Zero-Shot Kernel Learning

Hongguang Zhang[*,2,1] Piotr Koniusz[*,1,2]
[1]Data61/CSIRO, [2]Australian National University
firstname.lastname@{data61.csiro.au[1], anu.edu.au[2]}

Abstract

In this paper, we address an open problem of zero-shot learning. Its principle is based on learning a mapping that associates feature vectors extracted from i.e. images and attribute vectors that describe objects and/or scenes of interest. In turns, this allows classifying unseen object classes and/or scenes by matching feature vectors via mapping to a newly defined attribute vector describing a new class. Due to importance of such a learning task, there exist many methods that learn semantic, probabilistic, linear or piece-wise linear mappings. In contrast, we apply well-established kernel methods to learn a non-linear mapping between the feature and attribute spaces. We propose an easy learning objective inspired by the Linear Discriminant Analysis, Kernel-Target Alignment and Kernel Polarization methods [12, 8, 4] that promotes incoherence. We evaluate the performance of our algorithm on the Polynomial as well as shift-invariant Gaussian and Cauchy kernels. Despite simplicity of our approach, we obtain state-of-the-art results on several zero-shot learning datasets and benchmarks including a recent AWA2 dataset [45].

1. Introduction

The goal of zero-shot approaches is to learn a mapping that matches any given data point, say an image descriptor, to a predefined set of attributes which describe contents of that data point/image, *etc*. The hope is that such a mapping will generalize well to previously unseen combinations of attributes, therefore facilitating recognition of new classes of objects without the need for retraining the mapping itself on these new objects. The quality of zero-shot learning may depend on many factors *i.e*., (i) the mapping has to match visual traits captured by descriptors and attributes well, (ii) some visual traits and attributes describing them have to be shared between the classes of objects used for training and testing, otherwise transfer of knowledge is impossible, (iii) the mapping itself should not overfit to the training set.

[*]Both authors contributed equally.

The task of zero-shot learning can be considered as a form of transfer learning [6, 27]. Given a new (target) task to learn, the arising question is how to identify the so-called commonality [39, 21] between this task and previous (source) tasks, and transfer the knowledge from source tasks to the target task. Thus, one has to address three questions: what to transfer, how, and when [39]. For zero-shot learning, the visual traits and attributes describing them, which are shared between the training and testing sets, form this commonality. However, the objects in training and testing sets are described by disjoint classes. The role of mapping is to facilitate the identification of presence/absence of such attributes, therefore enabling the knowledge transfer. From that point of view, zero-shot learning assumes that the commonality is pre-defined (attributes typically are) and can be identified in both training and testing data. Alternatively, one can try to capture the commonality from the training and/or testing data (manually or automatically) beforehand *e.g.*, by discovering so-called *word2vec* embeddings [28, 1].

Going back to the zero-shot learning terminology, we focus in this paper on the design of mapping a.k.a. the so-called compatibility function. Various works addressing zero-shot learning and the design of mapping functions have been proposed [26, 11, 33, 30, 38, 29, 13, 2], to name but a few of approaches evaluated in [44]. We note the use of two kinds of compatibility functions: linear and non-linear.

In this paper, we recognize the gap in the trends and employ kernel methods [37] combined with an objective inspired by the Linear Discriminant Analysis (LDA) [12, 9], a related convex-concave relaxation KISSME [23], Kernel-Target Alignment [8] and Kernel Polarization [4] methods. Specifically, we are interested in training a mapping function via a non-linear kernel which 'elevates' datapoints from the Euclidean space together with attribute vectors into a non-linear high-dimensional Hilbert space in which the classifier can more easily separate datapoints that come from different classes. Our objective seeks a mapping for which all datapoints sharing the same label with attribute vectors are brought closer in the Hibert space to these attribute vectors while datapoints from different classes are pushed far apart from each other. The mapping function

takes a form of projection with soft/implicit weak incoherence mechanism related to idea [32]. Thus, our algorithm is related to subspace selection methods as our projection allows the rotation and scaling but limits the amount of shear and excludes translation. Figure 1 illustrates our approach.

For kernels, we experiment with the Polynomial family [36] and the shift-invariant Radial Basis Function (*RBF*) family including the Gaussian [36, 37] and Cauchy [5] kernels. Our choice of the Polynomial family is motivated by its simplicity while the RBF family by its ability to embed datapoints in a potentially infinite-dimensional Hilbert space. Moreover, kernels are known to impose implicitly a regularization on the plausible set of functions from which a solution to the classification problem is inferred *e.g.*, a small radius of an RBF kernel implies a highly complex decision boundary while a large radius has the opposite impact.

To our best knowledge, we are the first to combine non-linear kernels with an objective inspired by LDA [12] and kernel alignment [8] in the context of zero-shot learning.

2. Related Work

We describe first the most popular zero-shot learning methods and explain how our work differs from them.

Linear mapping a.k.a. the linear compatibility function is widely utilized in zero-shot learning. Some notable works include [13, 2, 3, 34], to name but a few of methods.

Deep Visual-Semantic Embedding (*DeViSE*) model [13] utilizes a pre-trained neural language model and a deep neural network later retrained for zero-shot learning. DeViSE uses a pairwise ranking loss inspired by the Ranking SVM [17] to learn parameters in the linear transformation layer.

Attribute Label Embedding (*ALE*) model [2] is inspired by the structured prediction approaches. The authors introduce a linear compatibility function and learn its parameters by solving a WSABIE [42] ranking objective which ensures that more importance is given to the top of the ranking list.

Structured Joint Embedding (*SJE*) framework [3] employs the linear compatibility function and the structured SVM [40] to give importance only to the top of the ranked list. For the latent representation which describes the seen and unseen classes, SJE uses either human annotated attribute vectors or latent *word2vec* embeddings [28] or global vectors *glove* [31] learned from a text corpora.

Embarrassingly Simple Zero-Shot Learning (*ESZSL*) combines a linear mapping, a simple empirical loss and regularization terms which penalize the projection of feature vectors from Euclidean into the attribute space, and the projection of attribute vectors back into the Euclidean space.

Our approach differs in that we do explicitly use a non-linear mapping function. With the use of kernel methods, we first embed datapoints into the attribute space. Then we utilize a kernel of our choice for scoring the compatibility between a given datapoint and its corresponding attribute.

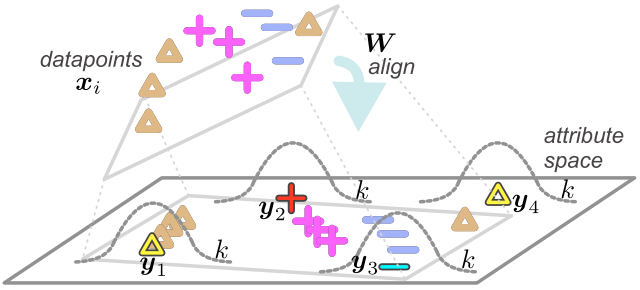

Figure 1: Zero-shot kernel alignment. Datapoints $\boldsymbol{x}$ are projected to the attribute space via the rotation and scaling matrix $\boldsymbol{W}$ which we learn. Kernels k are centered at $\boldsymbol{y}$ which are attribute vectors.

Non-linear methods have also been utilized in zero-shot learning [38, 43] and shown to improve the performance.

Cross Modal Transfer (*CMT*) approach [38] employs a two-layer neural network with a hyperbolic tangent activating function which maps images into a semantic space of words. Unlike many zero-shot learning models, CMT works well on a mixture of seen and unseen classes.

We note that our approach also handles well both seen and unseen classes simultaneously. Our non-linear mapping is obtained via a non-linear kernel [22, 19, 20] and subspace learning rather than via non-linear layers of CNN per se.

Latent Embeddings (*LatEm*) model [43] uses a piecewise linear compatibility function by combining multiple mappings learned via a pairwise ranking loss similar to the DeViSE model [13]. At the test time, the scoring function selects a mapping from the learned set which is maximally compatible with a given pair of feature and attribute vectors.

Our approach extends easily to learning multiple mapping functions such as class-specific subspaces. Moreover, our method uses a non-linear kernel that scores how compatible a given pair of feature and attribute vector is.

Semantic and probabilistic approaches to zero-shot learning include Direct and Indirect Attribute Prediction models (*DAP*) and (*IAP*) [25] which learn a probabilistic attribute classifier and predict the label by combining classifier scores. Convex Combination of Semantic Embeddings (*ConSE*) [29] maps images into a so-called semantic embedding space via convex combination of the class label embedding vectors. Semantic Similarity Embedding (*SSE*) [46] models the source and target data as a mixture of seen class histograms and uses a structured scoring function. Synthesized Classifiers (*SYNC*) [7] learn mapping between the model space and the semantic class embedding space with so-called phantom classes. The formulations and evaluations of several recent methods are provided in [45].

Classifiers. Linear Discriminant Analysis (LDA) uses two types of statistics: within- and between-class scatters computed from datapoints sharing the same label and the mean vectors of each within-class scatter, respectively. For bi-

Kernel	$k(\boldsymbol{x},\boldsymbol{y};\boldsymbol{W})$	$\mathcal{S}_{++}$ if	Shift-inv.	$\frac{\partial k(\boldsymbol{x},\boldsymbol{y};\boldsymbol{W})}{\partial \boldsymbol{W}}$
Polynomial [36]	$\left(\boldsymbol{x}^T\boldsymbol{W}\boldsymbol{y}+c\right)^r$	$c\geq 0, r\in\mathcal{I}_\infty$	no	$r\boldsymbol{x}\boldsymbol{y}^T\left(\boldsymbol{x}^T\boldsymbol{W}\boldsymbol{y}+c\right)^{r-1}$
Gaussian [37]	$\exp\left(\frac{-\|\|\boldsymbol{W}^T\boldsymbol{x}-\boldsymbol{y}\|\|_2^2}{2\sigma^2}\right)$	$\sigma>0$	yes	$-\frac{\boldsymbol{x}(\boldsymbol{W}^T\boldsymbol{x}-\boldsymbol{y})^T}{\sigma^2}\exp\left(-\frac{\|\|\boldsymbol{W}^T\boldsymbol{x}-\boldsymbol{y}\|\|_2^2}{2\sigma^2}\right)$
Cauchy [5]	$\frac{1}{1+\sigma\|\|\boldsymbol{W}^T\boldsymbol{x}-\boldsymbol{y}\|\|_2^2}$	$\sigma>0$	yes	$-\frac{2\sigma\boldsymbol{x}(\boldsymbol{W}^T\boldsymbol{x}-\boldsymbol{y})^T}{(1+\sigma\|\|\boldsymbol{W}^T\boldsymbol{x}-\boldsymbol{y}\|\|_2^2)^2}$

Table 1: Polynomial, Gaussian and Cauchy kernels $k(\boldsymbol{x},\boldsymbol{y};\boldsymbol{W})$, their properties and derivatives w.r.t. the projection matrix $\boldsymbol{W}$ introduced by us. Column $\mathcal{S}_{++}$ indicates when these kernels are positive definite. Parameters r, c and σ denote the degree, bias and radius, respectively.

nary classification, one gets a combined within-class scatter $\boldsymbol{\Sigma}=\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2$ and a between-class scatter $\boldsymbol{\Sigma}^*$. Then LDA seeks a unit length vector $\boldsymbol{w}$ to find the direction of maximum and minimum variance of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^*$, respectively: $\arg\max_{\boldsymbol{w},||\boldsymbol{w}||_2=1}\boldsymbol{w}^T(\boldsymbol{M})\boldsymbol{w}$ where $\boldsymbol{M}=\boldsymbol{\Sigma}^{-0.5}\boldsymbol{\Sigma}^*\boldsymbol{\Sigma}^{-0.5}$.

KISSME [23] for metric learning [14, 24] assumes $\boldsymbol{\Sigma}_S$ for the datapoints considered similar and $\boldsymbol{\Sigma}_D$ for the dissimilar datapoints. Given two datapoints $\boldsymbol{x}$ and $\boldsymbol{x}'$, KISSME is given by $\Delta\boldsymbol{x}^T(\boldsymbol{M})_+\Delta\boldsymbol{x}$, where $\Delta\boldsymbol{x}=\boldsymbol{x}-\boldsymbol{x}'$, $\boldsymbol{M}=\boldsymbol{\Sigma}_S^{-1}-\boldsymbol{\Sigma}_D^{-1}$ and $(\boldsymbol{M})_+$ is re-projection onto the SPD cone.

Our objective is related to LDA/KISSME; we use within- and between-class terms learnt in a non-linear setting.

3. Background

Below we review our notations and useful kernel tools.

3.1. Notations

Let $\boldsymbol{x}\in\mathbb{R}^d$ be a d-dimensional feature vector. $\mathcal{I}_N$ stands for the index set $\{1,2,...,N\}$. The Frobenius norm of matrix is given by $\|\boldsymbol{X}\|_F=\sqrt{\sum_{m,n}X_{mn}^2}$, where X_{mn} represents the (m,n)-th element of $\boldsymbol{X}$. The spaces of symmetric positive semidefinite and definite matrices are $\mathcal{S}_+^d$ and $\mathcal{S}_{++}^d$. Operator $[k_{ij}]_{i,j\in\mathcal{I}_N}$ denotes stacking coefficients k_{ij} into matrix $\boldsymbol{K}$ of size $N\times N$. Moreover, $\delta(x)=\lim_{\sigma\to 0}\exp(-x^2/(2\sigma^2))$ returns one for $x=0$ and zero for $x\neq 0$. We also define $\mathbb{1}=[1,...,1]^T$.

3.2. Kernel Alignment

Our model relies on the Kernel Alignment and Kernel Polarization methods [8, 4] detailed below.

Proposition 1. *Let $k:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ and $k':\mathbb{R}^{d'}\times\mathbb{R}^{d'}\to\mathbb{R}$ be two positive (semi-)definite kernel functions. Let two data matrices $\boldsymbol{X}\in\mathbb{R}^{d\times N}$ and $\boldsymbol{X}'\in\mathbb{R}^{d'\times N}$ contain column vectors $\boldsymbol{x}_i\in\mathbb{R}^d$ and $\boldsymbol{x}_i'\in\mathbb{R}^{d'}$ for $i\in\mathcal{I}_N$. Assume two kernel matrices $\boldsymbol{K},\boldsymbol{K}'\in\mathcal{S}_{++}^{N\times N}$ (or $\mathcal{S}_+$) for which their (i,j)-th element is given by $k(\boldsymbol{x}_i,\boldsymbol{x}_j)$ and $k'(\boldsymbol{x}_i',\boldsymbol{x}_j')$, respectively, and $i,j\in\mathcal{I}_N$. Then the empirical alignment of two kernels, which also forms a positive (semi-)definite kernel, is the quantity given by the dot-product between $\boldsymbol{K}$ and $\boldsymbol{K}'$:*

$$\langle\boldsymbol{K},\boldsymbol{K}'\rangle=\sum_{i,j\in\mathcal{I}_N}k(\boldsymbol{x}_i,\boldsymbol{x}_j)k'(\boldsymbol{x}_i',\boldsymbol{x}_j'). \tag{1}$$

Proof. Mercer theorem [36] states that for any $c_i,c_j\in\mathbb{R}$, the ineq. $\sum_{i,j\in\mathcal{I}_N}c_ic_jk(\boldsymbol{x}_i,\boldsymbol{x}_j)k'(\boldsymbol{x}_i',\boldsymbol{x}_j')\geq 0$ must hold for a positive (semi-)definite kernel kk':

$$\begin{aligned}&\textstyle\sum_{i,j\in\mathcal{I}_N}c_ic_jk(\boldsymbol{x}_i,\boldsymbol{x}_j)k'(\boldsymbol{x}_i',\boldsymbol{x}_j')=\\&\textstyle\sum_{i,j\in\mathcal{I}_N}c_ic_j\left\langle\boldsymbol{\phi}(\boldsymbol{x}_i),\boldsymbol{\phi}(\boldsymbol{x}_j)\right\rangle\left\langle\boldsymbol{\psi}(\boldsymbol{x}_i'),\boldsymbol{\psi}(\boldsymbol{x}_j')\right\rangle=\\&\textstyle\left\langle\sum_{i\in\mathcal{I}_N}c_i\boldsymbol{\phi}(\boldsymbol{x}_i)\boldsymbol{\psi}(\boldsymbol{x}_i')^T,\sum_{j\in\mathcal{I}_N}c_j\boldsymbol{\phi}(\boldsymbol{x}_j)\boldsymbol{\psi}(\boldsymbol{x}_j')^T\right\rangle=\\&\textstyle||\sum_{i\in\mathcal{I}_N}c_i\boldsymbol{\phi}(\boldsymbol{x}_i)\boldsymbol{\psi}(\boldsymbol{x}_i')^T||_{\mathcal{H}}^2\geq 0,\end{aligned} \tag{2}$$

where $\boldsymbol{\phi}(\boldsymbol{x})\in\mathbb{R}^{\hat{d}}$ and $\boldsymbol{\psi}(\boldsymbol{x}')\in\mathbb{R}^{\hat{d}'}$ are so-called feature maps [36] for kernels k and k'. Such maps always exist for positive (semi-)definite kernels by definition [36]. □

Remark 1. *The empirical alignment $\langle\boldsymbol{K},\boldsymbol{K}'\rangle$ can also be evaluated between rectangular matrices $\boldsymbol{K},\boldsymbol{K}'\in\mathbb{R}^{M\times N}$.*

Proposition 2. *Let $k:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ be a kernel function which is positive (semi-)definite. Assume that data and kernel matrices $\boldsymbol{X}\in\mathbb{R}^{d\times N}$ and $\boldsymbol{K}\in\mathcal{S}_{++}^{N\times N}$ (or $\mathcal{S}_+$) are formed as in Prop. 1. Moreover, let each column vector $\boldsymbol{x}_i$ of $\boldsymbol{X}$ have a corresponding label l_i for $i\in\mathcal{I}_N$, so that $\boldsymbol{X}$ has a corresponding label vector $\boldsymbol{l}\in\{-1,1\}^N$. By construction, $\boldsymbol{L}=\boldsymbol{l}\boldsymbol{l}^T\in\mathcal{S}_+^{N\times N}$ is a rank-1 kernel. Then the empirical alignment of kernels $\boldsymbol{X}$ and $\boldsymbol{L}$, which also forms a positive semidefinite kernel, forms a so-called kernel polarization:*

$$\langle\boldsymbol{K},\boldsymbol{L}\rangle=\sum_{i,j\in\mathcal{I}_N}l_il_jk(\boldsymbol{x}_i,\boldsymbol{x}_j)=\boldsymbol{l}^T\boldsymbol{K}\boldsymbol{l}=\sum_{(i,j):\,l_i=l_j}k(\boldsymbol{x}_i,\boldsymbol{x}_j)-\sum_{(i,j):\,l_i\neq l_j}k(\boldsymbol{x}_i,\boldsymbol{x}_j) \tag{3}$$

Proof. It follows the same steps as for Proposition 1. Moreover, $\langle\boldsymbol{K},\boldsymbol{L}\rangle$ itself forms a positive semidefinite kernel as $\boldsymbol{K}$ is positive (semi-)definite and $\boldsymbol{L}$ is positive semidefinite (*e.g.*, rank-1) by design. □

Proposition 3. *Assume matrix $\boldsymbol{W}\in\mathbb{R}^{d\times d}$ which column vectors $\boldsymbol{w}_1,...,\boldsymbol{w}_d$ are orthogonal and a projection of datapoint matrix $\boldsymbol{X}\in\mathbb{R}^{d\times N}$ into the attribute space, that is, $\boldsymbol{Y}=\boldsymbol{W}^T\boldsymbol{X}\in\mathbb{R}^{d\times N}$. Then the inverse proj. is $\boldsymbol{X}=\boldsymbol{W}\boldsymbol{Y}$.*

Proof. It follows from the orthogonality of column vectors $\boldsymbol{w}_1,...,\boldsymbol{w}_d$ of $\boldsymbol{W}$ that $\boldsymbol{W}^T\boldsymbol{W}=\mathrm{diag}(s_1,...,s_d)$, where $\mathrm{diag}(\boldsymbol{s})$ is a diagonal matrix with $s_1,...,s_d$ on its diagonal. □

3.3. Kernel Choices

Table 1 details kernels used in this paper, their parameters and properties such as the shift-invariance and positive definiteness. We also list derivatives w.r.t. the projection matrix $\boldsymbol{W}$ introduced by us to map datapoints to the space of attribute vectors. We detail this in Section 4.

4. Problem Formulation

In this section, we detail our zero-shot kernel learning. First, we explain our notations. Let us define data matrices $\boldsymbol{X}\in\mathbb{R}^{d\times N}$ and $\boldsymbol{Y}\in\mathbb{R}^{d'\times N}$ which contain N datapoints and attribute vectors as column vectors $\boldsymbol{x}_i\in\mathbb{R}^d$ and $\boldsymbol{y}_i\in\mathbb{R}^{d'}$ for $i\in\mathcal{I}_N$, respectively. The number of datapoints per class $c\in\mathcal{I}_C$ is N_c. A vector $\boldsymbol{l}\in\mathcal{I}_C^N$ contains corresponding class labels, one per datapoint/image, *etc*. In our case, datapoints and attribute vectors, which constitute an input to our algorithm, are taken from the standard zero-shot learning package [44]. As the available attribute vectors are one per class, we replicate each for all datapoints of a given class. Figure 1 illustrates our approach. Below we first analyze our proposed weak incoherence mechanism.

Proposition 4. *Let us assume some loss ℓ which we minimize w.r.t. projection $\boldsymbol{W}$, and $\boldsymbol{X}$ and $\boldsymbol{Y}$ are defined as in Section 4. Then the following expression promotes weak incoherence between column vectors of $\boldsymbol{W}$:*

$$\arg\min_{\boldsymbol{W}}\ \ell\left(||\boldsymbol{W}^T\boldsymbol{X}-\boldsymbol{Y}||_2^2\right)+\ell\left(||\boldsymbol{W}\boldsymbol{Y}-\boldsymbol{X}||_2^2\right). \quad (4)$$

Proof. For brevity, we drop loss ℓ and consider terms $||\boldsymbol{W}^T\boldsymbol{X}-\boldsymbol{Y}||_2^2$ and $||\boldsymbol{W}\boldsymbol{Y}-\boldsymbol{X}||_2^2$. Clearly, Eq. (4) will yield approximation error matrices $\Delta\boldsymbol{Y}$ and $\Delta\boldsymbol{X}$ *e.g.*, $\boldsymbol{W}^T\boldsymbol{X}=\boldsymbol{Y}+\Delta\boldsymbol{Y}$ and $\boldsymbol{W}\boldsymbol{Y}=\boldsymbol{X}+\Delta\boldsymbol{X}$. Combining both, we obtain:

$$(\boldsymbol{W}^T\boldsymbol{W}-\mathbb{I})\boldsymbol{Y}=\boldsymbol{W}^T\Delta\boldsymbol{X}+\Delta\boldsymbol{Y}=\boldsymbol{\xi}, \quad (5)$$

where $\boldsymbol{\xi}$ is the total approximation error matrix. From Eq. (5) it follows that if $||\boldsymbol{\xi}||_F^2\rightarrow 0$, then $\boldsymbol{W}^T\boldsymbol{W}\rightarrow\mathbb{I}$. This means the lower the approximation error is, the higher the incoherence becomes. Moreover, if $||\boldsymbol{\xi}||_F^2=0$, then columns of $\boldsymbol{W}$ are orthogonal w.r.t. each other. Proposition 3 is a special case of Proposition 4. □

4.1. Zero-Shot Kernel Alignment

For shift-invariant kernels, that is kernels which can be written as $k(\boldsymbol{x}-\boldsymbol{x}',0)$, we maximize the following objective which performs the kernel alignment for zero-shot learning:

$$\boldsymbol{W}^*=\arg\max_{\boldsymbol{W}}\left\langle\boldsymbol{K}_\sigma(\boldsymbol{W})+\boldsymbol{K}'_\sigma(\boldsymbol{W}),\boldsymbol{L}_\lambda\right\rangle. \quad (6)$$

$\boldsymbol{K}_\sigma(\boldsymbol{W})\equiv\left[k_\sigma(\boldsymbol{W}^T\boldsymbol{x}_i,\boldsymbol{y}_j)\right]_{i,j\in\mathcal{I}_N}$ and $\boldsymbol{K}'_\sigma(\boldsymbol{W})\equiv\left[k_\sigma(\boldsymbol{x}_i,\boldsymbol{W}\boldsymbol{y}_j)\right]_{i,j\in\mathcal{I}_N}$ denote RBF kernels *e.g.*, Gaussian or Cauchy with radius σ. Note that kernels $\boldsymbol{K}_\sigma$ and $\boldsymbol{K}'_\sigma$ use projections $\boldsymbol{W}^T\boldsymbol{x}$ and $\boldsymbol{W}\boldsymbol{y}$, respectively, which follow Prop. 4. This implies that Eq. (6) is encouraged to find a solution $\boldsymbol{W}^*$ for which its column vectors $\boldsymbol{w}_1,...,\boldsymbol{w}_{d'}$ are weakly incoherent/closer to being orthogonal w.r.t. each other. Therefore, our projection matrix $\boldsymbol{W}$ is constrained in a soft/implicit manner to be well-regularized. Such a constrained $\boldsymbol{W}$ is closer to being a subspace than an unconstrained $\boldsymbol{W}$. As such, we can rotate and scale datapoints to project them into the attribute space. Excluding the shear prevents overfitting *e.g.*, it acts implicitly as a regularization[3].

We apply the polarization mechanism detailed in Prop. 2 which, in some non-linear Hilbert space, will bring closer/push apart all class-related/unrelated datapoints and attribute vectors, respectively. This is also similar in spirit to LDA and KISSME. We define $\boldsymbol{L}_\lambda$ which encodes labels for our polarization inspired zero-shot kernel learning as:

$$\boldsymbol{L}_\lambda\equiv\left[\delta(l_i-l_j)-\lambda(1-\delta(l_i-l_j))\right]_{i,j\in\mathcal{I}_N}, \quad (7)$$

where l_i is the i-th coefficient of $\boldsymbol{l}$. By sorting all labels, it can be easily verified that $[\delta(l_i-l_j)]_{i,j\in\mathcal{I}_N}$ equals one when $l_i=l_j$ and that this term contributes an equivalent of the block-diagonal entries in $\boldsymbol{L}$. Moreover, it moves within-class datapoints close to each other. In contrast, $\lambda\left[(1-\delta(l_i-l_j))\right]_{i,j\in\mathcal{I}_N}$ contributes -λ off-diagonally and its role is to move between-class datapoints away from each other. Thus, λ controls a form of regularization. It balances positive and negative entries. Depending on labels $\boldsymbol{l}$ and λ, kernel $\boldsymbol{L}$ may be positive or negative (semi-)definite[1].

Complexity. For Eq. (6), we obtain $\mathcal{O}(Ndd'+N^2d')$ complexity which reduces to $\mathcal{O}(Ndd'+NCd')$ if one attribute vector per class is defined or multiple attribute vectors per class are defined but only one is drawn per iteration of SGD as detailed below.

4.2. Practical Implementation

Below we show practical expansions of Eq. (6) for the RBF kernels from Table 1, which demonstrate the simplicity of our approach:

$$\begin{aligned}\boldsymbol{W}^*&=\arg\max_{\boldsymbol{W}}\ \left\langle\boldsymbol{K}_\sigma(\boldsymbol{W})+\boldsymbol{K}'_\sigma(\boldsymbol{W}),\boldsymbol{L}_\lambda\right\rangle= \quad (8)\\ &=\sum_{(i,j):\,l_i=l_j}k_\sigma(\boldsymbol{W}^T\boldsymbol{x}_i,\boldsymbol{y}_j)+k_\sigma(\boldsymbol{x}_i,\boldsymbol{W}\boldsymbol{y}_j)\ -\lambda\sum_{(i,j):\,l_i\neq l_j}k_\sigma(\boldsymbol{W}^T\boldsymbol{x}_i,\boldsymbol{y}_j)+k_\sigma(\boldsymbol{x}_i,\boldsymbol{W}\boldsymbol{y}_j)\\ &=\sum_{i\in\mathcal{I}_N}N_{l_i}\left(k_\sigma(\boldsymbol{W}^T\boldsymbol{x}_i,\boldsymbol{y}_i)+k_\sigma(\boldsymbol{x}_i,\boldsymbol{W}\boldsymbol{y}_i)\right)\ -\lambda\sum_{(i,j):\,l_i\neq l_j}k_\sigma(\boldsymbol{W}^T\boldsymbol{x}_i,\boldsymbol{y}_j)+k_\sigma(\boldsymbol{x}_i,\boldsymbol{W}\boldsymbol{y}_j).\end{aligned}$$

We simplify our problem in Eq. (8) to work with SGD:

$$\begin{aligned}f_i(\boldsymbol{W})=N_{l_i}\left(k'_\sigma(\boldsymbol{W}^T\boldsymbol{x}_i,\boldsymbol{y}_i)+k'_\sigma(\boldsymbol{x}_i,\boldsymbol{W}\boldsymbol{y}_i)\right) \quad (9)\\ +\lambda\sum_{j\in\text{Rnd}(\mathcal{I}_C\setminus\{l_i\})}k''_\sigma(\boldsymbol{W}^T\boldsymbol{x}_i,\boldsymbol{y}_j)+k''_\sigma(\boldsymbol{x}_i,\boldsymbol{W}\boldsymbol{y}_j),\end{aligned}$$

[1] Compare *e.g.*, $\boldsymbol{L}_{0.2}=\mathbb{I}-0.2\mathbb{1}\mathbb{1}^T\in\mathcal{S}_+^{5\times5}$ vs. $\boldsymbol{L}_1=\mathbb{I}-\mathbb{1}\mathbb{1}^T\in\mathbb{R}^{5\times5}$.
[2] If there is one attribute vector per class, the choice of index is fixed.
[3] We also exclude translation as we mean-center our data/attr. vectors.

where $\boldsymbol{W}_{t+1} := \boldsymbol{W}_t - \frac{\beta_t}{I}\sum_{i\in B_t}\frac{1}{\sqrt{\boldsymbol{A}_t}}\frac{\partial f_i(\boldsymbol{W})}{\partial \boldsymbol{W}}$ is an SGD-based update for $\boldsymbol{W}$, B_t are mini-batches of size I, β_t is a decaying learning rate, operator $\text{Rnd}(\mathcal{I}_C \setminus \{l_i\})$ selects one index $j \in \mathcal{I}_N$ per class $c \in \mathcal{I}_C \setminus \{l_i\}$ *e.g.*, if $C = 40$, we get 39 indexes[2], each one randomly sampled per class for all 39 classes. This way, we are able to reduce the complexity as detailed earlier. Setting $k'_\sigma = (1-k_\sigma)^2$ and $k''_\sigma = k^2_\sigma$ instead of $k'_\sigma = -k_\sigma$ and $k''_\sigma = k_\sigma$, respectively, results in a slightly faster convergence of our algorithm. Also, we set $N_{l_i} = \frac{N}{C}$ for simplicity. Moreover, $\boldsymbol{A}_t = \gamma\boldsymbol{A}_{t-1} + (1-\gamma)\frac{1}{I}\sum_{i\in B_t}\left(\frac{\partial f_i(\boldsymbol{W})}{\partial \boldsymbol{W}}\right)^2$ defines the so-called moving average of the squared gradient used in the Root Mean Square Propagation (RMSprop) [16] solver.

No incoherence. Equations for RBF kernels with no soft/implicit incoherence on $\boldsymbol{W}$ can be easily derived from Prop. 2 by maximizing $\langle \boldsymbol{K}_\sigma(\boldsymbol{W}), \boldsymbol{L}_\lambda\rangle$ w.r.t. $\boldsymbol{W}$. This yields a solution similar to Eq. (9):

$$f_i(\boldsymbol{W}) = N_{l_i} k'_\sigma(\boldsymbol{W}^T\boldsymbol{x}_i, \boldsymbol{y}_j) + \lambda \sum_{j\in \text{Rnd}(\mathcal{I}_C\setminus\{l_i\})} k''_\sigma(\boldsymbol{W}^T\boldsymbol{x}_i, \boldsymbol{y}_j). \tag{10}$$

Polynomial kernel. As $K + K' = 2K$ for Polynomial kernels, the objective in Eq. (6) cannot implicitly impose weak incoherence constraints on column vectors of $\boldsymbol{W}$. Thus, we add a soft penalty on $\boldsymbol{W}$ to promote incoherence, adjusted via variable α, and we define a modified problem:

$$\boldsymbol{W}^* = \arg\max_{\boldsymbol{W}} \langle \boldsymbol{K}_\sigma(\boldsymbol{W}), \boldsymbol{L}_\lambda\rangle - \alpha||\boldsymbol{W}^T\boldsymbol{W}||^2_F + \alpha\,\text{Tr}(\boldsymbol{W}^T\boldsymbol{W}). \tag{11}$$

4.3. Classification

Having learned $\boldsymbol{W}^*$, at the classification stage we apply a simple maximization over testing attribute vectors:

$$\arg\max_{j\in\mathcal{I}_{N^*}: l^*_j\in\mathcal{I}_{C^*}} k\left(\boldsymbol{W}^{*T}\boldsymbol{x}^*, \boldsymbol{y}^*_j\right), \tag{12}$$

where k is a kernel used during training, $\boldsymbol{x}^*$ is a testing datapoint. Moreover, $\boldsymbol{y}^*$ are typically previously unseen testing attribute vectors while $\boldsymbol{l}^*$ contains testing labels (typically disjoint with $\boldsymbol{l}$). Variable C^* is the number of testing classes and N^* is the number of testing attribute vectors (typically $N^* = C^*$). For the problems which require the Nearest Neighbor classifier with a dot-product based similarity measure, one can apply *e.g.* the Nyström approximation [36] which linearizes kernel k via feature maps $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^{\hat{d}}$:

$$k(\boldsymbol{W}^{*T}\boldsymbol{x}^*, \boldsymbol{y}^*_j) \approx \left\langle \boldsymbol{\phi}(\boldsymbol{W}^{*T}\boldsymbol{x}^*), \boldsymbol{\phi}(\boldsymbol{y}^*_j)\right\rangle. \tag{13}$$

This is however outside of the scope of our evaluations and will be explored in our future work.

Figure 2: The top and bottom row include example images from the SUN and CUB datasets, respectively. The first two and the last two columns show examples of training and testing images.

5. Experiments

In what follows, we explain our experimental setup followed by evaluations of the proposed zero-shot kernel learning approach. Subsequently, we discuss our findings.

Datasets. We use five datasets frequently applied in evaluations of zero-shot learning algorithms. Attribute Pascal and Yahoo (*aPY*) [11] is a small-scale dataset which contains 15339 images, 64 attributes and 32 classes. The 20 classes known from Pascal VOC [10] are used for training and 12 classes collected from Yahoo! [11] are used for testing. Animals with Attributes (*AWA1*) [25] contains 30475 images of 50 classes. It has a standard zero-shot learning split with 40 training classes and 10 test classes. Each class is annotated with 85 attributes. At present, the original images of AWA1 are not available due to copyrights, therefore, a new version of Animals with Attributes (*AWA2*) was proposed in work [45]. AWA2 also has 40 classes for training and 10 classes for testing. Caltech-UCSD-Birds 200-2011 (*CUB*) [41] has 11788 images, 200 classes and 312 attributes to describe each class. The SUN dataset (*SUN*) [30] consists of 14340 images from 717 classes which are annotated with 102 attributes. Figure 2 shows example images from datasets.

We note that a recent evaluation paper [44] reports that some of the testing classes from the standard zero-shot learning datasets overlap with the classes of ImageNet [35] which is typically used for fine-tuning image embeddings. This results in a biased evaluation protocol which favors such classes. Therefore, we use newly proposed splits [44] for the above five datasets to prevent such a bias and make results of our work follow protocols that offer a fair comparability with the latest state-of-the-art methods.

Parameters. In this paper, we make use of the data available in [44]. For image embeddings, we use the 2048 dimensional feature vectors extracted from top-layer pooling units of ResNet-101 [15] which was pre-trained on the ImageNet dataset [35]. For class embeddings, we use real-valued per-class attribute vectors provided for the aPY, AWA1, AWA2, CUB and SUN datasets. We perform the

mean subtraction and the ℓ_2-norm normalization on the datapoints and attribute vectors, respectively.

To learn $\boldsymbol{W}$, we applied SGD with mini-batches of size $I = 10$, we set the moving average of the squared gradient used in the Root Mean Square Propagation (RMSprop) [16] solver to $\gamma = 0.99$ and ran the solver for 5–10 epochs. Kernel parameters σ and λ were selected via cross-validation. For the Polynomial kernel, we chose $r = 2, 4, 6$ and parameter $\alpha = 1$. The bias c was selected via cross-validation.

Testing protocols. To evaluate our algorithms, we follow two standard protocols as detailed in [44]. Firstly, we report the mean top-1 accuracy when training on the training data and testing on the classes unseen at the training time. Next, for the generalized zero-shot learning protocol, we perform testing on classes both seen and unseen during the training step. For that, we use the harmonic mean of training and test accuracies as advocated by [44]:

$$H = 2\frac{\text{Acc}_S \cdot \text{Acc}_U}{\text{Acc}_S + \text{Acc}_U}, \tag{14}$$

where Acc_S and Acc_U denote the accuracy for the seen and unseen at the training stage classes. This strategy can flag up algorithms which overfit to either seen or unseen classes.

Baselines. We compare our zero-shot kernel learning approach to several works which include: DAP and IAP [25], CONSE [29], CMT [38], SSE [46], LATEM [43], ALE [2], DEVISE [13], SJE [3], ESZSL [34], SYNC [7] and SAE [18]. Section 2 contains more details about these methods.

Our methods. We compare the following approaches to the state of the art: the Polynomial kernels denoted (*Polynomial*) for $\alpha = 1$ and degree $r = 2, 4, 6$, respectively, which are combined with Eq. (11). Moreover, we evaluate the RBF kernels such as (*Cauchy*) and (*Gaussian*) which are combined with the formulation in Eq. (10). Lastly, we evaluate the Cauchy and Gaussian kernels (*Cauchy-Ort*) and (*Gaussian-Ort*) combined with our soft/implicit incoherence formulation according to Eq. (6) and (9).

5.1. Evaluations

We start our evaluations on the standard protocol followed by the generalized zero-shot learning protocol. Subsequently, we perform a sensitivity analysis of our model w.r.t. its hyperparameters given the validation and testing data to demonstrate the robustness of zero-shot kernel learning. The results presented in Tables 2 and 3 were obtained via cross-validation to prevent overfitting to the testing data.

Standard protocol. Table 2 lists our results and indicates the scores attained by other recent approaches. Firstly, we note that the Polynomial kernels (*Polynomial*, $r = 2$) and (*Polynomial*, $r = 4$), and the RBF kernels (*Cauchy-Ort*) and (*Gaussian-Ort*) for which we imposed soft/implicit incoherence constraints on $\boldsymbol{W}$, outperform the state-of-the-art approaches (the upper part of the table) on 4 out of 5 datasets

Method		AWA1	AWA2	SUN	CUB	aPY	Better than SOA
DAP	[25]	44.1	46.1	39.9	40.0	33.8	
IAP	[25]	35.9	35.9	19.4	24.0	36.6	
CONSE	[29]	45.6	44.5	38.8	34.3	26.9	
CMT	[38]	39.5	37.9	39.9	34.6	28.0	
SSE	[46]	60.1	61.0	51.5	43.9	34.0	
LATEM	[43]	55.1	55.8	55.3	49.3	35.2	
ALE	[2]	59.9	62.5	58.1	54.9	39.7	
DEVISE	[13]	54.2	59.7	56.5	52.0	39.8	
SJE	[3]	65.6	61.9	53.7	53.9	32.9	
ESZSL	[34]	58.2	58.6	54.5	53.9	38.3	
SYNC	[7]	54.0	46.6	56.3	55.6	23.9	
SAE	[18]	53.0	54.1	40.3	33.3	8.3	
Polynomial, $r = 2$		66.2	64.9	58.7	54.5	41.6	**4/5**
Polynomial, $r = 4$		65.0	64.3	59.7	**57.1**	41.7	**4/5**
Polynomial, $r = 6$		63.6	63.3	59.2	54.6	41.5	3/5
Cauchy		60.1	58.3	57.8	48.7	35.2	0/5
Cauchy-Ort		**71.0**	69.9	60.4	49.3	41.9	**4/5**
Gaussian		60.5	61.6	60.6	52.2	38.9	1/5
Gaussian-Ort		70.1	**70.5**	**61.7**	51.7	**45.3**	**4/5**

Table 2: Evaluations on the standard protocol and the newly proposed datasplits. (*Better than SOA*) column indicates the number of datasets on which our methods outperform the state-of-the-art methods listed in the upper part of the table.

(indicated by 4/5 in the table). In contrast, kernels (*Cauchy*) and (*Gaussian*), for which we imposed no such constraints, perform notably worse. This validates the benefits of decoherence in our model. Moreover, we note that the Gaussian kernel (*Gaussian-Ort*) attains the best results on 3 out of 5 datasets (highlighted in bold in the table) when compared to our (*Cauchy-Ort*) and (*Polynomial*, $r = 4$). Result-wise, (*Gaussian-Ort*) outperforms the other state-of-the-art methods on AWA1, AWA2, SUN and aPY by 4.5, 8, 3.6, 5.5% top-1 accuracy. In contrast, (*Polynomial*, $r = 4$) outperforms other state-of-the-art approaches on CUB by 2.2%.

We conjecture that the good performance of the Gaussian kernel can be attributed to its ability to 'elevate' datapoints to a potentially infinite-dimensional Hilbert space where a decision boundary separating datapoints according to labels can be found with ease *e.g.*, linearly non-separable datapoints become separable. We expect that the Cauchy kernel may also be beneficial due to its slowly decaying tails (compared to Gaussian) which results in stronger non-local influences. Lastly, we expect that the non-linearity of Polynomial kernels of degree $r = 4$ may also be sufficient for separating otherwise linearly non-separable data.

Generalized protocol. Table 3 presents our results on the generalized zero-shot learning protocol. Firstly, we note that our (*Cauchy-Ort*) approach outperforms other state-of-the-art approaches on 3 out of 5 datasets (indicated by 3/5

Method		AWA1			AWA2			SUN			CUB			aPY			Better than SOA
		ts	*tr*	*H*	*ts*	*tr*	*H*	*ts*	*tr*	*H*	*ts*	*tr*	*H*	*ts*	*tr*	*H*	
DAP	[25]	0.0	88.7	0.0	0.0	84.7	0.0	4.2	25.1	7.2	1.7	67.9	3.3	4.8	78.3	8.0	
IAP	[25]	2.1	78.2	4.1	0.9	87.6	1.8	1.0	37.8	1.8	0.2	72.8	0.4	5.7	65.6	10.4	
CONSE	[29]	0.4	88.6	0.8	0.5	90.6	1.0	6.8	39.9	11.6	1.6	72.2	3.1	0.0	91.2	0.0	
CMT	[38]	0.9	87.6	1.8	0.5	90.0	1.0	8.1	21.8	11.8	7.2	60.1	8.7	1.4	85.2	2.8	
*CMT**	[38]	8.4	86.9	15.3	8.7	89.0	15.9	8.7	28.0	13.3	4.7	60.1	8.7	10.9	74.2	19.0	
SSE	[46]	7.0	80.5	12.9	8.1	82.5	14.8	2.1	36.4	4.0	8.5	46.9	14.4	0.2	78.9	0.4	
LATEM	[43]	7.3	71.7	13.3	11.5	77.3	20.0	14.7	28.8	19.5	15.2	57.3	24.0	0.1	73.0	0.2	
ALE	[2]	16.8	76.1	27.5	14.0	81.8	23.9	21.8	33.1	26.3	23.7	62.8	34.4	4.6	73.7	8.7	
DEVISE	[13]	13.4	68.7	22.4	17.1	74.7	27.8	16.9	27.4	20.9	23.8	53.0	32.8	4.9	76.9	9.2	
SJE	[3]	11.3	74.6	19.6	8.0	73.9	14.4	14.7	30.5	19.8	23.5	59.2	33.6	3.7	55.7	6.9	
ESZSL	[34]	6.6	75.6	12.1	5.9	77.8	11.0	11.0	27.9	15.8	12.6	63.8	21.0	2.4	70.1	4.6	
SYNC	[7]	8.9	87.3	16.2	10.0	90.5	18.0	7.9	43.3	13.4	11.5	70.9	19.8	7.4	66.3	13.3	
SAE	[18]	1.8	77.1	3.5	1.1	82.2	2.2	8.8	18.0	11.8	7.8	54.0	13.6	0.4	80.9	0.9	
Polynomial, $r=2$		5.8	77.3	10.7	6.4	78.8	11.8	20.6	31.5	24.9	16.7	61.3	26.2	4.8	77.5	9.0	0/5
Polynomial, $r=4$		5.7	78.7	10.6	7.0	83.0	13.0	20.0	31.7	24.5	**24.2**	63.9	**35.1**	5.7	79.2	10.6	1/5
Polynomial, $r=6$		8.3	78.1	15.0	8.7	81.6	15.7	21.0	31.0	25.1	23.8	58.6	33.8	4.9	78.3	9.2	0/5
Cauchy		6.0	79.9	11.1	6.2	82.7	11.5	16.1	29.7	20.9	18.2	49.6	26.6	1.0	84.9	2.0	0/5
Cauchy-Ort		**18.3**	79.3	**29.8**	17.6	80.9	29.0	19.8	29.1	23.6	19.9	52.5	28.9	11.9	76.3	**20.5**	**3/5**
Gaussian		6.1	81.3	11.4	7.3	79.1	13.3	18.2	33.2	23.5	17.5	59.9	27.1	3.0	82.3	5.8	0/5
Gaussian-Ort		17.9	82.2	29.4	**18.9**	82.7	**30.8**	20.1	31.4	24.5	21.6	52.8	30.6	10.5	76.2	18.5	2/5

Table 3: Evaluations on the generalized zero-shot learning protocol and the newly proposed datasplits. We indicate the mean top-1 accuracy on (*tr*) train+test classes and (*ts*) test classes only. Moreover, (*Better than SOA*) indicates the number of datasets on which our methods outperform the other state-of-the-art methods (the upper part of the table) according to the harmonized score (*H*).

in the table) according to the generalized score (*H*) which takes into account the quality of zero-shot learning when testing it on classes which were seen and unseen during the training step. Our (*Cauchy-Ort*) is closely followed in terms of scores by (*Gaussian-Ort*) and (*Polynomial*, $r=4$) which outperform other state-of-the-art methods on 2 and 1 out of 5 datasets, respectively. Moreover, (*Cauchy-Ort*) attains the best results on AWA1 and aPY (highlighted in bold in the table) when compared to our (*Gaussian-Ort*) which performs the best on AWA2. Number-wise, (*Gaussian-Ort*) outperforms the other state-of-the-art methods (the upper part of the table) on AWA1, AWA2 and aPY by 2.3, 1.2 and 2.5% measured according to the generalized score (*H*).

We note that our models which impose our soft/implicit incoherence outperform again the variants which do not impose it. Moreover, the Cauchy kernel appears to be an adequate choice for varied testing tasks. Cauchy may be less prone to overfitting to local clusters of datapoints as its tails decay slower compared to tails of Gaussian.

Sensitivity analysis. Robust algorithms are expected to generalize well to unseen data and avoid overfitting *e.g.*, best classification scores on the validation and test data may be far away from each other when considered as a function of hyperparameters. Moreover, oversensitivity to the choice of hyperparameters may result in an algorithm which is hard to fine-tune. Below we show how our methods behave w.r.t. the choice of radius σ for the shift-invariant Gaussian and Cauchy kernels, the bias parameter c of the Polynomial kernel and the regularization parameter λ from Eq. (6).

Figure 3 (columns 1–3) shows how our zero-shot kernel learning performs on the AWA1 dataset w.r.t. the listed above hyperparameters. The standard evaluation protocol is used. The top row demonstrates that the kernel radius is an important parameter in our setup. For (*Gaussian-Ort*), the validation and test curves vary smoothly. The best results are attained for $\sigma=0.6$ and $\sigma=1$, respectively. The difference in the testing accuracy evaluated at $\sigma=0.6$ and $\sigma=1$ amounts to 2.5%. A similar trend emerges for (*Cauchy-Ort*). Such a discrepancy between the validation and testing scores shows that there is a visible domain shift between the validation and test problems. This result is typical for the knowledge transfer tasks such as domain adaptation and zero-shot learning. For (*Polynomial*, $r=4$), the validation and testing scores attain maximum for the same $c=1$, however, they are $\sim$5% below the performance of (*Gaussian-Ort*) and (*Cauchy-Ort*). This reveals that while the Polynomial kernel may generalize well, it lacks the capacity of the RBF kernels to capture highly non-linear data patterns.

Figure 3 (columns 1–3, bottom) show our scores w.r.t. the regularization parameter λ which, in our setting, controls how strongly the between-class datapoints are pushed from each other after the projection into the attribute space.

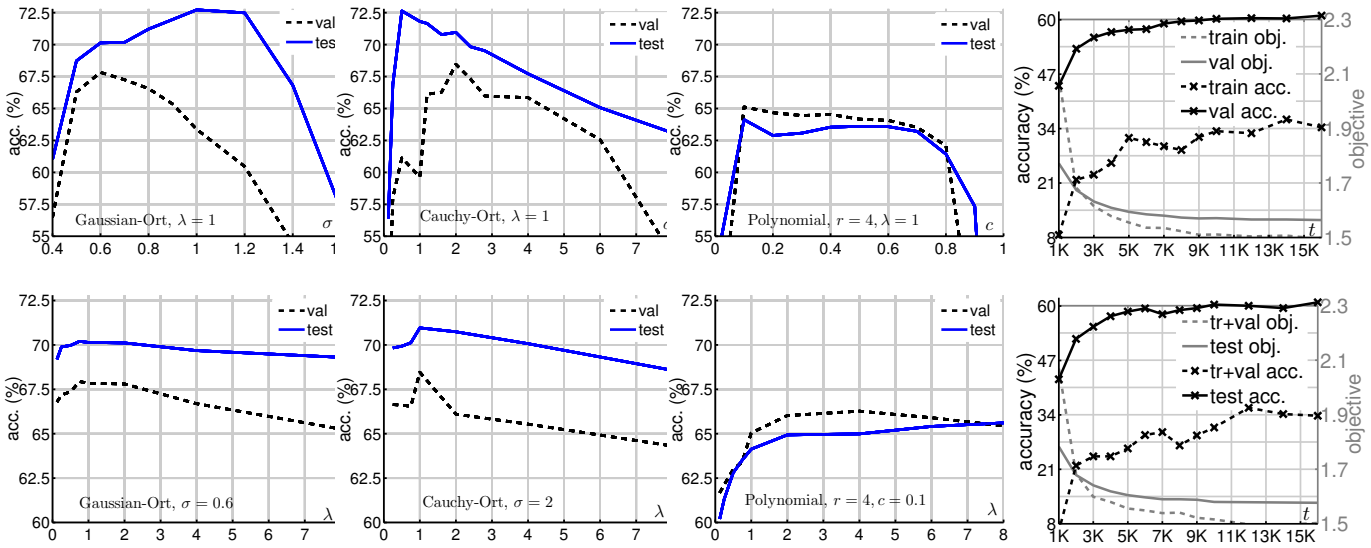

Figure 3: (Columns 1–3) Validation (*val*) and testing (*test*) accuracies on AWA1 for the standard evaluation protocol. We vary σ, λ and c. In the top row, we vary the radius σ for kernels (*Gaussian-Ort*) and (*Cauchy-Ort*). For (*Polynomial*, $r = 4$), we vary its bias c. The bottom row shows the performance w.r.t. regularization λ of our kernel polarization model given fixed σ and/or c selected earlier via cross-validation. (Column 4) The accuracy and the objective w.r.t. the number of iterations t for (*Gaussian-Ort*) on the SUN dataset. The top and bottom plots concern training (*train*) vs. validation (*val*) and training plus validation (*tr+val*) vs. testing (*test*), respectively.

It appears that the peaks in validation and testing accuracies match each other well for all three kernels used in this experiment. The value of λ does not affect dramatically the performance of our algorithm and the range for which the best performance is attained varies from 0.8–2. However, it is also clear that if $\lambda \to 0$ or $\lambda \to \infty$, the scores drop. This demonstrates the importance of balancing the impact of so-called within- and between-class statistics in our zero-shot kernel model which 'polarizes' these statistics.

Figure 3 (column 4) shows how our algorithm behaves w.r.t. the number of solver iterations t on the SUN dataset. The top row shows that the training objective attains lower values compared to the validation objective for $t \geq 2000$. As the objective decreases, there is a clear increase in the accuracy for both the training and validation curves. The same behavior is observed in the bottom row which demonstrates that our algorithm is stable w.r.t. to t.

Finally, Table 4 shows that Proposition 4 and following from it Eq. (6) promote the weak incoherence.

6. Conclusions

In this paper, we have proposed a novel approach to zero-shot learning by the use of kernels. Our model is inspired by the the Linear Discriminant Analysis [12, 9] and kernel alignment methods [8, 4]. To the best of our knowledge, we are the first to show how to combine zero-shot learning with the Polynomial and the RBF family of kernels to obtain a non-linear compatibility function. Our model shows that a learned projection that embeds datapoints in the attribute space and from there, in the non-linear Hilbert space, is a robust tool for zero-shot learning. We learn an approximate subspace by encouraging in a soft/implicit manner the incoherence between column vectors of the projection matrix. Therefore, our projection incorporates the rotation and scaling but prevents the shear which causes overfitting due to more degrees of freedom in such an unconstrained model.

Each of our models achieve state-of-the-art results on up to four out of five datasets on the standard zero-shot learning benchmark for new stricter recently proposed datasplits. Moreover, each of our models obtain state-of-the-art results on up to three out of five datasets on the new generalized zero-shot learning benchmark which takes into account so-called harmonized scores for classes seen and unseen during the training process. We note that if we were to pick one best kernel per dataset, this would lead to further improvements in accuracy. For future directions, this warrants an investigation into multiple kernel learning in the context of zero-shot kernel learning. We also plan to investigate the benefit of learning class-wise subspace matrices.

Gauss.	AWA1	AWA2	SUN	CUB	aPY
Non-Ort	376.4	420.7	209.1	2361.4	342.1
Eq. (6)	**131.6**	**205.0**	**59.1**	**1018.8**	**259.5**
Cauchy	AWA1	AWA2	SUN	CUB	aPY
Non-Ort	357.7	346.9	187.1	2614.4	334.0
Eq. (6)	**138.3**	**178.2**	**124.1**	**1227.5**	**214.9**

Table 4: Incoherence on various kernels and datasets. For the ℓ_2-norm normalized columns of $\boldsymbol{W}$ we computed $||\boldsymbol{W}^T\boldsymbol{W} - \mathbb{I}||_F^2$. Lower values indicate more incoherence between columns of $\boldsymbol{W}$.

References

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

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. *CVPR*, pages 819–826, 2013. 1, 2, 6, 7

[3] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. *CVPR*, pages 2927–2936, 2015. 2, 6, 7

[4] Y. Baram. Learning by kernel polarization. *Neural Computation*, 17:1264–1275, 2005. 1, 3, 8

[5] J. Basak. A least square kernel machine with box constraints. *ICPR*, 2008. 2, 3

[6] J. Baxter, R. Caruana, T. Mitchell, L. Y. Pratt, D. L. Silver, and S. Thrun. Learning to learn: Knowledge consolidation and transfer in inductive systems. NIPS Workshop, http://plato.acadiau.ca/courses/comp/dsilver/NIPS95_LTL/transfer.workshop.1995.html, 1995. Accessed: 30-10-2016. 1

[7] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. *CVPR*, pages 5327–5336, 2016. 2, 6, 7

[8] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. S. Kandola. On kernel-target alignment. *NIPS*, pages 367–373, 2002. 1, 2, 3, 8

[9] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, 2 edition, 2001. 1, 8

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://pascallin.ecs.soton.ac.uk/challenges/VOC, 2007. 5

[11] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. *CVPR*, pages 1778–1785, 2009. 1, 5

[12] R. A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7(2):179–188, 1936. 1, 2, 8

[13] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. *NIPS*, pages 2121–2129, 2013. 1, 2, 6, 7

[14] M. Harandi, M. Salzmann, and R. Hartley. Joint dimensionality reduction and metric learning: A geometric take. *ICML*, page 14041413, 2017. 3

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, June 2016. 5

[16] G. Hinton. Neural Networks for Machine Learning Lecture 6a: Overview of mini-batch gradient descent Reminder: The error surface for a linear neuron. Lecture notes, https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf, 2017. Accessed: 10-11-2017. 5, 6

[17] T. Joachims. Optimizing search engines using clickthrough data. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142, 2002. 2

[18] E. Kodirov, T. Xiang, and S. Gong. Semantic autoencoder for zero-shot learning. *CVPR*, 2017. 6, 7

[19] P. Koniusz and A. Cherian. Sparse coding for third-order super-symmetric tensor descriptors with application to texture recognition. *CVPR*, 2016. 2

[20] P. Koniusz, A. Cherian, and F. Porikli. Tensor representations via kernel linearization for action recognition from 3D skeletons. *ECCV*, 2016. 2

[21] P. Koniusz, Y. Tas, and F. Porikli. Domain adaptation by mixture of alignments of second- or higher-order scatter tensors. *CVPR*, 2017. 1

[22] P. Koniusz, F. Yan, P. Gosselin, and K. Mikolajczyk. Higher-order occurrence pooling for bags-of-words: Visual concept detection. *PAMI*, 2016. 2

[23] M. Kstinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. *CVPR*, pages 2288–2295, 2012. 1, 3

[24] S. Kumar Roy, Z. Mhammedi, and M. Harandi. Geometry aware constrained optimization techniques for deep learning. *CVPR*, 2018. 3

[25] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *TPAMI*, 36(3):453–465, 2014. 2, 5, 6, 7

[26] H. Larochelle, D. Erhan, and Y. Bengio. Zero-data learning of new tasks. *AAAI*, 1(2):3, 2008. 1

[27] W. Li, T. Tommasi, F. Orabona, D. Vázquez, M. López, J. Xu, and H. Larochelle. Task-cv: Transferring and adapting source knowledge in computer vision. ECCV Workshop, http://adas.cvc.uab.es/task-cv2016, 2016. Accessed: 22-11-2016. 1

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *NIPS*, 2013. 1, 2

[29] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *CoRR*, abs/1312.5650, 2013. 1, 2, 6, 7

[30] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. *CVPR*, pages 2751–2758, 2012. 1, 5

[31] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. *EMNLP*, 2014. 2

[32] I. Ramrez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. *CVPR*, pages 3501–3508, 2010. 2

[33] M. Rohrbach, M. Stark, and B. Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. *CVPR*, pages 1641–1648, 2011. 1

[34] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. *ICML*, pages 2152–2161, 2015. 2, 6, 7

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 5

[36] B. Scholkopf and A. Smola. *Learning with kernels*. The MIT Press, 2002. 2, 3, 5

[37] B. Scholkopf, K. Sung, C. J. C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers. *SP*, 45(11):2758–2765, Nov. 1997. 1, 2, 3

[38] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. *NIPS*, pages 935–943,

2013. 1, 2, 6, 7

[39] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. *CVPR*, pages 3081–3088, 2010. 1

[40] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 2005. 2

[41] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 5

[42] J. Weston, S. Bengio, and N. Usunier. Learning to rank with joint word-image embedding. *ECML*, 2010. 2

[43] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. *CVPR*, pages 69–77, 2016. 2, 6, 7

[44] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *arXiv preprint arXiv:1707.00600*, 2017. 1, 4, 5, 6

[45] Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning – the Good, the Bad and the Ugly. *CVPR*, 2017. 1, 2, 5

[46] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. *ICCV*, pages 4166–4174, 2015. 2, 6, 7