The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)

Zero Shot Learning with the Isoperimetric Loss

Shay Deutsch,[1] Andrea Bertozzi,[1] Stefano Soatto[2]
[1]Department of Mathematics, [2]Department of Computer Science
University of California, Los Angeles
{shaydeu, bertozzi}@math.ucla.edu,[1] soatto@cs.ucla.edu[2]

Abstract

We introduce the isoperimetric loss as a regularization criterion for learning the map from a visual representation to a semantic embedding, to be used to transfer knowledge to unknown classes in a zero-shot learning setting. We use a pretrained deep neural network model as a visual representation of image data, a Word2Vec embedding of class labels, and linear maps between the visual and semantic embedding spaces. However, the spaces themselves are not linear, and we postulate the sample embedding to be populated by noisy samples near otherwise smooth manifolds. We exploit the graph structure defined by the sample points to regularize the estimates of the manifolds by inferring the graph connectivity using a generalization of the isoperimetric inequalities from Riemannian geometry to graphs. Surprisingly, this regularization alone, paired with the simplest baseline model, outperforms the state-of-the-art among fully automated methods in zero-shot learning benchmarks such as AwA and CUB. This improvement is achieved solely by learning the structure of the underlying spaces by imposing regularity.

Introduction

Motivating example. A pottopod is a pot with limbs. Not even a single example image of a pottopod is needed to find one in Fig. 1. However, one has surely seen plenty of examples of animals with limbs, as well as pots. In zero-shot learning (ZSL) one aims to exploit models trained with supervision, together with maps to some kind of attribute or "semantic" space, to then recognize objects as belonging to classes for which no previous examples have ever been seen.

The ingredients of a ZSL method are illustrated in Fig. 2. Seen samples X_s and their corresponding labels Y_s are used to train a model ϕ, typically a deep neural network (DNN), in a supervised fashion, to yield a vector in a high-dimensional (visual embedding) space Z. At the same time, a function s maps semantic attributes such as "has legs," "is short," or simply a word embedding of the ground truth (seen and unseen) labels of interest $Y = \{Y_s, Y_u\}$, to some metric space. The name of the game in ZSL is to learn a map ψ, possibly along with other components of the diagram, from the visual embedding space Z to the semantic space S, that can serve to transfer knowledge from the seen labels (reflected in Z) to the unseen ones. Alternatively, one can try to cluster samples with unseen labels in the visual embedding space Z, and then associate clusters to unseen labels.

Figure 1: Find the Pottopod (Courtesy of P. Perona).

Focus on regularization. Transfer of knowledge hinges on some kind of regularity of the maps involved in ZSL. In practice, the visual embedding space Z and the semantic embedding S are only known through discrete samples, and the maps are learned restricted to these finite samples. One crucial theme in ZSL is, interpreting sample embeddings as "noisy points" on the otherwise differentiable manifolds Z and S, to attempt to regularize the spaces Z, S, and/or the map between them.

Key contribution. Of all the various components of a ZSL method, we choose the simplest possible (Romera-Paredes and Torr 2015), except for the regularization of the semantic map. There, we introduce a sophisticated model, based on an extension of the isoperimetric inequalities from Riemannian geometry to discrete structures (graphs). We treat sample visual embeddings as vertices in a graph, with affinities as edges, and the visual-to-embedding map ψ interpreted as a linear function on the graph. We then introduce the isoperimetric loss (IPL) to enforce regularity on the domain Z based on the flow of the function defined on it through the boundary of sets of a given volume. The resulting reg-

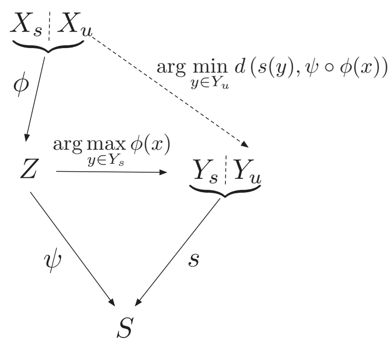


Figure 2: Illustration of the components of a ZSL algorithm: For images in the set with seen labels X_s, the labels can be estimated by maximum a-posterior over labels in the seen set Y_s on the visual representation $\phi(X_s)$. For the unseen labels, there is no direct connection to the data because they are not seen during training. Inference is then indirect: A visual representation is inferred, and from there a semantic representation, which is compared to the semantic representation of unseen labels, minimizing *some* distance in the semantic space, over all possible unseen labels Y_u.

ularized graph is informed by both the visual and semantic maps. We use it to perform clustering and map clusters to labels. Therefore, we take a very simple visual-to-semantic embedding function, namely a linear transformation, and indirectly regularize it by regularizing its domain and range spaces.

We expected our regularization to improve the baseline (Romera-Paredes and Torr 2015) on which our ZSL model is based. We did not expect it to surpass the (far more sophisticated) state-of-the-art in the two most common benchmark datasets used in ZSL, namely AwA (Lampert, Nickisch, and Harmeling 2009) and CUB (Welinder et al. 2010). Yet, the experiments indicate so. In some cases, it even outperformed methods that used human annotation for the unseen labels.

At heart, we solve a topology estimation problem. We determine the connectivity between nodes of the visual embedding graph, which defines a topology in that space informed by the semantic representation of seen attributes. Much of the literature in this area focuses on what kind of graph signal (embedding, or descriptor) to attribute to the nodes, whereas the connectivity of the graph is decided a-priori. We focus on the complementary problem, which is to determine the graph connectivity and learn the graph weights. Unlike other approaches, the connectivity in our method is informed both by the value of the visual descriptors at the vertices, and the values of the semantic descriptors in the range space. Our framework allows us to use automated semantic representation to perform ZSL, resulting in a framework which is entirely free of human annotation.

Preliminaries

We first describe a general formalization of ZSL that helps place our contribution in context. Every ZSL includes a supervised component, which results in a visual embedding, a collection of unseen labels or attributes, a map from these attributes to a vector (semantic) space, and a map from visual to semantic spaces. It is important to understand the assumptions underlying the transfer of information from seen to unseen attributes, which translates in regularity assumptions on the visual-to-semantic map.

Supervised component. In standard supervised classification, a dataset $\mathcal{D}_s$ is given where both the input data x and the output labels y_s are *seen*:

$$\mathcal{D}_s = \{x^i, y_s^i\}_{i=1}^N \tag{1}$$

where the set of seen labels, for instance 1 = "cat" and 2 = "dog," is indicated by Y_s, with cardinality $|Y_s| = n_s$. The data belong to X, for instance the set of natural images. The goal of supervised learning is to infer the parameters w of a function $\phi_w : X \to \mathbb{R}_+^K$ that approximates the (log)-posterior probability over Y_s,

$$\phi_w(x)_j \simeq \log P(y_s = j|x). \tag{2}$$

where the subscript j denotes the j-th component of the vector $\phi_w(x)$. At test time, given an unseen image x, one can infer the unknown label $\hat{y}$ associated with it as the maximum a-posteriori estimate

$$\hat{y}(x) = \arg\max_{y\in Y_s} \phi_w(x)_y. \tag{3}$$

We indicate with Z the (latent, or representation) space where the data X are mapped,

$$z^i = \phi_w(x^i) \in Z. \tag{4}$$

Visual embedding. Although z^i can be interpreted as log-probabilities, one can simply consider them as an element of a vector space of dimension at least K, $Z \subset \mathbb{R}^K$, called "visual embedding." It is also customary to use intermediate layers of a deep network, rather than the last one that is used for classification, as a visual embedding, so in general $K \neq n_s$. A general classifier, rather than a linear one, can then be used to determine the class, based on the latent representation z. We want the formalism to be flexible, so we do not constrain the dimension of the embedding to be the same of the dimension of the seen classes, and we consider $z = \phi_w(x)$ to be any (pre-trained) visual feature.

Unseen labels. In ZSL there is a second set of "unseen" labels[1] Y_u, disjoint from the first $Y_u \cap Y_s = \emptyset$. We call $Y = Y_u \cup Y_s$. At training time we do not have any sample images with labels in Y_u. However, we do at test time.

Zero-shot learning. The goal of ZSL is to classify test images as belonging to the unseen classes. That is, to learn a map from X to Y_u. Absent any assumption on how the unseen labels are related to the seen labels, ZSL makes little sense.

Assumptions. In ZSL one assumes there is a "semantic" metric vector space $S \subset \mathbb{R}^M$, to which all labels – seen and unseen – can be mapped via a function $s : Y \to S$. If the

[1] 'A misnomer, since one knows ahead of time what these labels are, for instance 3 = "sailboat", 4 = "car." However, one is not given images with those labels during training. So, while the labels are seen, image samples with those labels are not seen during training.

metric is Euclidean, a distance between two labels, y^i, y^j, can be induced via $d_s(y^i, y^j) = \|s(y^i) - s(y^j)\|$. Otherwise, any other distance $d(s(y^i), s(y^j))$ on S can be used to find the label associated to an element $\sigma \in S$ of the semantic space (embedding), for instance using a nearest-neighbor rule

$$\hat{y}(\sigma) = \arg\min_{y \in Y} d(s(y) - \sigma). \tag{5}$$

Note that the minimum could be any label, seen or unseen. This is just a metric representation of the set of labels, independent of the ZSL problem.

The second assumption is that there exists a map $\psi : Z \to S$ from the visual to the semantic embedding, which can be learned to map embeddings of seen classes z to semantic vectors $\sigma = \psi(z)$ in such a way that they land close to the semantic embedding $s(y_s)$ of the seen labels:

$$\arg\min_{\psi} \sum_{i=1}^{N} d(s(y_s^i) - \psi \circ \phi_w(x_s^i)). \tag{6}$$

One could learn both the visual embedding ϕ_w and the visual-to-semantic map ψ simultaneously, or fix the former and just learn the latter. In some cases, even the latter is fixed.

Validation. The merit of any ZSL approach is usually evaluated empirically, since the assumptions cannot be validated absent samples in the unseen label class or knowledge of the transfer between the seen and unseen tasks. Once training has terminated and we have embeddings $\phi_{\hat{w}}$ and ψ, given test data x^i, we can compare the imputed labels obtained via

$$\hat{y}(x_u) = \arg\min_{y \in Y_u} d(\psi \circ \phi_{\hat{w}}(x_u), y) \tag{7}$$

with labels y_u in the validation set. The construction of this loss function (6) is illustrated in Fig. 2.

Baseline. ZSL methods differ by whether they learn both the visual and semantic embedding, only one, or none; by the method and the criterion used for learning. Since the unseen labels are never seen during training, the transfer of information from seen to unseen labels hinges on the *regularity* of the learned maps. For this reason, much of the recent work in ZSL aims to explore different regularization methods for the learned maps. The simplest case, which requires no regularization, is to assume that all the maps are linear: $\phi(x) = Fx$ and $s(z) = Vz$ for suitable matrices F, V (Romera-Paredes and Torr 2015). The results are not state-of-the-art (see Sect.), but we nevertheless adopt this baseline and focus on regularizing, rather than the map ψ directly, the spaces Z and S, which is our key contribution.

Related Work

There are many variants for zero-shot learning (ZSL). The general formalism developed, along with the diagram in Fig. 2, helps understanding the various approaches in relation to one another.

The problem of zero-shot learning dates back to the early days of visual recognition when the desire to transfer knowledge from painfully learned model to more general classes emerged (Li, Fergus, and Perona 2006; Bart and Ullman 2005). Early modern methods consistent with our approach include (Lampert, Nickisch, and Harmeling 2009; Norouzi et al. 2013) which can be described by a choice of fixed visual embedding ϕ, semantic embedding s, and an imposed structure of the visual-semantic map (eq. (2) of (Norouzi et al. 2013) in our notation)

$$\psi(z) = \sum_{i=1}^{T} z_i s(\hat{y}(z_i))$$

where the sum is truncated at the T largest elements of z, so nothing is learned. A particularly simple approach, which we adopt as baseline, is (Romera-Paredes and Torr 2015), who assume that all the maps of interest are linear. In particular, they postulate

$$\psi(z) = Vz.$$

Although the map is linear, the domain and range where it is defined are not linear spaces (although their embedding space is). We adopt this choice and focus on smoothing the domain and range of the map.

Roughly speaking, zero shot learning methods can be classified into two main categories: inductive and transductive. In the inductive settings (Zhang and Saligrama 2016; Akata et al. 2013; Lampert and Hannes Nickisch 2014; Akata et al. 2015) which has dominated zero shot learning, the unseen classes are introduced one by one and decision about each unseen instance is made instantly once it is introduced. In the transductive setting (Li et al. 2017; Kodirov et al. 2015; Gopalan, Li, and Chellappa 2014; Changpinyo et al. 2016; Deutsch et al. 2017; Song et al. 2018) typically all unseen instances are processed simultaneously by constructing a graph where one exploits the underlying manifold structure, for example using the graph-based label propagation approach (Gopalan, Li, and Chellappa 2014). The problem of learning the graph Laplacian or the graph weights directly from given input data has been recently addressed in a number of works (Kalofolias 2016; Dong et al. 2016; Egilmez, Pavez, and Ortega 2017a; Pavez and Ortega 2016; Lake and Tenenbaum 2010). In the most general case, both the graph connectivity and the graph weights are unknown, in which case a common way to enforce smoothness is to use a regularizer which controls some level of sparsity. Perhaps the most widely used criterion is that the energy of the graph Laplacian computed from the graph signal at the vertices be small. Our approach is inspired from the isoperimetric problem, which is a classic problem in geometry. In Euclidean spaces the study of isoperimetric inequalities provides exact measures for general domains, while in Riemannian manifolds they provide some qualitative understanding of the geometry. Isoperimetric inequalities on manifolds were extended to graphs (Chung 1997; Chung, Grigor'yan, and Yau 1999) where the analysis shares some similarities and intuition from the continuous settings.

Description of our approach

We select a fixed visual embedding ϕ consisting of a ResNet101 architecture trained on ImageNet using all classes, to map images x onto a 2048-dimensional embedding $z = \phi(x)$. We assume $y_s^i \in Y_s$ is a subset of $n_s = 40$ to 150 classes depending on the dataset: In AwA (Lampert, Nickisch, and Harmeling 2009) there are 50 classes, of which we consider 40 as seen and sequester $n_u = 10$ as unseen. In CUB (Welinder et al. 2010) there are 200 classes, of which we consider 150 as seen and the rest unseen. We exploit a fixed semantic map s from text attributes, namely labels, onto a vector space $S = \mathbb{R}^M$ with dimension $M = 100$ (AwA) to 300 (CUB), using Word2Vec (Mikolov et al. 2013): The map $\psi : Z \to S$ is assumed linear, $\psi(z) = Vz$, where V is an $M \times 2048$ matrix, learned as in (Romera-Paredes and Torr 2015) using the seen dataset $\mathcal{D}_s$. To facilitate comparison to some algorithms we also use a VGGverydeep-19 on CUB rather than ResNet101.

At test time, given data x_u^i with unseen labels, we compute the visual representation $z_u^i = \phi(x_u^i) \in \mathbb{R}^K$ and then semantic embeddings $s_u^i = Vz_u^i$ for all $i = 1, \dots, N_u$. We construct a graph $\mathcal{G} = (Z, W)$ with vertices[2] $z_u^i \in Z$ and edges $\{w_{ij}\} = W$ that measure the affinities between embeddings $w_{ij} = \langle z_u^i, z_u^j \rangle$. $\mathcal{G}$ is a discrete representation of the smooth manifold $\phi(X) \subset Z$. The function ψ, restricted to $\mathcal{G}$, yields s_u^i, with range $\psi(Z) \subset S$, which we also assume to be a smooth manifold. In practice, because of the finite sampling and the nuisance variability in the descriptors, both the domain and range of ψ are far from smooth.

Key idea. Rather than smoothing the map $\psi : \mathcal{G} \to S$, we assume it is linear in the embedding space, and instead smooth both its domain and range. We seek a non-parametric deformation represented by changes in the connectivity matrix W of the underlying graph, that minimizes the isoperimetric loss (IPL). This is a form of regularization which we introduce in the field of ZSL. The IPL measures the flow through a closed neighborhood relative to the area of its boundary. For two-dimensional (2-D) surfaces in 3-D, it is minimized when the neighborhood is a sphere. The IPL extends this intuition to higher dimensions.

Application to ZSL. The result of our regularization is a new graph $\mathcal{G}'$, informed by the domain, range and map of the function ψ. We perform spectral clustering on $\mathcal{G}'$ to obtain a set of $n_u = |Y_u|$ clusters $\{c_1, \dots, c_{n_u}\}$. Each of these clusters can then be associated with a label in the unseen set Y_u. We do not need to know the association explicitly to evaluate our method. However, one could align the clusters to the semantic representation of the unknown labels if so desired. [3]

Results. In general, there is no "right" regularizer, so we validate our approach empirically on the two most common datasets for ZSL, namely AwA and CUB. Compared to the current best methods that do not use any manual annotation, Zero-IPL reduces errors by 3.06% on AwA1 (increased precision from 73.7% to 76.03%), and by 6.91% (increased precision from 36.9% to 39.45%) on CUB. Next, we describe the specific contribution, which is the smoothing of the graph-representation of ψ, in detail.

[2]We abuse the notation to indicate with Z the visual embedding space, the range of the function $\phi(X)$, which we assume to be a differentiable manifold, and the vertices of a discrete graph sampled from Z.

[3]For instance, by finding a transformation U that solves

$$\min_{U \geq 0} \sum_{y_j \in Y_u} d\left(Us(y_j), \frac{1}{|c_j|} \sum_{z_u^i \in c_j} s_u^i\right). \tag{8}$$

If so desired, one could also add to the regularization procedure a term to align the clusters to the semantic representations of the unseen labels:

$$\sum_{z_i \in c_j} d(Vz_i, Us(y_j)). \tag{9}$$

We, however, skip this as the alignment issue is beyond our focus in this paper. In practice, we use the clustering of the regularized semantic attributes and the mapping found by using the Kuhn-Munkres algorithm, similar to (Trigeorgis et al.). This does not have any impact on our method.

Regularization

In this section we describe in more detail our graph smoothing based on the isoperimetric loss (IPL).

Our baseline gives us a graph $\mathcal{G}$ with weights w_{ij} that we want to modify. We can think of these weights as "noisy," and seek a way to regularize them, by exploiting also the function ψ defined on $\mathcal{G}$ that yields semantic embeddings. Our regularization criterion is to achieve some level of compactness of bounded subsets: For a collection of subsets of the vertices with fixed size (corresponding to the volume of a subset) we want to find the subsets with the smallest size boundary. Why this might be a good criterion rests on classical differential geometry of Riemannian manifolds, where in the most basic case, the most compact manifold that encloses a fixed area with minimum size boundary is a circle. However, tools and concepts from classical differential geometry do not translate easily to graphs. Thus, we seek a technique that uses a key invariant, the isoperimetric dimension. It is transferred to the discrete setting, and we introduce the IPL as a way to control smoothness in the graph. Our criterion, quantified by the isoperimetric gap, generalizes this bias towards compactness to more general sets.

Isoperimetric loss

Let $B_r(\xi)$ be the ball around $\xi \in Z$ of radius r, that is the set of nodes within a distance $d_{\mathcal{G}}$ less than r. Let

$$\mu_i^{(\xi)} = \sum_{i \sim j,\ d_{\mathcal{G}}(j,\xi) < d_{\mathcal{G}}(i,\xi)} w_{ij} \tag{10}$$

be the flow from i towards ξ, that is, the sum of weights of edges connecting j with points closer to ξ. The geodesic flows $\mu_r^{(\xi)}$ are

$$\mu_r^{(\xi)} = \sum_{\substack{i \\ d_{\mathcal{G}}(i,\xi) = r}} \mu_i^{(\xi)}. \tag{11}$$

Note that $\mu_r^{(\xi)}$ equals $\mu(\partial B_r(\xi))$ - the sum of all the edges that connect vertices in $B_r(\xi)$ and its complement in Z, where μ is a measure on the edges in the boundary $\partial B_r(\xi)$. Next we define the isoprimetric inequality.

Definition 1 *(Chung, Grigor'yan, and Yau 1999) We say that a graph $\mathcal{G}$ has an isoperimetric dimension δ with a constant c_δ, if, for every bounded subset $B_r(\xi)$ of Z, the number of edges between $B_r(\xi)$ and the complement of $B_r(\xi)$, $Z \setminus B_r(\xi)$ satisfies*

$$\mu(\partial B_r(\xi)) \geq c_\delta(\mu(B_r(\xi)))^{1-\frac{1}{\delta}} \tag{12}$$

where $\partial B_r(\xi)$ denotes the boundary of $B_r(\xi)$.

In our notation, we have that $\partial B_r(\xi) = \mu_r^{(\xi)}$.

Next, we define the *isoperimetric gap* using the isoperimetric inequality above, which is the quantity to be minimized in the isoperiemtric loss:

Definition 2 *The* isoperimetric gap *is defined as*

$$\beta(\xi; \delta, W) \doteq c_\delta \left(\sum_{i,j \in B_r(\xi)} w_{i,j} \right)^{1-\frac{1}{\delta}} - \mu_r^{(\xi)} \tag{13}$$

To minimize the gap we propose solving the following optimization problem:

$$\begin{aligned} \min_{W \geq 0} \; & \sum_{\xi \in Z} \sum_{z_u^i, z_u^j \in B_r(\xi)} f_{s_u^i, s_u^j} w_{i,j} + \lambda \beta(\xi; \delta, W) \\ & \text{s.t. } 0 \leq w_{i,j} \leq 1 \, \forall \, i, j \end{aligned} \tag{14}$$

where $f_{s_u^i, s_u^j}$ is a function of the embedding distance between s_u^i and s_u^j (Specifically, $f : S \to \mathbb{R}$, where S is the semantic embedding, and f is the Euclidean distance; other choices are also possible) and λ is a positive scalar tuning parameter. Note that the gap β depends on δ, the isoperimetric dimension, which is unknown, and will have to be approximated.

Approximating the IPL To approximate the IPL loss, we elaborate on a spectral reduction of the isoperimetric loss, which provides a fast alternative to solving (14) directly in the vertex domain. Approximating the IPL loss using a direct method to solve (14) would entail approximating the isoperimetric dimension of the graph, which is challenging in general and even more for graphs constructed from noisy high dimensional data. Therefore, we choose to focus on a spectral reduction method.

Spectral Reduction We introduce a spectral reduction method for the isoperimetric inequalities, which reduces the isoperimteric gap directly in the spectral domain by using the vertex localization of Spectral Graph Wavelets (SGW) (Hammond, Vandergheynst, and Gribonval 2011). Specifically, we use the Spectral Graph Wavelet Transform of the semantic embedding space s_u for each of the semantic dimensions e of s_u.

Let $s_u^i(e)$ be a component of s_u^i in a fixed dimension e

$$\begin{aligned} s_u^i(e) = \sum_{j=1}^{N} s_u^j(e) \sum_{k=0}^{r} a_k \sum_{l=1}^{N} \lambda_l^k \phi_l(j) \phi_l(i) = \\ \sum_{j=1}^{N} s_u^i(e) \sum_{k=0}^{r} a_k (\mathbf{L}^k)_{i,j} \end{aligned} \tag{15}$$

Algorithm 1: Learning the Graph Connectivity Structure

Process 1: Initialization: Embedding visual - semantic domains
Input: $z_u = \{z_u^i\}_{i=1}^{N_u}$, $s_u = \{s_u^i\}_{i=1}^{N_u}$, k nearest neighbor parameter, r radius of the ball around z_u^i
Step 1: Construct k nearest neighbor graph $\mathcal{G} = (Z, W)$ from $\{z_u^i\}_{i=1}^{N_u}$ (using cosine similarity).
Step 2 Assign semantic attributes s_u^i to its corresponding node i.
Output: $\mathcal{G} = (Z, W)$
Process 2: **IPL Regularization**
Input: graph $\mathcal{G} = (Z, W)$, $s_u = \{s_u^i\}_{i=1}^{N_u}$.
Step 1: Construct the unnormalized Laplacian $\mathbf{L} = \mathbf{D} - W$ using $\mathcal{G} = (Z, W)$. Take the SGW transform (Eq. 15) for each dimension i of of s_u.
Step 2 Apply regularization using the spectral reduction of the IPL loss.
Step 3 Construct a new graph using the regularized $\hat{s}_u$.
Output: A new Semantic embedding graph $\mathcal{G}' = (S, W_s)$

a filtered signal of a fixed dimension e of s_u^i, where ϕ_l is the corresponding eigenvector of the unnormalized Laplacian $\mathbf{L}$ associated with eigenvalue λ_l. The coefficients a_k are constants of a polynomial function and for a specific choice correspond to spectral graph wavelet (SGW) coefficients. Note that the terms $\sum_{k=0}^{r} a_k(\mathbf{L}^k)_{i,j}$ can be interpreted as the localized spectral transform of the graph around the ball $B_r(z_u^i(i))$, which vanishes for all $z_u^j \notin B_r(z_u^i)$. With the SGW transform, we employ a redundant representation with r polynomials $\kappa_{t(k)}(\lambda)$, $1 \leq k \leq r$, each is approximating a kernel function localized in a frequency bands with corresponding scaling $t(k)$. Let δ_i be the impulse unit vector for the vertex i. Fixing a scale $t(k)$, the SGW coefficients $\psi_i^{t(k)}$ can be realized by $\psi_i^{t(k)} = \sum_{j=1}^{N} \sum_{k=1}^{r} a_k(\mathbf{L}^k)_{i,j}\delta_i$ and $\psi_{i,j}^{t(k)} = \sum_{k=1}^{r} a_k(\mathbf{L}^k)_{i,j}\delta_i$, where $\psi_{i,j}^{t(k)}$ is a scalar indicating the amount of diffusion propagated from vertex i to j in $B_r(z_u^i)$. Note that positive values $\psi_{i,j}^{t(k)}$ indicate that j is informative about i, where small negative or zero values indicate insignificant influence with respect to the scale $t(k)$. Next choose the smallest r_0, $1 < r_0 < r$ where we have $\psi_{i,j}^{t(r_0)} \leq 0$ for all j, with the corresponding polynomial $\kappa_{t(r_0)}(\lambda)$. Then, for all SGW coefficients $s_u^i(e)$ in bands $k \geq r_0$, we annihilate all terms $s_u^i(e)$, which has the effect of shrinking the boundaries of each ball around each vertex i and thus reducing the isoperimetric loss directly in the spectral domain. Take the inverse transform to obtain the denoised signal and construct the new graph from the regularized semantic embedding space $\hat{s}_u$.

The algorithm is summarized in pseudo code in Algorithm 1.

Clustering and validation in ZSL We employ a standard procedure for spectral clustering as follows: the input is the graph $\mathcal{G}' = (S, W_s)$ obtained from applying the IPL algorithm, and the number of clusters, n_u. Construct the Laplacian matrix $\mathbf{L}_s$ using W_s and compute the first n_u eigenvectors of $\mathbf{L}_s$. Letting $\Phi \in \mathbb{R}^{N_u \times n_u}$ correspond to the first n_u eigenvectors of $\mathbf{L}_s$ stacked in a matrix form, we cluster the vectors $\Phi_i \in \mathbb{R}^{n_u}, i = 1, ..N_u$ corresponding to the rows of Φ into n_u classes, using the k-means algorithm. Once n_u clusters are found, we can associate each of them to a different unseen label. While it is not required that the semantic embedding of the unseen labels $s(y_u)$ correspond to the clusters in the same space, mapped from the visual embeddings, this alignment can be performed *post-hoc*. For the purpose of comparison, however, it is sufficient to perform the assignment by searching over permutations of the unknown labels. Since we have at most 50 unseen labels in our experiments, this is not a bottleneck. More in general, one may consider introducing the alignment as part of the regularization, but this is beyond our scope in this paper.

Experimental Results

Experimental Settings. In the first set of experiments, we restrict our comparisons to approaches that are fully automated beyond the definition of the visual embedding (best performance is marked in boldface). In addition, we also report the evaluation of the state of the art methods that have access to embeddings of ground-truth semantic attributes.

Using our approach we choose each component of our ZSL pipeline to be the simplest possible one, corresponding to the baseline (Romera-Paredes and Torr 2015). A sanity check is whether our proposed regularization scheme improves over this baseline. Ideally, however, our method would take the baseline beyond the state-of-the-art.

To test this hypothesis, we use the two most common benchmarks for ZSL, AwA and CUB. AwA (Animals with Attributes) consists of 30,745 images of 50 classes of animals and has a source/target split of 40 and 10 classes, respectively. In addition we test on the new released dataset AWA2 which consists of 37,322 images of 50 classes which is an extension of AwA (which will be refereed from now and on AwA1). AwA2 also has source/target of 40 and 10 classes respectively with a number of 7913 unseen testing classes. We used the proposed new splits for AwA1 and AwA2 (Xian, Schiele, and Akata 2018).

The CUB dataset contains 200 different bird classes, with 11,788 images in total. We use the standard split (Changpinyo et al. 2016) with 150 classes for training and 50 disjoint classes for testing (Xian, Schiele, and Akata 2018) which is employed in most automated based methods we compare to, while (Xian, Schiele, and Akata 2018) also suggested a new split for the CUB dataset. Note that the CUB dataset is considered fine-grained, hence more challenging with both of the input features (visual and semantic) being very noisy. We present the evaluations in Tables 1, 3 and 4 using methods which are either representative or competitive for ZSL using automated attributes including (Deutsch et al. 2017; Kodirov et al. 2015; Xian et al. 2016; Frome et al. 2013; Rahman, Khan, and Porikli 2018; Changpinyo et al. 2016) as well as ones that used human annotation (Zhang and Koniusz 2018; Song et al. 2018; Zhang, Xiang, and Gong 2017; Sung et al. 2018) for a more general overview.

Method/Data	AwA1	AwA2
EZSL (Romera-Paredes and Torr 2015)	58.2	58.6
SJE (Akata et al. 2015)	65.6	61.9
ALE (Akata et al. 2016b)	59.9	62.5
LatEm (Xian et al. 2016)	50.8	-
DEVISE (Frome et al. 2013)	50.4	-
SynC (Changpinyo et al. 2016)	58.6	-
CAPD (Rahman, Khan, and Porikli 2018)	64.73	-
Kernel ZSL (Zhang and Koniusz 2018)	71.0	70.51
DEM (Zhang, Xiang, and Gong 2017)	68.4	67.1
RELATION NET (Sung et al. 2018)	68.2	64.2
QFSL (Song et al. 2018)	-	79
LisGAN (Li et al. 2019)	70.6	-
GMN (Sariyildiz and Cinbis 2019)	82.5	-
MSMR(Deutsch et al. 2017)	73.7	72
Proposed	**76.03**	**73.46**

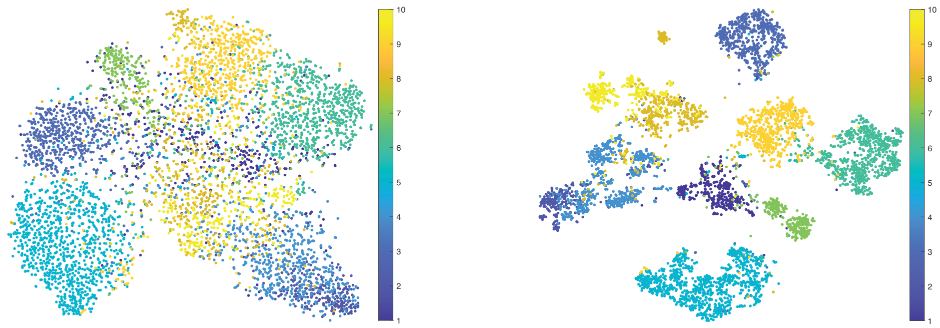
Table 1: Mean average precision accuracy (top-1 in %) results using our method compared to the state of the art in ZSL on the AwA1 and AwA2 datasets. Best performance using automated semantic representation is marked in boldface. The evaluation for the state of the art methods which are using human semantic annotation is also presented.

Method/Data	AwA1	AwA2	CUB
(Kalofolias 2016)	58.8	55.2	33.45
(Egilmez, Pavez, and Ortega 2017b)	66.6	66.7	**39.6**
Proposed	**76.03**	**73.46**	39.4

Table 2: Mean average precision accuracy (top-1 in %) results using our method compared to the state of the art graph learning methods on the AwA1, AwA2 and CUB datasets. Best performance using automated semantic representation is marked in boldface.

Implementation details: For all the splits of AwA and CUB datasets, we fix $k = 15$, $r = 3$, and $k = 8$, $r = 3$ for the k nearest neighbor graph parameter and radius r of the ball around each point, respectively. The edges w_{ij} are chosen using the cosine similarity between the visual observations.

Experimental results on the AwA1 and AwA2 datasets using the new proposed splits (Xian, Schiele, and Akata 2018) are shown in Table 1. Note that the new proposed AWA2 data-set is more challenging, as evident from the significant drop in performance compare to AwA for most of the state of the art methods. We also compare to state of the art methods which are employing human attributes (85 dimensional attribute vectors provided for each class in (Lampert, Nickisch, and Harmeling 2009)). A "-" indicates that the performance of the method was not reported in the lit-



(a) 2D t-SNE Embedding of the AWA1 noisy attributes

(b) 2D t-SNE Embedding of the AWA1 regularized attributes using the IPL regularization

Figure 3: An illustration using t-SNE embedding fot the AWA1 dataset comparing (a) noisy 2D embedding of the semantic attributes and (b) 2D embedding of the regularized semantic attributes using the proposed IPL regularization. Nodes with the same color correspond to the same class. The effect of the IPL regularization is clearly observed in (b), such that comparing to (a) the boundary of subsets, most in the same class, is typically shrinking and more compact

erature for the corresponding dataset. Mean average precision of the baseline is 58.6%. We improve it to 76.03% and 73.46% on the AwA1 and AwA2 datasets, respectively, by using our regularizer, taking the baseline past the state of the art, which is 73.7 % and 72% using (Deutsch et al. 2017) on AwA1 and AwA2 respectively, reducing the error by 3.06 percentage points on AwA1. Note that among the most competitive state of the art methods which is also using automated attributes, (Deutsch et al. 2017) is using a much more complex and computationally heavy method. Furthermore, for both AwA1 and AwA2, our method outperforms the state of the art methods which are using human attributes. Fig. 3 shows a comparison between the t-sne embedding of the noisy embedded semantic representation and the regularized semantic representation using IPL.

Experimental results on the CUB dataset is the next benchmark we consider. The baseline achieves a disappointing 23.8% precision on CUB. Surprisingly, our regularizer takes it past the state-of-the-art automatic method (36.9%), to 39.45%, corresponding to an error decrease of over 6.9%. The experimental results comparison on the CUB dataset is shown in Table 3.

Influence of the k nearest neighbor graph parameter We also provide additional experiments which test the influence of the k nearest neighbor graph parameter. Changing the k nearest neighbor graph parameter by 50% for the AWA1, AWA2, and CUB datasets, results in a performance drop of less than 2.6, 8.77, and 2.02%, respectively.

Comparison to Learning Graph Methods

In addition to direct comparison to ZSL methods, since our approach uses a graph-based smoothing approach, one may wonder whether applying state-of-the-art graph learning methods one might also improve performance of ZSL.

We focused on the state-or-the-art graph learning method (Egilmez, Pavez, and Ortega 2017b) and also (Kalofolias 2016) as representative of graph based learning. We use the same method and protocol to arrive at a graph, but replace our approach with (Egilmez, Pavez, and Ortega 2017b) (Kalofolias 2016) for evaluation in AwA1, AwA2, and CUB. This experiment is comparable to the one reported in Table 1. While performance in CUB is comparable, our approach outperforms (Egilmez, Pavez, and Ortega 2017b; Kalofolias 2016) on the AwA benchmarks.

Experimental results on generalized ZSL (GZSL) We also compare our performance in the generalized zero shot learning setting. We follow the standard protocol of the generalized ZSL (GZSL) (Schiele and Akata 2017) settings where the search space at evaluation time includes both the target and the source classes while the evaluation metric use the harmonic mean between while source and test data as the evaluation metrics. Thus, letting Acc_s, Acc_t the mean class accuracy achieved for the source and target classes, respectively, the harmonic mean H is given by:

$$H = \frac{2 * Acc_s * Acc_t}{Acc_s + Acc_t} \tag{16}$$

The settings of the GZSL is more challenging, as can be

Method/Data	CUB
EZSL (Romera-Paredes and Torr 2015)	23.8
SJE (Akata et al. 2015)	28.4
LatEm (Xian et al. 2016)	33.1
Less Is more (Qiao et al. 2016)	29.2
ALE (Akata et al. 2016b)	54.9
Kernel ZSL (Zhang and Koniusz 2018)	57.1
DEM (Zhang, Xiang, and Gong 2017)	51.7
RELATION NET (Sung et al. 2018)	55.6
QFSL (Song et al. 2018)	72.1
LisGAN (Li et al. 2019)	58.8
GMN (Sariyildiz and Cinbis 2019)	-
CAPD (Rahman, Khan, and Porikli 2018)	32.08
Multi-Cue ZSL (Akata et al. 2016a)	32.1
DMaP(Li et al. 2017)	30.34
MSMR (Deutsch et al. 2017)	36.9
Proposed	**39.45**

Table 3: Mean average precision accuracy (top-1 in %) results using our method compared to the state of the art methods in zero shot learning on the CUB dataset using Word2Vec or other automated semantic representation. Methods using automated semantic representation are marked in boldface. The evaluation for the state of the art methods which are using human semantic annotation is also presented.

seen in the evaluation comparison (Table 4, for most methods the performance degrades significantly in comparison to the standard ZSL. We compare to the recent automated methods tested in the GZSL settings on the AwA1, AwA2 and CUB datasets (those which are available on GZSL and can scale to generalized ZSL settings). The experimental results summarized in Table 4 show not only improvement over method using automatic attributes (error decrease of over 9.8% and 44% for the AWA1 and CUB datasets), but is also outperforming many of the recent state of the art methods which are using human annotation.

Computational complexity: The IPL with spectral reduction has a computational cost of $O(N_u K \log(N_u))$, which includes fast computation of the k nearest neighbor graph using $k-d$ tree, the SGW transform which is $O(N_u)$ for each dimension of the manifold for sparse graphs (Hammond, Vandergheynst, and Gribonval 2011), thus total complexity of $O(N_u K \log(N_u))$ for N_u samples in K dimensional space. Limitations includes processing very large graphs which consists more than millions of points. The execution time of our code implementation using Intel Core i7 7700 Quad-Core 3.6GHz with 64B Memory on the AWA1 dataset with 5685 points using $k = 10$ nearest neighbor graph takes $\approx$ 21.9 seconds for the initialization of the visual-semantic embedding space, and ≈ 44.8 seconds for our IPL regularization.

Method/Data	AwA1	AwA2	CUB
EZSL (Romera-Paredes and Torr 2015)	12.1	11.0	21.0
SJE (Akata et al. 2015)	19.6	14.4	33.6
LatEm (Xian et al. 2016)	13.3	20.0	24.0
ALE (Akata et al. 2016b)	27.5	23.9	34.4
DEVISE (Frome et al. 2013)	22.4	27.8	32.8
SynC (Changpinyo et al. 2016)	16.2	18.0	19.8
DMaP (Li et al. 2017)	6.44	-	2.07
CAPD (Rahman, Khan, and Porikli 2018)	43.70	-	31.6
Kernel ZSL (Zhang and Koniusz 2018)	29.8	30.8	35.1
DEM (Zhang, Xiang, and Gong 2017)	47.3	45.1	13.6
RELATION NET (Sung et al. 2018)	46.7	45.3	47.0
LisGAN (Li et al. 2019)	62.3	-	51.6
GMN (Sariyildiz and Cinbis 2019)	74.8	-	65.0
QFSL (Song et al. 2018)	-	77.4	73.2
Proposed	**48.0**	**49.2**	**45.6**

Table 4: Comparison results in generalized ZSL on the AwA1, AwA2, and CUB data-sets. The harmonic mean is measured using the Mean average precision top-1% accuracy using the unseen and seen classes. Methods using automated semantic representation are marked in boldface.

Discussion

We have introduced the use of isoperimetric inequalities, known for centuries, into clustering in general, and zero-shot learning in particular. We use the isoperimetric loss to indirectly regularize a learned map from visual representations of data to their semantic embedding. Regularization is done by representing the domain of the map as a graph, the map as a graph signal, and regularizing the graph, obtaining another "denoised" graph where clustering is performed to reveal the unseen labels, once cluster-to-label association is performed. This regularization appears to be so effective as to take the simplest possible ZSL approach, where all maps are assumed linear, and improve it to beyond the current state-of-the-art for fully automatic ZSL approaches. Typical failure modes of our regularization and clustering algorithms are when the compactness values of the geodesic flows lie within a very wide range of different intervals corresponding to the bounded sets of the different input classes, which will result in incorrect estimation and detection of the geodesic flows and therefore ineffective regularization.

Since our model is general, it could be used in conjunction with more sophisticated ZSL components, including those where the various maps are not linear, and learned jointly with regularization.

Acknowledgments

Research supported by ONR N00014-19-1-2229, N00014-19-1-2066, NSF grant DMS-1737770 and DARPA grant FA8750-18-2-0066.

References

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

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

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

Akata, Z.; Perronnin, F.; Harchaoui, Z.; and Schmid, C. 2016b. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38:1425–1438.

Bart, E., and Ullman, S. 2005. In *CVPR*, 672–679. IEEE Computer Society.

Changpinyo, S.; Chao, W.; Gong, B.; and Sha, F. 2016. Synthesized classifiers for zero-shot learning. *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chung, F.; Grigor'yan, A.; and Yau, S.-T. 1999. Higher eigenvalues and isoperimetric inequalities on riemannian manifolds and graphs.

Chung, F. R. K. 1997. *Spectral Graph Theory*. American Mathematical Society.

Deutsch, S.; Kolouri, S.; Kim, K.; Owechko, Y.; and Soatto, S. 2017. Zero shot learning via multi-scale manifold regularization. *CVPR*.

Dong, X.; Thanou, D.; Frossard, P.; and Vandergheynst, P. 2016. Learning laplacian matrix in smooth graph signal representations. *IEEE Trans. Signal Processing* 64(23):6160–6173.

Egilmez, H. E.; Pavez, E.; and Ortega, A. 2017a. Graph learning from data under laplacian and structural constraints. *J. Sel. Topics Signal Processing* 11(6):825–841.

Egilmez, H. E.; Pavez, E.; and Ortega, A. 2017b. Graph learning from data under laplacian and structural constraints. *J. Sel. Topics Signal Processing* 11(6):825–841.

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

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

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

Kalofolias, V. 2016. How to learn a graph from smooth signals. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 920–929.

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

Lake, B. M., and Tenenbaum, J. B. 2010. Discovering structure by learning sparse graph. In *Proceedings of the 33rd Annual Cognitive Science Conference*.

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

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

Li, Y.; Wang, D.; Hu, H.; Lin, Y.; and Zhuang, Y. 2017. Zero-shot recognition using dual visual-semantic mapping paths. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 5207–5215.

Li, J.; Jing, M.; Lu, K.; Ding, Z.; Zhu, L.; and Huang, Z. 2019. Leveraging the invariant side of generative zero-shot learning. In *IEEE Computer Vision and Pattern Recognition (CVPR)*.

Li, F.; Fergus, R.; and Perona, P. 2006. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.* 28(4):594–611.

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

Norouzi, M.; Mikolov, T.; Bengio, S.; Singer, Y.; Shlens, J.; Frome, A.; Corrado, G.; and Dean, J. 2013. Zero-shot learning by convex combination of semantic embeddings. *NIPS*.

Pavez, E., and Ortega, A. 2016. Generalized laplacian precision matrix estimation for graph signal processing. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, 6350–6354.

Qiao, R.; Liu, L.; Shen, C.; and van den Hengel, A. 2016. Less is more: zero-shot learning from online textual documents with noise suppression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*.

Rahman, S.; Khan, S. H.; and Porikli, F. 2018. A unified approach for conventional zero-shot, generalized zero-shot and few-shot learning. *IEEE Transactions on Image Processing*.

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

Sariyildiz, M. B., and Cinbis, R. G. 2019. Gradient matching generative networks for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schiele, Y. X. B., and Akata, Z. 2017. Zero-shot learning - the good, the bad and the ugly. *cvpr*.

Song, J.; Shen, C.; Yang, Y.; Liu, Y.; and Song, M. 2018. Transductive unbiased embedding for zero-shot learning. *CoRR* abs/1803.11320.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Trigeorgis, G.; Bousmalis, K.; Zafeiriou, S.; and Schuller, B. W. A deep matrix factorization method for learning attribute representations. *IEEE Trans. Pattern Anal. Mach. Intell.*

Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; and Perona, P. 2010. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology.

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

Xian, Y.; Schiele, B.; and Akata, Z. 2018. Zero-shot learning - the good, the bad and the ugly. *CVPR*.

Zhang, H., and Koniusz, P. 2018. Zero-shot kernel learning. *CoRR* abs/1802.01279.

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

Zhang, L.; Xiang, T.; and Gong, S. 2017. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.