Write a Classifier: Zero-Shot Learning Using Purely Textual Descriptions

Mohamed Elhoseiny Babak Saleh Ahmed Elgammal
Department of Computer Science, Rutgers University, New Brunswick, NJ
[m.elhoseiny,babaks,elgammal]@cs.rutgers.edu

Abstract

The main question we address in this paper is how to use purely textual description of categories with no training images to learn visual classifiers for these categories. We propose an approach for zero-shot learning of object categories where the description of unseen categories comes in the form of typical text such as an encyclopedia entry, without the need to explicitly defined attributes. We propose and investigate two baseline formulations, based on regression and domain adaptation. Then, we propose a new constrained optimization formulation that combines a regression function and a knowledge transfer function with additional constraints to predict the classifier parameters for new classes. We applied the proposed approach on two fine-grained categorization datasets, and the results indicate successful classifier prediction.

1. Introduction

One of the main challenges for scaling up object recognition systems is the lack of annotated images for real-world categories. Typically there are few images available for training classifiers for most of these categories. This is reflected in the number of images per category available for training in most object categorization datasets, which, as pointed out in [27], shows a Zipf distribution. The problem of lack of training images becomes even more severe when we target recognition problems within a general category, *i.e.*, fine-grained categorization, for example building classifiers for different bird species or flower types (there are estimated over 10000 living bird species, similar for flowers). Researchers try to exploit shared knowledge between categories to target such scalability issue. This motivated many researchers who looked into approaches that learn visual classifiers from few examples, *e.g.* [4, 9, 2]. This even motivated some recent work on zero-shot learning of visual categories where there are no training images available for test categories (unseen classes), *e.g.* [17]. Such approaches exploit the similarity (visual or semantic) between seen classes and unseen ones, or describe unseen classes in terms of a learned vocabulary of semantic visual attributes.

In contrast to the lack of reasonable size training sets for a large number of real world categories, there are abundant of textual descriptions of these categories. This comes in the form of dictionary entries, encyclopedia articles, and various online resources. For example, it is possible to find several good descriptions of a "bobolink" in encyclopedias of birds, while there are only a few images available for that bird online.

The main question we address in this paper is how to use purely textual description of categories with no training images to learn visual classifiers for these categories. In other words, we aim at zero-shot learning of object categories where the description of unseen categories comes in the form of typical text such as an encyclopedia entry. We explicitly address the question of how to automatically decide which information to transfer between classes without the need of human intervention. In contrast to most related work, we go beyond the simple use of tags and image captions, and apply standard Natural Language Processing techniques to typical text to learn visual classifiers.

Similar to the setting of zero-shot learning, we use classes with training data (seen classes) to predict classifiers for classes with no training data (unseen classes).Recent works on zero-shot learning of object categories focused on leveraging knowledge about common attributes and shared parts [17]. Typically, attributes [28, 7] are manually defined by humans and are used to transfer knowledge between seen and unseen classes. In contrast, in our work we do not use any explicit attributes. The description of a new category is purely textual and the process is totally automatic without human annotation beyond the category labels.

The contribution of the paper is on exploring this new problem, which to the best of our knowledge, is not explored in the computer vision community. We learn from an image corpus and a textual corpus, however not in the form of image-caption pairs, instead the only alignment between the corpora is at the level of the category. We propose and investigate two baseline formulations based on regression and domain adaptation. Then we propose a new constrained optimization formulation that combines a re-

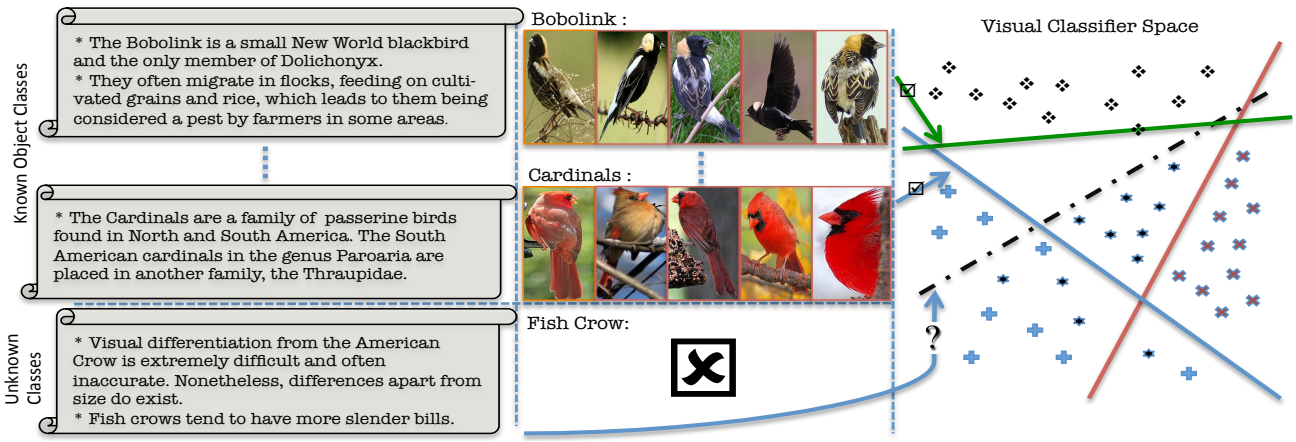

Figure 1: Problem Definition: Zero-shot learning with textual description. Left: synopsis of textual descriptions for bird classes. Middle: images for "seen classes". Right: classifier hyperplanes in the feature space. The goal is to estimate a new classifier parameter given only a textual description

gression function and a knowledge transfer function with additional constraints to solve the problem.

Beyond the introduction and the related work sections, the paper is structured as follows: Sec 3 introduces the problem definition and proposed baseline solutions. Sec 4 describes the solution framework. Sec 5 explains the experiments performed on Flower Dataset [20] (102 classes) and Caltech-UCSD dataset [32] (200 classes).

2. Related Work

Our proposed work can be seen in the context of knowledge sharing and inductive transfer. In general, knowledge transfer aims at enhancing recognition by exploiting shared knowledge between classes. Most existing research focused on knowledge sharing within the visual domain only, *e.g.* [12]; or exporting semantic knowledge at the level of category similarities and hierarchies, *e.g.* [10, 27]. We go beyond the state-of-the-art to explore cross-domain knowledge sharing and transfer. We explore how knowledge from the visual and textual domains can be used to learn across-domain correlation, which facilitates prediction of visual classifiers from textual description.

Motivated by the practical need to learn visual classifiers of rare categories, researchers have explored approaches for learning from a single image (one-shot learning [18, 9, 11, 2]) or even from no images (zero-shot learning). One way of recognizing object instances from previously unseen test categories (the zero-shot learning problem) is by leveraging knowledge about common attributes and shared parts. Typically an intermediate semantic layer is introduced to enable sharing knowledge between classes and facilitate describing knowledge about novel unseen classes, *e.g.* [22]. For instance, given adequately labeled training data, one can learn classifiers for the attributes occurring in the training object categories. These classifiers can then be used to recognize the same attributes in object instances from the novel test categories. Recognition can then proceed on the basis of these learned attributes [17, 7]. Such attribute-based "knowledge transfer" approaches use an intermediate visual attribute representation to enable describing unseen object categories. Typically attributes are manually defined by humans to describe shape, color, surface material, *e.g.*, furry, striped, *etc*. Therefore, an unseen category has to be specified in terms of the used vocabulary of attributes. Rohrbach *et al*. [25] investigated extracting useful attributes from large text corpora. In [23], an approach was introduced for interactively defining a vocabulary of attributes that are both human understandable and visually discriminative. In contrast, our work does not use any explicit attributes. The description of a new category is purely textual.

The relation between linguistic semantic representations and visual recognition have been explored. For example in [4], it was shown that there is a strong correlation between semantic similarity between classes, based on WordNet, and confusion between classes. Linguistic semantics in terms of nouns from WordNet [19] have been used in collecting large-scale image datasets such as ImageNet[5] and Tiny Images [30]. It was also shown that hierarchies based on WordNet are useful in learning visual classifiers, *e.g.* [27].

One of the earliest work on learning from images and text corpora is the work of Barnard *et al*. [1], which showed that learning a joint distribution of words and visual elements facilitates clustering the images in a semantic way, generating illustrative images from a caption, and generating annotations for novel images. There has been an increasing recent interest in the intersection between computer vision and natural language processing with researches that focus on generating textual description of im-

ages and videos, *e.g.* [8, 16, 34, 14]. This includes generating sentences about objects, actions, attributes, patial relation between objects, contextual information in the images, scene information, *etc*. In contrast, our work is different in two fundamental ways. In terms of the goal, we do not target generating textual description from images, instead we target predicting classifiers from text, in a zero-shot setting. In terms of the learning setting, the textual descriptions that we use is at the level of the category and do not come in the form of image-caption pairs, as in typical datasets used for text generation from images, *e.g.* [21].

3. Problem Definition

Fig 1 illustrates the learning setting. The information in our problem comes from two different domains: the visual domain and the textual domain, denoted by $\mathcal{V}$ and $\mathcal{T}$, respectively. Similar to traditional visual learning problems, we are given training data in the form $V = \{(x_i, l_i)\}_N$, where x_i is an image and $l_i \in \{1 \cdots N_{sc}\}$ is its class label. We denote the number of classes available at training as N_{sc}, where sc indicates "seen classes". As typically done in visual classification setting, we can learn N_{sc} binary one-vs-all classifiers, one for each of these classes. Let us consider a typical binary linear classifier in the feature space in the form

$$f_k(\mathbf{x}) = \mathbf{c}_k^\mathsf{T} \cdot \mathbf{x}$$

where $\mathbf{x}$ is the visual feature vector amended with 1, and $\mathbf{c}_k \in \mathbb{R}^{d_v}$ is the linear classifier parameters for class k. Given a test image, its class is determined by

$$l^* = \arg\max_k f_k(\mathbf{x})$$

Our goal is to be able to predict a classifier for a new category based only on the learned classes and a textual description(s) of that category. In order to achieve that, the learning process has to also include textual description of the seen classes (as shown in Fig 1). Depending on the domain we might find a few, a couple, or as little as one textual description to each class. We denote the textual training data for class k by $\{t_i \in \mathcal{T}\}^k$. In this paper we assume we are dealing with the extreme case of having only one textual description available per class, which makes the problem even more challenging. However, the formulation we propose in this paper directly applies to the case of multiple textual descriptions per class. Similar to the visual domain, the raw textual descriptions have to go through a feature extraction process, which will be described in Sec 5. Let us denote the extracted textual feature by $T = \{\mathbf{t}_k \in \mathbb{R}^{d_t}\}_{k=1\cdots N_{sc}}$.

Given a textual description $\mathbf{t}_*$ of a new unseen category, $\mathcal{C}$, the problem can now be defined as predicting a one-vs-all classifier parameters $c(\mathbf{t}_*)$, such that it can be directly used to classify any test image $\mathbf{x}$ as

$$\begin{aligned} c(\mathbf{t}_*)^\mathsf{T} \cdot \mathbf{x} > 0 \quad & \text{if } \mathbf{x} \text{ belongs to } \mathcal{C} \\ c(\mathbf{t}_*)^\mathsf{T} \cdot \mathbf{x} < 0 \quad & \text{otherwise} \end{aligned} \tag{1}$$

In what follows, we introduce two possible frameworks for this problem and discuss potential limitations for them, which leads next to the proposed formulation.

3.1. Regression Models

A straightforward way to solve this problem is to pose it as a regression problem where the goal is to use the textual data and the learned classifiers, $\{(\mathbf{t}_k, \mathbf{c}_k)\}_{k=1\cdots N_{sc}}$ to learn a regression function from the textual feature domain to the visual classifier domain, *i.e.*, a function $c(\cdot) : \mathbb{R}^{d_t} \rightarrow \mathbb{R}^{d_v}$. The question is which regression model would be suitable for this problem? and would posing the problem this way give reasonable results?

A typical regression model, such as ridge regression [13] or Gaussian Process (GP) Regression [24], learns the regressor to each dimension of the output domain (the parameters of a linear classifier) separately, *i.e.* a set of function $c^j(\cdot) : \mathbb{R}^{d_t} \rightarrow \mathbb{R}$. Clearly this will not capture the correlation between the visual and textual domain. Instead, a structured prediction regressor would be more suitable since it would learn the correlation between the input and output domain. However, even a structure prediction model, will only learn the correlation between the textual and visual domain through the information available in the input-output pairs $(\mathbf{t}_k, \mathbf{c}_k)$. Here the visual domain information is encapsulated in the pre-learned classifiers and prediction does not have access to the original data in the visual domain. Instead we need to directly learn the correlation between the visual and textual domain and use that for prediction.

Another fundamental problem that a regressor would face, is the sparsity of the data; the data points are the textual description-classifier pairs, and typically the number of classes can be very small compared to the dimension of the classifier space (*i.e.* $N_{sc} \ll d_v$). In a setting like that, any regression model is bound to suffer from an under fitting problem. This can be best explained in terms of GP regression, where the predictive variance increases in the regions of the input space where there are no data points. This will result in poor prediction of classifiers at these regions.

3.2. Knowledge Transfer Models

An alternative formulation is to pose the problem as domain adaptation from the textual to the visual domain. In the computer vision context, domain adaptation work has focused on transferring categories learned from a source domain, with a given distribution of images, to a target domain with different distribution, *e.g.*, images or videos from different sources [33, 26, 15, 6]. What we need is an approach

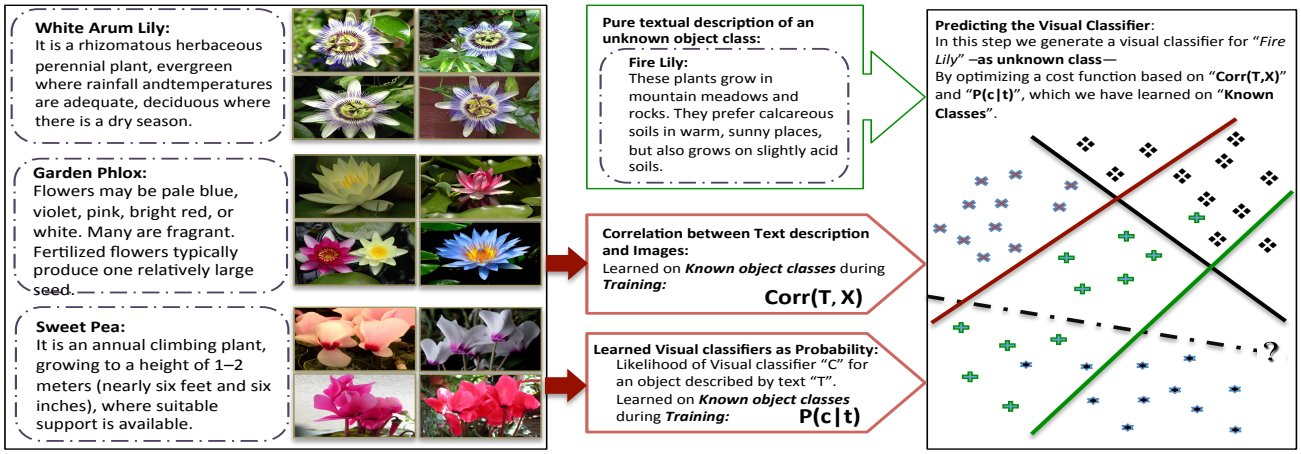

Figure 2: Illustration of the Proposed Solution Framework for the task Zero-shot learning from textual description.

that learns the correlation between the textual domain features and the visual domain features, and uses that correlation to predict new visual classifier given textual features.

In particular, in [15] an approach for learning cross domain transformation was introduced. In that work a regularized asymmetric transformation between points in two domains were learned. The approach was applied to transfer learned categories between different data distributions, both in the visual domain. A particular attractive characteristic of [15], over other domain adaptation models, is that the source and target domains do not have to share the same feature spaces or the same dimensionality.

Inspired by [15], we can formulate the zero-shot learning problem as a domain adaptation. This can be achieve by learning a linear (or nonlinear kernalized) transfer function $\mathbf{W}$ between $\mathcal{T}$ and $\mathcal{V}$. The transformation matrix $\mathbf{W}$ can be learned by optimizing, with a suitable regularizer, over constraints of the form $\mathbf{t}^\mathsf{T}\mathbf{W}\mathbf{x} \geq l$ if $\mathbf{t} \in \mathcal{T}$ and $\mathbf{x} \in \mathcal{V}$ belong to the same class, and $\mathbf{t}^\mathsf{T}\mathbf{W}\mathbf{x} \leq u$ otherwise. Here l and u are model parameters. This transfer function acts as a compatibility function between the textual features and visual features, which gives high values if they are from the same class and a low value if they are from different classes.

It is not hard to see that this transfer function can act as a classifier. Given a textual feature $\mathbf{t}^*$ and a test image, represented by $\mathbf{x}$, a classification decision can be obtained by $\mathbf{t}_*^\mathsf{T}\mathbf{W}\mathbf{x} \gtrless b$ where b is a decision boundary which can be set to $(l+u)/2$. Hence, our desired predicted classifier in Eq 1 can be obtained as $c(\mathbf{t}_*) = \mathbf{t}_*^\mathsf{T}\mathbf{W}$ (note that the features vectors are amended with ones). However, since learning $\mathbf{W}$ was done over seen classes only, it is not clear how the predicted classifier $c(\mathbf{t}_*)$ will behave for unseen classes. There is no guarantee that such a classifier will put all the seen data on one side and the new unseen class on the other side of that hyperplane.

4. Problem Formulation

4.1. Objective Function

The proposed formulation aims at predicting the hyperplane parameter $\mathbf{c}$ of a one-vs-all classifier for a new unseen class given a textual description, encoded by $\mathbf{t}$ and knowledge learned at the training phase from seen classes. Fig 2 illustrates our solution framework. At the training phase three components are learned:

Classifiers: a set of one-vs-all classifiers $\{\mathbf{c}_k\}$ are learned, one for each seen class.

Probabilistic Regressor: Given $\{(\mathbf{t}_k, \mathbf{c}_k)\}$ a regressor is learned that can be used to give a prior estimate for $p_{reg}(\mathbf{c}|\mathbf{t})$ (Details in Sec 4.3).

Domain Transfer Function: Given T and V a domain transfer function, encoded in the matrix $\mathbf{W}$ is learned, which captures the correlation between the textual and visual domains (Details in Sec 4.2).

Each of these components contains partial knowledge about the problem. The question is how to combine such knowledge to predict a new classifier given a textual description. The new classifier has to be consistent with the seen classes. The new classifier has to put all the seen instances at one side of the hyperplane, and has to be consistent with the learned domain transfer function. This leads to the following constrained optimization problem

$$\begin{aligned}\hat{c}(\mathbf{t}_*) = &\underset{\mathbf{c},\zeta_i}{\operatorname{argmin}}\big[\mathbf{c}^\mathsf{T}\mathbf{c} - \alpha\mathbf{t}_*{}^\mathsf{T}\mathbf{W}\mathbf{c} - \beta\ln(p_{reg}(\mathbf{c}|\mathbf{t}_*)) \\ &+ \gamma\sum\zeta_i\big] \\ &s.t.: -(\mathbf{c}^\mathsf{T}\mathbf{x}_i) \geq \zeta_i,\ \zeta_i \geq 0,\ \ i = 1\cdots N \\ &\mathbf{t}_*{}^\mathsf{T}\mathbf{W}\mathbf{c} \geq l \\ &\alpha,\beta,\gamma,l: \text{hyperparameters}\end{aligned} \tag{2}$$

The first term is a regularizer over the classifier $\mathbf{c}$. The second term enforces that the predicted classifier has high correlation with $\mathbf{t}_*^\mathsf{T}\mathbf{W}$. The third term favors a classifier that has high probability given the prediction of the regressor. The constraints $-\mathbf{c}^\mathsf{T}\mathbf{x}_i \geq \zeta_i$ enforce all the seen data instances to be at the negative side of the predicted classifier hyperplane with some missclassification allowed through the slack variables ζ_i. The constraint $\mathbf{t}_*{}^\mathsf{T}\mathbf{W}\mathbf{c} \geq l$ enforces that the correlation between the predicted classifier and $\mathbf{t}_*{}^\mathsf{T}\mathbf{W}$ is no less than l, this is to enforce a minimum correlation between the text and visual features.

4.2. Domain Transfer Function

To learn the domain transfer function $\mathbf{W}$ we adapted the approach in [15] as follows. Let $\mathbf{T}$ be the textual feature data matrix and $\mathbf{X}$ be the visual feature data matrix where each feature vector is amended with a 1. Notice that amending the feature vectors with a 1 is essential in our formulation since we need $\mathbf{t}^\mathsf{T}\mathbf{W}$ to act as a classifier. We need to solve the following optimization problem

$$\min_{\mathbf{W}} r(\mathbf{W}) + \lambda \sum_i c_i(\mathbf{T}\mathbf{W}\mathbf{X}^\mathsf{T}) \tag{3}$$

where c_i's are loss functions over the constraints and $r(\cdot)$ is a matrix regularizer. It was shown in [15], under condition on the regularizer, that the optimal $\mathbf{W}$ in Eq 3 can be computed using inner products between data points in each of the domains separately, which results in a kernalized non-linear transfer function; hence its complexity does not depend on the dimensionality of either of the domains. The optimal solution of 3 is in the form $\mathbf{W}^* = \mathbf{T}\mathbf{K}_T^{-\frac{1}{2}}\mathbf{L}^*\mathbf{K}_X^{-\frac{1}{2}}\mathbf{X}^\mathsf{T}$,where $\mathbf{K}_T = \mathbf{T}\mathbf{T}^\mathsf{T}$, $\mathbf{K}_X = \mathbf{X}\mathbf{X}^\mathsf{T}$. $\mathbf{L}^*$ is computed by minimizing the following minimization problem

$$\min_{\mathbf{L}}[r(\mathbf{L}) + \lambda \sum_p c_p(\mathbf{K}_T^{\frac{1}{2}}\mathbf{L}\mathbf{K}_X^{\frac{1}{2}})], \tag{4}$$

where $c_p(\mathbf{K}_T^{\frac{1}{2}}\mathbf{L}\mathbf{K}_X^{\frac{1}{2}}) = (max(0,(l - e_i\mathbf{K}_T^{\frac{1}{2}}\mathbf{L}\mathbf{K}_X^{\frac{1}{2}}e_j)))^2$ for same class pairs of index i,j, or $= (max(0,(e_i\mathbf{K}_T^{\frac{1}{2}}\mathbf{L}\mathbf{K}_X^{\frac{1}{2}}e_j - u)))^2$ otherwise, where e_k is a vector of zeros except a one at the k^{th} element, and $u > l$ (note any appropriate l, u could work. In our case, we used $l = 2$, $u = -2$). We used a Frobenius norm regularizer. This energy is minimized using a second order BFGS quasi-Newton optimizer. Once L is computed W^* is computed using the transformation above.

4.3. Probabilistic Regressor

There are different regressors that can be used, however we need a regressor that provide a probabilistic estimate $p_{reg}(\mathbf{c}|(t))$. For the reasons explained in Sec 3, we also need a structure prediction approach that is able to predict all the dimensions of the classifiers together. For these reasons, we use the Twin Gaussian Process (TPG) [3]. TGP encodes the relations between both the inputs and structured outputs using Gaussian Process priors. This is achieved by minimizing the Kullback-Leibler divergence between the marginal GP of the outputs (i.e. classifiers in our case) and observations (i.e. textual features). The estimated regressor output ($\tilde{c}(\mathbf{t}_*)$) in TGP is given by the solution of the following non-linear optimization problem [3] [1].

$$\begin{aligned}\tilde{c}(\mathbf{t}_*) = \operatorname*{argmin}_{\mathbf{c}}[K_C(\mathbf{c},\mathbf{c}) - 2k_c(\mathbf{c})^\mathsf{T}\mathbf{u} - \eta\log(K_C(\mathbf{c},\mathbf{c}) \\ - k_c(\mathbf{c})^\mathsf{T}(\mathbf{K}_C + \lambda_c\mathbf{I})^{-1}k_c(c))]\end{aligned} \tag{5}$$

where $\mathbf{u} = (\mathbf{K}_\mathsf{T} + \lambda_t\mathbf{I})^{-1}k_t(\mathbf{t}_*)$, $\eta = K_T(\mathbf{t}_*,\mathbf{t}_*) - k(\mathbf{t}_*)^\mathsf{T}\mathbf{u}$, $K_T(\mathbf{t}_l,\mathbf{t}_m)$ and $K_C(\mathbf{c}_l,\mathbf{c}_m)$ are Gaussian kernel for input feature $\mathbf{t}$ and output vector $\mathbf{c}$. $k_c(\mathbf{c}) = [K_C(\mathbf{c},\mathbf{c}_1),\cdots,K_C(\mathbf{c},\mathbf{c}_{N_{sc}})]^\mathsf{T}$. $k_t(\mathbf{t}_*) = [K_T(\mathbf{t}_*,\mathbf{t}_1),\cdots,K_T(\mathbf{t}_*,\mathbf{t}_{N_{sc}})]^\mathsf{T}$. λ_t and λ_c are regularization parameters to avoid overfitting. This optimization problem can be solved using a second order, BFGS quasi-Newton optimizer with cubic polynomial line search for optimal step size selection [3]. In this case the classifier dimension are predicted jointly. In this case $p_{reg}(\mathbf{c}|\mathbf{t}_*)$ is defined as a normal distribution.

$$p_{reg}(\mathbf{c}|\mathbf{t}_*) = \mathcal{N}(\mu_c = \tilde{c}(\mathbf{t}_*), \Sigma_c = \mathbf{I}) \tag{6}$$

The reason that $\Sigma_c = \mathbf{I}$ is that TGP does not provide predictive variance, unlike Gaussian Process Regression. However, it has the advantage of handling the dependency between the dimensions of the classifiers $\mathbf{c}$ given the textual features $\mathbf{t}$.

4.4. Solving for $\hat{c}$ as a quadratic program

According to the definition of $p_{reg}(\mathbf{c}|\mathbf{t}_*)$ for TGP, $\ln p(\mathbf{c}|\mathbf{t}_*)$ is a quadratic term in c in the form

$$\begin{aligned}-\ln p(\mathbf{c}|\mathbf{t}_*) &\propto (\mathbf{c} - \tilde{c}(\mathbf{t}_*))^\mathsf{T}(\mathbf{c} - \tilde{c}(\mathbf{t}_*)) \\ &= \mathbf{c}^\mathsf{T}\mathbf{c} - 2\mathbf{c}^\mathsf{T}\tilde{c}(\mathbf{t}_*) + \tilde{c}(\mathbf{t}_*)^\mathsf{T}\tilde{c}(\mathbf{t}_*)\end{aligned} \tag{7}$$

We reduce $-\ln p(\mathbf{c}|\mathbf{t}_*)$ to $-2\mathbf{c}^\mathsf{T}\tilde{c}(\mathbf{t}_*))$, since 1) $\tilde{c}(\mathbf{t}_*)^\mathsf{T}\tilde{c}(\mathbf{t}_*)$ is a constant (*i.e.* does not affect the optimization), 2) $\mathbf{c}^\mathsf{T}\mathbf{c}$ is already included as regularizer in equation 2. In our setting, the dot product is a better similarity measure between two hyperplanes. Hence, $-2\mathbf{c}^\mathsf{T}\tilde{c}(\mathbf{t}_*)$ is minimized. Given $-\ln p(\mathbf{c}|\mathbf{t}_*)$ from the TGP and $\mathbf{W}$, Eq 2 reduces to a quadratic program on $\mathbf{c}$ with linear constraints. We tried different quadratic solvers, however the IBM CPLEX solver [2] gives the best performance in speed and optimization for our problem.

[1] notice we are using $\tilde{\mathbf{c}}$ to denote the output of the regressor, while using $\hat{\mathbf{c}}$ to denote the output of the final optimization problem in Eq 2

[2] http://www-01.ibm.com/software/integration/optimization/cplex-optimizer

5. Experiments

5.1. Datasets

We used the CU200 Birds [32] (200 classes - 6033 images) and the Oxford Flower-102 [20] (102 classes - 8189 images) image dataset to test our approach, since they are among the largest and widely used fine-grained datasets. We generate textual descriptions for each class in both datasets. The CU200 Birds image dataset was created based on birds that have a corresponding Wikipedia article, so we have developed a tool to automatically extract Wikipedia articles given the class name. The tool succeeded to automatically generate 178 articles, and the remaining 22 articles was extracted manually from Wikipedia. These mismatches happens only when article title is a different synonym of the same bird class. On the other hand, Flower image dataset was not created using the same criteria as the Bird dataset, so classes of the Flower dataset classes does not necessarily have corresponding Wikipedia article. The tool managed to generate about 16 classes from Wikipedia out of 102, the remaining 86 articles was generated manually for each class from Wikipedia, Plant Database [3], Plant Encyclopedia [4], and BBC articles [5]. We plan to make the extracted textual description available as augmentations of these datasets. Sample textual description can be found in the supplementary material.

5.2. Extracting Textual Features

The textual features were extracted in two phases, which are typical in document retrieval literature. The first phase is an indexing phase that generates textual features with tf-idf (Term Frequency-Inverse Document Frequency) configuration (Term frequency as local weighting while inverse document frequency as a global weighting). The tf-idf is a measure of how important is a word to a text corpus. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others. We used the normalized frequency of a term in the given textual description [29]. The inverse document frequency is a measure of whether the term is common; in this work we used the standard logarithmic idf [29]. The second phase is a dimensionality reduction step, in which Clustered Latent Semantic Indexing (CLSI) algorithm [35] is used. CLSI is a low-rank approximation approach for dimensionality reduction, used for document retrieval. In the Flower Dataset, tf-idf features $\in \mathbb{R}^{8875}$ and after CLSI the final textual features $\in \mathbb{R}^{102}$. In the Birds Dataset, tf-idf features is in $\mathbb{R}^{7086}$ and after CLSI the final textual features is in $\mathbb{R}^{200}$.

[3] http://plants.usda.gov/java/

[4] http://www.theplantencyclopedia.org/wiki/Main_Page

[5] http://www.bbc.co.uk/science/0/

5.3. Visual features

We used the Classeme features [31] as the visual feature for our experiments since they provide an intermediate semantic representation of the input image. Classeme features are output of a set of classifiers corresponding to a set of C category labels, which are drawn from an appropriate term list defined in [31], and not related to our textual features. For each category $c \in \{1 \cdots C\}$, a set of training images is gathered by issuing a query on the category label to an image search engine. After a set of coarse feature descriptors (Pyramid HOG, GIST, *etc*.) is extracted, a subset of feature dimensions was selected [31], and a one-versus-all classifier ϕ_c is trained for each category. The classifier output is real-valued, and is such that $\phi_c(x) > \phi_c(y)$ implies that x is more similar to class c than y is. Given an image x, the feature vector (descriptor) used to represent it is the classeme vector $[\phi_1(x), \cdots, \phi_C(x)]$. The Classeme feature is of dimensionality 2569.

5.4. Experimental Results

Evaluation Methodology and Metrics: Similar to zero-shot learning literature, we evaluated the performance of an unseen classifier in a one-vs-all setting where the test images of unseen classes are considered to be the positives and the test images from the seen classes are considered to be the negatives. We computed the ROC curve and report the area under that curve (AUC) as a comparative measure of different approaches. In zero-shot learning setting the test data from the seen class are typically very large compared to those from unseen classes. This makes other measures, such as accuracy, useless since high accuracy can be obtained even if all the unseen class test data are wrongly classified; hence we used ROC curves, which are independent of this problem. Five-fold cross validation over the classes were performed, where in each fold 4/5 of the classes are considered as "seen classes" and are used for training and 1/5th of the classes were considered as "unseen classes" where their classifiers are predicted and tested. Within each of these class-folds, the data of the seen classes are further split into training and test sets. The hyper-parameters for the approach were selected through another five-fold cross validation within the class-folds (i.e. the 80% training classes are further split into 5 folds to select the hyper-parameters).
Baselines: Since our work is the first to predict classifiers based on pure textual description, there are no other reported results to compare against. However, we designed three state-of-the-art baselines to compare against, which are designed to be inline with our argument in Sec 3. Namely we used: 1) A Gaussian Process Regressor (GPR) [24], 2) Twin Gaussian Process (TGP) [3] as a structured regression method, 3) Nonlinear Asymmetric Domain Adaptation (DA) [15]. The TGP and DA baselines are of particular importance since our formulation utilizes

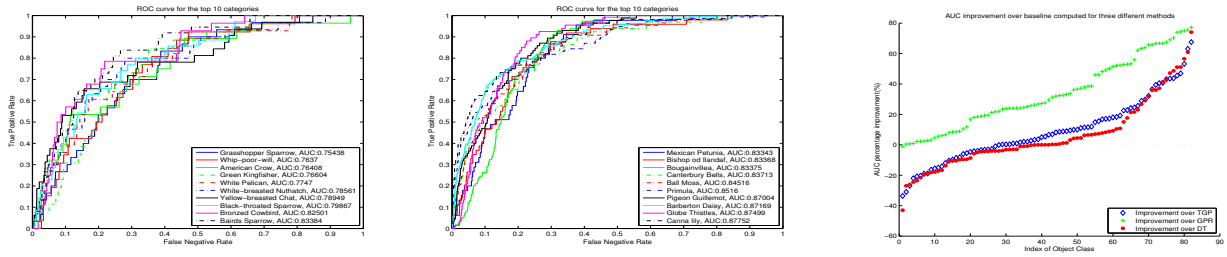

Figure 3: **Left and Middle**: ROC curves of best 10 predicted classes (best seen in color) for Bird and Flower datasets respectively, **Right**: AUC improvement over the three baselines on Flower dataset. The improvement is sorted in an increasing order for each baseline separately

Table 1: Comparative Evaluation on the Flowers and Birds

Approach	Flowers Avg AUC (+/- std)	Birds Avg AUC (+/- std)
GPR	0.54 (+/- 0.02)	0.52 (+/- 0.001)
TGP	0.58 (+/- 0.02)	0.61 (+/- 0.02)
DA	0.62(+/- 0.03)	0.59 (+/- 0.01)
Our Approach	**0.68** (+/- 0.01)	**0.62** (+/- 0.02)

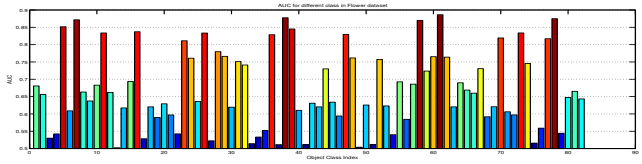

Figure 4: AUC of the predicated classifiers for all classes of the flower datasets

them, so we need to test if the formulation is making any improvement over them. It has to be noted that we also evaluate TGP and DA as alternative formulations that we are proposing for the problem, none of them was used in the same context before.

Results: Table 1 shows the average AUCs for the proposed approach in comparison to the three baselines on both datasets. GPR performed poorly in all classes in both data sets, which was expected since it is not a structure prediction approach. The DA formulation outperformed TGP in the flower dataset but slightly underperformed on the Bird dataset. The proposed approach outperformed all the baselines on both datasets, with significant difference on the flower dataset. It is also clear that the TGP performance was improved on the Bird dataset since it has more classes (more points are used for prediction). Fig 3 shows the ROC curves for our approach on best predicted unseen classes from the Birds dataset on the Left and Flower dataset on the middle. Fig 4 shows the AUC for all the classes on Flower dataset. More results are attached in the supplementary materials.

Fig 3, on the right, shows the improvement over the three baseline for each class, where the improvement is calculated as (our AUC- baseline AUC)/ baseline AUC %. Table 2 shows the percentage of the classes which our approach makes a prediction improvement for each of the three baselines. Table 3 shows the five classes in Flower dataset where our approach made the best average improvement. The point of that table is to show that in these cases both TGP and DA did poorly while our formulation that is based on both of them did significantly better. This shows that our formulation does not simply combine the best of the two approaches but can significantly improve the prediction performance.

Table 2: Percentage of classes that the proposed approach makes an improvement in predicting over the baselines (relative to the total number of classes in each dataset

baseline	Flowers (102) % improvement	Birds (200) % improvement
GPR	100 %	98.31 %
TGP	66 %	51.81 %
DA	54%	56.5%

Table 3: Top-5 classes with highest combined improvement in Flower dataset

class	TGP (AUC)	DA (AUC)	Our (AUC)	% Improv.
2	0.51	0.55	0.83	57%
28	0.52	0.54	0.76	43.5%
26	0.54	0.53	0.76	41.7%
81	0.52	0.82	0.87	37%
37	0.72	0.53	0.83	35.7 %

To evaluate the effect of the constraints in the objective function, we removed the constraints $-(\mathbf{c}^{\mathsf{T}}\mathbf{x}_i) \geq \zeta_i$ which try to enforces all the seen examples to be on the negative side of the predicted classifier hyperplane and evaluated the approach. The result on the flower dataset (using one fold) was reduced to average AUC=0.59 compared to AUC=0.65 with the constraints. Similarly, we evaluated the effect of the constraint $\mathbf{t}_*^{\mathsf{T}}\mathbf{W}\mathbf{c} \geq l$. The result was reduced to average AUC=0.58 compared to AUC=0.65 with the constraint. This illustrates the importance of this constraint in the formulation.

6. Conclusion and Future Work

We explored the problem of predicting visual classifiers from textual description of classes with no training images. We investigated and experimented with different formulations for the problem within the fine-grained categorization

context. We proposed a novel formulation that captures information between the visual and textual domains by involving knowledge transfer from textual features to visual features, which indirectly leads to predicting the visual classifier described by the text. In the future, we are planning to propose a kernel version to tackle the problem instead of using linear classifiers. Furthermore, we will study predicting classifiers from complex-structured textual features.

Acknowledgment This research was partially funded by NSF award IIS-1218872

References

[1] K. Barnard, P. Duygulu, and D. Forsyth. Clustering art. In *CVPR*, 2001. 2

[2] E. Bart and S. Ullman. Cross-generalization: Learning novel classes from a single example by feature replacement. In *CVPR*, 2005. 1, 2

[3] L. Bo and C. Sminchisescu. Twin gaussian processes for structured prediction. *IJCV*, 2010. 5, 6

[4] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *ECCV*. 2010. 1, 2

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[6] L. Duan, D. Xu, I. W.-H. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. *TPAMI*, 2012. 3

[7] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 1, 2

[8] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*. 2010. 3

[9] L. Fe-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *CVPR*, 2003. 1, 2

[10] R. Fergus, H. Bernal, Y. Weiss, and A. Torralba. Semantic label sharing for learning with many categories. In *ECCV*. 2010. 2

[11] M. Fink. Object classification from a single example utilizing class relevance metrics. In *NIPS*, 2004. 2

[12] G. Griffin and P. Perona. Learning and using taxonomies for fast visual categorization. In *CVPR*, 2008. 2

[13] A. E. Hoerl and R. W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 1970. 3

[14] N. Krishnamoorthy, G. Malkarnenkar, R. Mooney, K. Saenko, U. Lowell, and S. Guadarrama. Generating natural-language video descriptions using text-mined knowledge. *NAACL HLT*, 2013. 3

[15] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011. 3, 4, 5, 6

[16] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating simple image descriptions. In *CVPR*, 2011. 3

[17] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by betweenclass attribute transfer. In *CVPR*, 2009. 1, 2

[18] E. G. Miller, N. E. Matsakis, and P. A. Viola. Learning from one example through shared densities on transforms. In *CVPR*, 2000. 2

[19] G. A. Miller. Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM*, 1995. 2

[20] M.-E. Nilsback and A. Zisserman. Automated flower classification over large number of classes. In *ICVGIP*, 2008. 2, 6

[21] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011. 3

[22] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *NIPS*, 2009. 2

[23] D. Parikh and K. Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *CVPR*, 2011. 2

[24] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005. 3, 6

[25] M. Rohrbach, M. Stark, G. Szarvas, and B. Schiele. Combining language sources and robust semantic relatedness for attribute-based knowledge transfer. In *Parts and Attributes Workshop at ECCV*, 2010. 2

[26] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*. 2010. 3

[27] R. Salakhutdinov, A. Torralba, and J. B. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, 2011. 1, 2

[28] B. Saleh, A. Farhadi, and A. Elgammal. Object-centric anomaly detection by attribute-based reasoning. In *CVPR*, 2013. 1

[29] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *IPM*, 1988. 6

[30] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *PAMI*, 2008. 2

[31] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, 2010. 6

[32] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical report, California Institute of Technology, 2010. 2, 6

[33] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *MULTIMEDIA*, 2007. 3

[34] Y. Yang, C. L. Teo, H. Daumé III, and Y. Aloimonos. Corpus-guided sentence generation of natural images. In *EMNLP*, 2011. 3

[35] D. Zeimpekis and E. Gallopoulos. Clsi: A flexible approximation scheme from clustered term-document matrices. In *In SDM*, 2005. 6