\mathbf{y}_i^*$) using Spectral Clustering [19] or Affinity Propagation [25]).

We summarize the proposed regularization approach for zero-shot learning using Spectral Graph Wavelets in the pseudo code in Tables 1 and 2 and in the block diagram.

Algorithm 1: Alignment Algorithm

Input: The data set target unseen classes instances $\left\{X_t, \tilde{Y}_t\right\}$, k nearest neighbors on the graph, K - the order of Chebyshev polynomial approximation

Construct an undirected affinity graph $\mathbf{W}$ based on the visual features X_t using (3) and construct the Laplacian $\mathbf{L}$ from $\mathbf{W}$. ;

for $r \leftarrow 1$ **to** D **do**

- Assign the corresponding coordinate values of the semantic representation in dimension r, $\tilde{\mathbf{f}}_r = (\tilde{Y}_t)_r$, to its corresponding vertex on the graph. ;
- Calculate the SGW transform of $\tilde{\mathbf{f}}_r$, $\mathbf{\Psi}_{\tilde{\mathbf{f}}_r}(s(j))$, with $1 \leq j \leq J$;
- Perform regularization directly in the SGW domain $\mathbf{\Psi}_{\tilde{\mathbf{f}}_r}$ using Algorithm 2 . ;
- Given the regularized SGWs, $A_{\mathbf{L_x}}\hat{\mathbf{f}}_r = \left\{\mathbf{\Psi}_{\hat{\mathbf{f}}_r}(s(j))\right\}_{j=1}^{J}$, take the pseudo-inverse SGW transform A_r^* to obtain $(h(\mathbf{x_t}))_r = \mathbf{f}_r^*$, and assign $(\hat{Y}_t)_r = \mathbf{f}_r^*$.

Classify the unseen classes into $c_i, i = 1..c_t$ classes using Spectral Clustering or by using Affinity Propagation [25] .;

Output: The regularized semantic space $\hat{Y}_t$, estimated classes z_t

Algorithm 2: Regularization Algorithm

Input: Semantic representation in dimension r, $\tilde{\mathbf{f}}_r = (\tilde{Y}_t)_r$, its corresponding SGW coefficients $\mathbf{\Psi}_{\tilde{\mathbf{f}}_r}(s(j))$, Laplacian $\mathbf{L}$, γ smoothing parameter, J - number of resolutions used for wavelet decompositions

Retain the low pass scaling coefficients. For each resolution $1 \leq j \leq J$, construct $\mathbf{L}_{\mathcal{N}(j)}$. .;

for $j \leftarrow 2$ **to** J **do**

- Solve (7) with respect to $\mathbf{L}_{\mathcal{N}(j)}$, and SGW coefficients $\mathbf{\Psi}_{\tilde{f}_r}(s(j))$

Output: Regularized SGW coefficients $\mathbf{\Psi}^*_{\tilde{f}_r}(s(j))$, $1 \leq j \leq J$

4. Experimental Results

4.1. Experimental Settings

We present experimental results on the AWA (animals with attributes) dataset which is among the most widely used for ZSL. AWA consists of 50 classes of animals (30,475 images). It has a source/target split for zero-shot learning of 40 and 10 classes, respectively. We use as visual feature space X the activations of a pre-trained GoogleNet [28, 5], consistent with most current methods that use the same or other deep learning features [29]. For the semantic space Y we used Word2Vec, where each instance is represented by a 100-dimensional vector, constructed automatically from a large unlabeled text corpora [20], in the form of word vectors, that does not incur additional manual annotation. Similar to transductive approaches in zero shot learning such as [10], we begin with an initial estimation of the semantic representation of the testing data by projecting X on the semantic embedding space Y using a learned projection function from the source data with a support vector regression function [17, 10]. Note that the semantic representation of the testing data (obtained using a projection function from X to Y) are first used as graph signals which is transformed to SGW coefficients, that provides information that is localized in both X and Y, which is further aligned in the regularization process (the noisy SGW are treated as graph signals themselves). Clustering is performed in the regularized semantic embedding space, where classification accuracy is evaluated using Rand index. Note that there is no supervision. We use $J = 4$ scales for the SGW transform, and $k = 20$ for the nearest neighbor parameter for the affinity graph.

4.2. Effectiveness of Noise Suppression

We first validate our approach's ability to denoise the semantic coordinate dimension when applied to the AWA dataset. Word2Vec is typically very noisy, making manifold alignment challenging for current ZSL methods [10, 5]. For each point in the test data we compute the percentage of k nearest neighbors from the same class and report the average accuracy for all points in the test set.

Figure 2 shows the average percentage of correct k nearest neighbors from the same unseen class in the noisy Word2Vec semantic space (bright-blue) and the same after our proposed regularization (magenta) for a wide range of k nearest neighbor parameters. As can be seen, after performing alignment using our approach, the average percentage of k nearest neighbors from the same unseen class has improved significantly compared to the noisy semantic space, which indicates the effectiveness and robustness of the alignment process. Moreover, due to the multi-resolution properties of SGWs, our regularization performs well for a wide range of k nearest neighbor selections. In Figure 3 we show an illustration of our method using t-SNE embedding, which is also compared to the noisy Word2Vec. As can be seen, the noisy graph signals (Word2Vec) embedding produce semantic representation that can be very different with respect to the graph constructed from the visual features. On the other hand, using our method produces embedding results which are significantly better, showing the

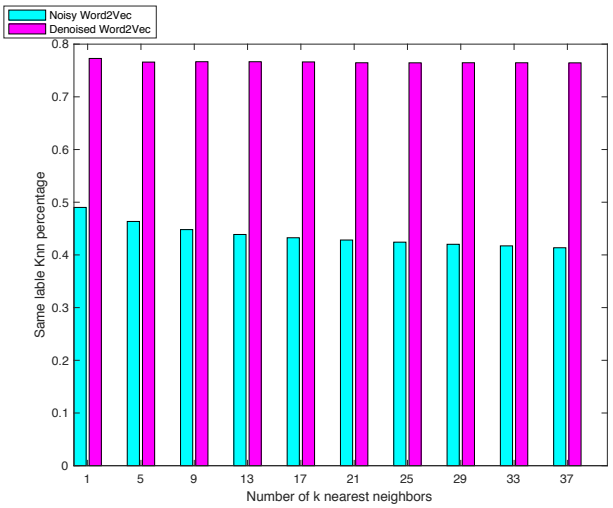

Figure 2. Average percentage of the same-class k nearest neighbors for unseen data, evaluated for $k \in \{1, 3, ..37\}$ (blue noisy semantic representation, Magenta- regularized semantic representation). It can be seen that the regularization improves the same class percentage of the k nearest neighbors for a wide range of k nearest neighbor graph parameter consistently

embedded graph signals of same instances having similar values with respect to the graph structure.

4.3. Comparison to the State of the Art

To evaluate the classification accuracy of our method, we performed Spectral Clustering [21] on the regularized semantic attributes and compared it to the original noisy semantic attributes for the AWA and CUB (see Section 4.4) datasets. As can be seen in Table 1, after performing regularization using our approach, the average percentage of k nearest neighbors which belong to the same unseen class improved significantly. In Table 2 we see a comparison to the state of the art in ZSL. The method is noted in brackets, where "H" correspond to human annotation, "W" to Word2Vec or other automated method to generate the semantic representation. It can seen that our method outperforms the state of the art, and is significantly better than all automatic methods including Transductive Multi-view Zero-Shot Learning (TMZL). We also tested our method using Affinity Propagation, which is a popular clustering method based on belief propagation that does not require specifying the number of clusters in the data in advance. Using Affinity Propagation we were able to outperform the state of the art and demonstrate the effectiveness of our method.

4.4. Comparison using the CUB dataset

In this section we compare our method on the Caltech-USCD Birds (CUB200) dataset which is another popular dataset used in zero shot learning as well as in other domains in computer vision applications. The CUB dataset contains 200 different bird classes, with 11,788 images in total. We use the same split as in [5]. with 150 classes for training and 50 disjoint classes for testing. The semantic attributes used with our method are 300 dimensional Word2Vec. In this case, the classes are close (fine-grained), and both the Word2Vec and the deep Learning features are extremely noisy, which makes the problem considerably more difficult as is evident from the low performance of nearly all ZSL methods on the CUB dataset. We compare to the state of the art methods which tested their algorithms using automated attributes.

For the CUB200 dataset, we used the method introduced in [24] (as opposed to the support vector regression used in the AWA dataset) to obtain an initial linear mapping $h : X \rightarrow Y$ from the image feature space to the semantic attribute space. This linear mapping, h, is used to provide an initial estimation for test attributes based on input image features, $\tilde{\mathbf{y}}_t = h(\mathbf{x}_t)$. Spectral Clustering is then used for clustering and Rand index for classification accuracy as in the previous Section. The experimental results, shown in Table 3 show that our manifold alignment method leads to the state of the art zero shot classification results on this dataset.

Method/Dataset	AWA	CUB
Word2Vec	36%	13 %
Regularized semantic Word2Vec	**80%**	**35%**

Table 1. Unsupervised Classification accuracy of Word2Vec before and after regularization using our method

Method/Data	AwA
DAP (A) [17]	57.5%
EZSL (A) [24]	62.85%
UDA (A)[15]	73.2%
UDA (A+ W)[15]	75.6%
Less is more (W)[23]	64.46%
Semantic Embedding (A)[29]	76.33%
LatEm (A)[28]	72.5%
LatEm (W)[28]	52.3%
SJE (W) [3]	51.2%
SJE (A.real) [3]	66.7%
TZSL (W) [10]	67%
Our approach (W), [21] based classification	**80%**
Our approach (W), AP [25] classification	**81.3%**

Table 2. Classification accuracy results using our method compared to the state of the art methods in zero shot learning on the AWA dataset. The corresponding semantic representation used is noted in brackets, where "A" corresponds to human annotated attributes and W corresponds to Word2Vec or other automated semantic representation

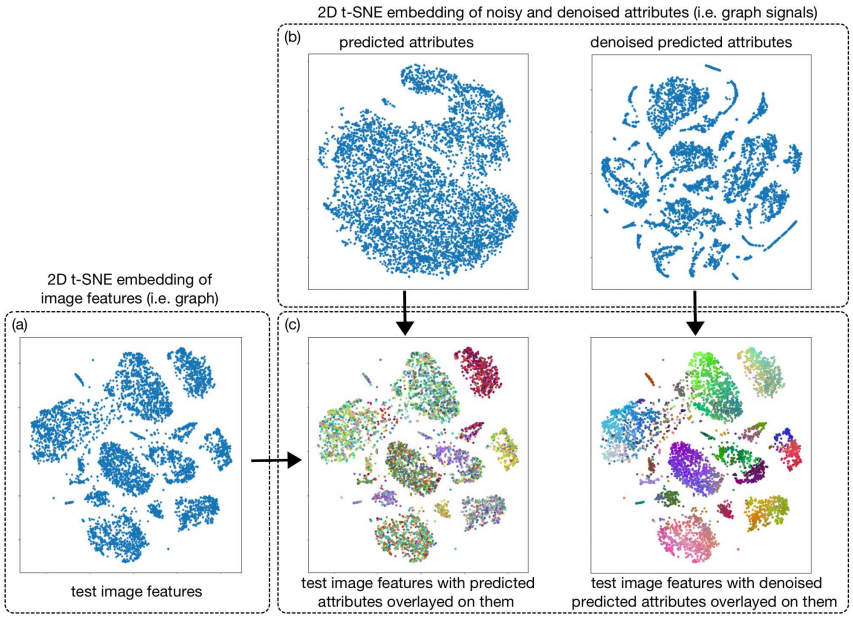

Figure 3. An Illustration of the our framework using t-SNE Embedding: on the left hand side: noise 2D embedding of the graph features. The middle and right hand side figures show the 2D embedding of the noisy and denoised graph signals, receptively. As can be seen, the embedding of the noisy graph signals result with (typically) very different values of the graph signals with respect to the intrinsic structure of the graph, while the embedding of the regularized graph signals successfully achieves the desired outcome, such that similar features have similar attributes

Method/Data	CUB
EZSL (Wikipedia) [24]	23.8%
SJW (GloVe) [3]	24.2 %
SJW (Word2Vec) [3]	28.4 %
LatEm (Word2Vec) [28]	33.1.%
LatEm (GloVe)[28]	30.7%
Less Is more (Wikipedia) [23]	29.2%
Multi-Cue Zero-Shot Learning (Word2Vec)[1]	32.1%
Our approach (Word2Vec)	**35** %

Table 3. Classification accuracy results using our method compared to the state of the art methods on the word2vec CUB dataset. The type of semantic representation used is denoted in brackets

5. Discussion

We cast the problem of zero-shot learning as the fitting ("alignment") of a smooth function h defined on a smooth (but non-flat) manifold X to sample data $(\mathbf{x}, \mathbf{y})$ where points on the visual domain X (activation functions of a convolutional network pre-trained on ImageNet) are represented as nodes on a graph, and noisy values in Y (outputs of Word2Vec) are the corresponding graph signals. Then, tools from graph signal processing are used to smooth the function $h : X \to Y$ in a way that respects the geometry of X. Once the map h is learned by smoothing, unsupervised clustering is performed in Y, thus allowing classification in a target label set. Our method is more complex than those modeling h as a linear map between linear spaces X and Y. However, the gain in accuracy is such that it allows us to operate with "noisy" samples in Y and therefore forgo all human annotation, resulting in an entirely automatic approach to zero-shot learning. Despite the lack of human annotation, our approach is competitive with the state of the art, as measured by performance in benchmarks such as the AWA and CUB200 datasets.

Acknowledgment

Research supported by ONR N00014-13-1-0563, ARO W911NF-15-1-0564/66731-CS.

References

[1] Z. Akata, M. Malinowski, M. Fritz, and B. Schiele. Multi-cue zero-shot learning with strong supervision. *CoRR*, 2016.

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[3] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *IEEE Computer Vision and Pattern Recognition*, 2015.

[4] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *European Conference on Computer Vision*, pages 663–676, 2010.

[5] S. Changpinyo, W. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. 2016.

[6] D. I Shuman, P. Vandergheynst, and P. Frossard. Chebyshev polynomial approximation for distributed signal processing. In *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems*, Barcelona, Spain, June 2011.

[7] S. Deutsch, A. Ortega, and G. Medioni. Graph-based manifold frequency analysis for denoising. *CoRR*, 2016.

[8] S. Deutsch, A. Ortega, and G. Medioni. Manifold denoising based on spectral graph wavelets. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[9] N. FarajiDavar, T. de Campos, and J. Kittler. Transductive transfer machine. In *Computer Vision - ACCV 2014 - 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part III*, pages 623–639, 2014.

[10] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Transductive multi-view zero-shot learning. *IEEE Trans. Pattern Anal. Mach. Intell.*

[11] C. Gan, T. Yang, and B. Gong. Learning attributes equals multi-source domain generalization. *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2016.

[12] R. Gopalan, R. Li, and R. Chellappa. Unsupervised adaptation across domain shift by generating intermediate data representations. 36:2288–2302, 2014.

[13] D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, pages 129–150, 2011.

[14] M. Hein and M. Maier. Manifold denoising. pages 561–568, 2007.

[15] E. Kodirov, T. Xiang, Z.-Y. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *International Conference on Computer Vision(ICCV)*, 2015.

[16] C. H. Lampert and a. S. H. Hannes Nickisch. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[17] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between class attribute transfer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[18] M. Long, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. *Advances in Neural Information Processing Systems*, 2016.

[19] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 2007.

[20] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, 2013.

[21] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2001.

[22] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: a survey of recent advances. *IEEE Signal Processing Magazine*, pages 53–69, 2015.

[23] R. Qiao, L. Liu, C. Shen, and A. van den Hengel. Less is more: zero-shot learning from online textual documents with noise suppression. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[24] B. Romera-Paredes and P. H. Torr. An embarrassingly simple approach to zero-shot learning. *Proceedings of The 32nd International Conference on Machine Learning (ICML)*, 2015.

[25] S. T. Roweis, L. K. Saul, and G. E. Hinton. Global coordination of local linear models. 2001.

[26] D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, May 2013.

[27] X. Wang and Q. Ji. A unified probabilistic approach modeling relationships between attributes and objects. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2120–2127, 2013.

[28] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[29] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. *International Conference on Computer Vision (ICCV)*, 2015.

[30] Z. Zhang and V. Saligrama. Zero-shot learning via joint latent similarity embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6034–6042, 2016.