– Semantic Transform –
Weakly Supervised Semantic Inference for Relating Visual Attributes

Sukrit Shankar
Machine Intelligence Lab
University of Cambridge
ss965@cam.ac.uk

Joan Lasenby
Comm & SigProc Group
University of Cambridge
jl221@cam.ac.uk

Roberto Cipolla
Machine Intelligence Lab
University of Cambridge
rc10001@cam.ac.uk

Abstract

Relative (comparative) attributes are promising for thematic ranking of visual entities, which also aids in recognition tasks [19, 23]. However, attribute rank learning often requires a substantial amount of relational supervision, which is highly tedious, and apparently impractical for real-world applications.

*In this paper, we introduce the **Semantic Transform**, which under minimal supervision, adaptively finds a semantic feature space along with a class ordering that is related in the best possible way. Such a semantic space is found for every attribute category. To relate the classes under weak supervision, the class ordering needs to be refined according to a cost function in an iterative procedure. This problem is ideally NP-hard, and we thus propose a constrained search tree formulation for the same.*

Driven by the adaptive semantic feature space representation, our model achieves the best results to date for all of the tasks of relative, absolute and zero-shot classification on two popular datasets.

1. Introduction

Visual recognition approaches have conventionally attempted to model visual attributes (thematic properties observable in visual entities like images with human-designated names) [6, 11, 12, 26, 27]. For example, images of natural scenes can have attributes such as *'open'* and *'depth-close'*; while images of faces can have the attributes like *'smiling'* and *'sad'*. A classifier is trained based on the knowledge of the attributes' presence/absence in the training images, and an unseen image is then categorized based on the attributes it has. Inspired by web page ranking formulations, some researchers [19, 23, 14] have recently used relative attributes for visual recognition tasks with considerable success, i.e. instead of the binary knowledge of whether an attribute is present/absent in an image, a real-valued attribute score is considered. For example, instead of an image having a binary attribute of *'smiling'* or *'not smiling'*, the image has a real-valued attribute score which depicts notions like *'more smiling than'* (Fig 1). Such relations between the attributes provide semantically richer image descriptions, and have been shown to be useful for relative, absolute and zero-shot classification tasks [19][1].

The process typically requires a tedious supervision step, where all the given classes need to be related for every attribute category. Using this relational supervision and the training images belonging to each class, *a latent feature space* is learnt, on which the projection of (visual) feature descriptors (of images) produces non-binary attribute scores. For example, if an image I_1 is known to be *'more open'* than an image I_2, for the attribute *'open'*, and the latent feature space is modelled as a 1D weighting curve; the weighting curve should guarantee that I_1 has a greater score than I_2, say 0.5 as to 0.3. Given a test image, its visual feature descriptor is multiplied by the learnt weighting curve to give an attribute score, which then aids in its ranking and classification.

Problem: In a supervised scenario (where the relations between all classes need to be given seeing the training images typically on a per-attribute basis), while the human annotations are helpful, it is a time-consuming task to obtain human annotations for all given attributes in a dataset, more so when it contains videos, documents, etc. Thus, it is important that the class relations are learnt automatically (or with minimum supervision) along with the associated latent feature spaces (per-attribute).

The accuracy of the attribute scores so that they correctly represent the underlying conceptual space, often depends on the model of latent feature (to be learnt) and/or on the availability of a sufficiently diverse set of training data. This can pose a major problem, when it comes to ranking data in a real-world scenario. For instance, consider a situation where for a given attribute, relations between some n types

[1] See Section 3 for a precise definition of these classification types.

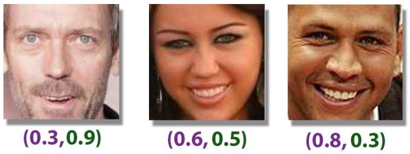


Figure 1: The scores of attributes **(Smiling, White)** on a scale of 0 to 1 for three images. The attribute score is proportional to the strength of attribute presence. The figure is for illustrative purposes.

of classes are known. Let us now have a number of data points (visual entities such as images, videos, etc.) belonging to a new class (unseen in the training data) [2], which we need to rank relative to the known classes. Ranking the data points of this new class is subject to the assumption that the learnt latent feature space encompasses the inherent diversity of the possible set of visual entities that will ever need to be ranked. This can be made true for small datasets with a proper choice of a latent feature model and availability of sufficiently diverse training data, but the procedures do not naturally scale well to the problems occurring with *big data*. With a large amount of complex data, the learnt model is often not able to relate the new classes in an acceptable way.

Ranking entities may also involve some sort of semantic supervision in addition to relational supervision, i.e. instead of only giving relations between all given classes, a rough attribute score (reflecting the underlying theme) on a predefined scale (such as 0 to 1) is also given for the known classes. In such a case, a model is learnt that conforms to the given attribute scores, which then automatically ranks the classes. However, for new classes, the attribute scores again depend on the assumption that the learnt model has encompassed the semantics of all possible data. Also, such scores can be reasonably erroneous and the model needs to take care of such uncertainties.

Proposal and Contributions: While many types of models have been proposed for learning the latent feature space under relational and semantic supervision (e.g. Gaussian process models, Topic Models, Neural Networks, etc.), we aim to answer a more generic question here:

Given a set of training images and a latent feature space model, can we learn this latent space (with minimal supervision) so as to relate all types of possible classes in a semantically best possible way ?[3]

In an attempt to answer the above question, we make the following two significant contributions in this paper:

1. We introduce the **Semantic Transform**, which under minimal supervision, adaptively finds a semantic feature space along with a class ordering that is related in the best possible way.

2. To relate the classes under weak supervision, the class ordering needs to be refined according to a cost function in an iterative procedure. This problem is ideally NP-hard, and we thus propose a constrained search tree formulation for the same.

Our approach is weakly supervised since we assume the semantic attribute score of at least two classes (out of a possible 8 for our datasets) to be known per-attribute. This information is important, since there should be some discriminative information about the attributes (thematic spaces) under consideration. This minimal supervision helps us to find an initial feature space (according to a chosen model), which we then keep adapting in an alternating framework, until the best (maximally separated) class ordering is found out. To learn the relative ordering, we also present an efficient algorithm using a constrained search tree structure, which makes the generic problem of relating classes tractable while achieving acceptable results.

2. Related Work

Binary and Relative Attributes: Learning attribute categories has been shown to be useful to provide cues for object/face recognition [12, 11], zero-shot transfer [12, 21] and part localization [6, 26]. There have been attempts to make the learning and classification tasks more robust, given the categorical attributes. The work of [20] tends to make the binary attributes more discriminative on a class basis.

Recent works [19, 23, 15] have focussed on learning per-attribute class relations for visual recognition tasks, and have shown considerable promise. While authors in [19, 23] assume the availability of relative ordering of classes on a per-attribute basis, [15] assumes the absence of such information. However, the unsupervised method of [15] achieves marginally better results than the supervised counterparts, because of their bottom-up learning structure which is directed more towards selecting some key attributes for classes, rather than ordering classes based on all given attributes.

Rank Learning: The machine learning literature comprises many works which tend to learn ranking models for web pages/documents. The works of [10, 5, 1, 25, 8] all propose different types of probabilistic models for efficient

[2] Note that in real-world situations, broader classes of visual entities are almost always known due to the presence of textual tags. However, attribute-specific information is often not present in the form of textual data with the visual entities, and hence, ranking multimedia entities according to various attributes is largely an unsolved and to an extent a non-targeted problem.

[3] Note that the main requirement is that the classes are ordered so as to correctly reflect the underlying attribute-specific themes and the learnt model maximally separates the classes while conforming to the (minimal) supervision that we have. Also, **Semantic Transform** is all about adaptively learning a feature space that can semantically rank multimedia entities/classes; instead of just transforming the inputs to some sort of static semantic features.

rank learning. While [5] models the uncertainty in the attribute scores with Gaussian processes, [8] does so with the help of probability distributions in a soft-rank methodology. Many methods of ensemble learning also exist in the literature [3, 16, 22, 9]. Researchers have tried to apply such methods (e.g. random forests, kernel max margin learning) [14, 19] in an attempt to learn feature spaces that produce minimally confusing attribute scores. However, the methods are normally highly supervised and the formulations lack a generic structure for adapting to new data.

One of the most famous methodologies for ranking entities based on semantic supervision is the supervised topic model [2]. The approach learns a topic model while trying to maintain the given attribute-specific semantic scores. This automatically ranks the given entities; and for new incoming data, its input features are projected on the learnt topic model to find a new score. This method again falls apart in terms of generality for any type of new class to be related.

Our approach of learning relative ordering (*Semantic Transform*) is in contrast different from that of the aforementioned approaches. We make the use of minimal semantic supervision (rough attribute scores) per-attribute, and then adaptively learn a semantic feature space along with the most plausible class ordering. We are thus able to counter the problem of tedious relational supervision while also adaptively learning a feature space that can exude attribute-specific themes for possibly all types of classes.

3. Approach

We are given a set of training images $I = \{i\}$ represented in $\mathbb{R}^d$ by feature-vectors $\boldsymbol{x_i}$, S image class labels $\{c_s\}$ and a set of M attributes $A = \{a_m\}$. Since we consider a weakly supervised scenario, for each attribute a_m, an ordering of only two class labels is given ($c_{p_m} \prec c_{q_m}$) along with the corresponding response variables ($r_{p_m} < r_{q_m}$). The response variables are given on a normalized scale of 0 to 1.

What exactly are these response variables, and how are they obtained for different types of attributes needs a special mention here. The attributes can be broadly classified as aesthetic and non-aesthetic. Aesthetic attributes normally refer to those which cannot be rated deterministically by all users. For instance, the choice of people for the *interesting* attribute of a movie, song or photo is more or less aesthetic. On the other hand, whether a person is looking angry, sad or happy is a non-aesthetic choice to make, and is expected to be far more deterministic. In case of aesthetic attributes, response variables may refer to normalized average ratings or numerical reviews, such as movie, song ratings, etc. For non-aesthetic attributes, a score on a normalized scale of 0 to 1 can be approximately given by a human, given that humans have an inherent interpolating ability [24], once they roughly know about the possible extremes. For our datasets, the attributes are non-aesthetic in nature.

With the minimal supervision that we consider, our aim is to learn a latent feature space (which conforms to the semantic response variables) along with the best possible class ordering.

3.1. Learning

For each attribute a_m, we need to learn a latent feature space $\boldsymbol{b_m}$ and a relative ordering of the classes $c_s; s = 1, \ldots, S$. Thus, given that $c_{p_m} \prec c_{q_m}$ and corresponding response variables $r_{p_m} < r_{q_m}$, we require that $\forall i \in c_{s_1}, j \in c_{s_2}, c_{s_1} \succ c_{s_2}$,

$$\varphi(\boldsymbol{b_m^T}, \boldsymbol{x_i^{s_1}}) > \varphi(\boldsymbol{b_m^T}, \boldsymbol{x_j^{s_2}}) \tag{1}$$

$$\text{where} \quad \varphi(\boldsymbol{b_m^T}, \boldsymbol{x_i^s}) = \boldsymbol{b_m^T} \boldsymbol{x_i^s} \quad \forall i \in c_s \tag{2}$$

The function $\varphi(.)$ produces a score for the attribute a_m, given the feature space $\boldsymbol{b_m}$ and input feature vector $\boldsymbol{x_i^s}$ of an image belonging to class c_s. For purposes of comparison of our results with those of [19], $\varphi(.)$ is given by Equ (2), and $\boldsymbol{b_m}$ is selected to be a 1D weighting curve, with length equal to the total number of pixels in an input image. Given a test image with its feature vector $\boldsymbol{x_t}$, its score for attribute a_m is given by $\varphi(\boldsymbol{b_m^T}, \boldsymbol{x_t})$.

Following the work of [10], it can be shown that for a given $\varphi(.)$ and relative ordering of classes, the above stated optimization problem can be reduced to the following Support Vector Machine (SVM) classification task: for a trade-off constant C between maximizing the margin (between attribute scores of classes) and satisfying relative ordering, and $\forall i \in c_{s_1}, j \in c_{s_2},\ c_{s_1} \succ c_{s_2}$

$$\underset{\boldsymbol{b_m}}{\arg\min} \quad \left(\frac{1}{2} \parallel \boldsymbol{b_m} \parallel^2 + \ C \sum_{i,j} \xi_{ij}^2 \right) \tag{3}$$

$$\text{s.t.} \quad \varphi(\boldsymbol{b_m^T}, \boldsymbol{x_i^{s_1}}) - \varphi(\boldsymbol{b_m^T}, \boldsymbol{x_j^{s_2}}) > 1 - \xi_{ij}, \quad \xi_{ij} \geq 0 \tag{4}$$

Since we have the semantic response variables (r_{p_m}, r_{q_m}), we make an initial estimate of the latent feature space $\boldsymbol{b_m}$ by solving the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{b_m}}{\arg\min} \ \Bigg(\left| \frac{\sum \ \varphi(\boldsymbol{b_m^T}, \boldsymbol{x_i^{p_m}})}{n_{p_m}} - r_{p_m} \right| \\ + \left| \frac{\sum \ \varphi(\boldsymbol{b_m^T}, \boldsymbol{x_i^{q_m}})}{n_{q_m}} - r_{q_m} \right| \Bigg) \end{aligned} \tag{5}$$

where n_{p_m} is the number of training images in class c_{p_m} and n_{q_m} in class c_{q_m}.

With different latent feature space models, the optimization problem of Equ (5) can be intractable. We thus use Genetic Algorithms [7] (a class of evolutionary algorithms)[4] to solve it. For q real numbers in the 1D weighting curve defining $\boldsymbol{b_m}$, each real number is encoded as a string of $\lceil \log_2 q \rceil$ bits, and thus an individual in the population is a string of $q\lceil \log_2 q \rceil$ bits. Since, genetic algorithms can sometimes converge to a local minimum instead of the desired global one, we give multiple runs with different mutation rates, and accept the solution that occurs the most number of times. To avoid trivial solutions, we put sparsity constrains on the learnt solution. The use of an evolutionary algorithm procedure here can serve as a major advantage for building a generic framework with the proposed *Semantic Transform*, especially when one aims to learn a neural network structure [28] in order to have an initial estimate of $\boldsymbol{b_m}$. When complex structures are involved, estimation of distribution algorithms (EDAs) [13] can also be used as evolutionary procedures.

Once we have the initial estimate of our latent feature space, we perturb it to a limited extent so as to approximately conform to the semantic scores (obtained from supervision), while also estimating the most plausible relative ordering of the classes. This is done in an alternating iterative procedure. Perturbation not only helps to make the initially learnt feature space adaptive, but also takes into account the inaccuracy inherent in the response variables. For a given relative ordering of classes and a perturbation model $\boldsymbol{\Delta_m}$, the equations (3) and (4) are modified as follows ($\forall i \in c_{s_1}, j \in c_{s_2},\ c_{s_1} \succ c_{s_2}$):

$$\arg\min_{\boldsymbol{\Delta_m}} \Bigg(\frac{1}{2} \parallel \boldsymbol{\Delta_m} \parallel^2 + \ C_1 \sum_{i,j} \xi_{ij}^2 \\ + \ C_2 \sum_{i \in c_{p_m}; j \in c_{q_m}} \Big[\varphi((\boldsymbol{b_m^T} + \boldsymbol{\Delta_m^T}), \boldsymbol{x_i^{p_m}}) \\ - \varphi((\boldsymbol{b_m^T} + \boldsymbol{\Delta_m^T}), \boldsymbol{x_j^{q_m}}) - (r_{p_m} - r_{q_m}) - \delta_m \Big]^2 \Bigg) \quad (6)$$

$$\text{s.t.} \quad \varphi((\boldsymbol{b_m^T} + \boldsymbol{\Delta_m^T}), \boldsymbol{x_i^{s_1}}) - \varphi((\boldsymbol{b_m^T} + \boldsymbol{\Delta_m^T}), \boldsymbol{x_j^{s_2}}) \\ > 1 - \xi_{ij}, \quad \xi_{ij} \geq 0, \quad C_2 > C_1 \quad (7)$$

Note that the perturbation model $\boldsymbol{\Delta_m}$ is also chosen to be a 1D weighting curve and of the same length as that of $\boldsymbol{b_m}$, and an addition operation is simply chosen to incorporate the perturbation. This is done since we wanted our finally adapted latent feature space to be a 1D weighting curve, in order to draw fair comparison of our results with those of [19]. In a general case, the perturbation model can be made different from $\boldsymbol{b_m}$ and can even encompass modification of network structures involved in $\boldsymbol{b_m}$. Following [10], the optimization problem posed by equations (6) and (7) is solved by the method of [4]. The term weighted by C_2 in equation (6) is important since it limits the perturbation so as to conform to the response variables obtained from supervision. Thus, $C_2 > C_1$, and the parameter δ_m limits the amount of change allowed in the semantic response variables. For our experiments, $C_1 = 0.1, C_2 = 0.5, \delta_m = 0.1$.

Now, we require that along with the weighting curve $\boldsymbol{b_m}$ and the perturbation model $\boldsymbol{\Delta_m}$, the relative ordering constrained by $c_{p_m} \prec c_{q_m}$ is also learnt. We follow an alternating optimization approach, where we fix a class ordering, and adapt the feature space, and then with this adapted feature space, we refine the relative ordering of the classes. This continues until convergence occurs. Our full model listed in Algorithm 1.

The alternating procedure stated above is feasible only when there is some efficient algorithm for refining the relative ordering of classes. This problem in itself is NP-hard since the class ordering needs to be refined subject to a cost function. We thus propose a top-down greedy algorithm with constraints on a search tree for estimating relative ordering. To provide an example of our approach, let us consider that we have 7 classes ($1-7$) which need to be ordered. Let the initial ordering be $1 \prec 2 \prec 3 \prec 4 \prec 5 \prec 6 \prec 7$. Then, a swapping of 1 and 4 alters the maximum number of relative orders, since all of $2, 3$ will change their order relative to 1 and 4[5]. A similar argument holds, if 4 and 7 are swapped. However, if 4 and 5 are swapped or 2 and 3 are swapped, other relative orders remain unchanged. For this, we form a search tree as shown in Fig 2, with each node as the class number, and root being the middle class. For a given node, all possible swappings with all its children are considered. Thus, while node 4 can have 6 possible alterations spanning all classes, node 2 can only have 2 possible alterations, with only 1 and 3 getting affected. Each swapping of the classes affects the loss function (encoding maximal separation between classes for a relative ordering) of Eqn (8). Since, we are given a prior ordering as $c_{p_m} \prec c_{q_m}$, we constrain the aforementioned swappings such that $c_{p_m} \prec c_{q_m}$ is always satisfied. Algorithmically, if $2 \prec 4$ is given, we will not swap 4 with 2 or any of its left children, i.e. 1 (Fig 2). Algorithm 2 formally sets out the method.

This approach is greedy, but in a reasonable number of steps ($\mathcal{O}(s\log(s))$ as compared to $\mathcal{O}(s!)$ for s classes), it

[4] Genetic Algorithms may not be the best choice to an optimization problem, in case an exact solution procedure for the problem is known. However, they can give a nearly correct estimate of the intended solution. This suffices for our case, since we are any ways going to refine this estimation later on.

[5] The greedy algorithm considers swapping with only its parents (except when the parents are at the penultimate level of the tree, i.e. theit immediate children are the leaf nodes). In this context, 1 to 4 changes maximum relative orders. Also, note that a swapping between 1 and 7 can eventually get done with (1-3,1-4,1-7,3-7) across iterations.

helps to achieve the most plausible relative ordering. The major reason for its success is the top-down approach (taking the middle class as the root node), since that ensures that we prioritize the changes in the order of that class, which can possibly affect the loss function to the maximum extent, and then we refine the other orders.

Algorithm 1 Semantic Transform (ST)

```
1. Initialize relative ordering for $S$ classes $c_s$ using $\boldsymbol{b_m}$ estimated from equation (5). This will automatically satisfy the constraint $c_{p_m} \prec c_{q_m}$.
2. Update $\boldsymbol{\Delta_m}$ using equations (6) and (7).
3. Update the relative order for $c_s$ using Algorithm 2. If the updated relative order is the same as previous one (convergence), go to Step (4); else go to Step (2).
4. The last ordering learnt is the final class ordering, and $\boldsymbol{b_m} + \boldsymbol{\Delta_m}$ is the adaptively learnt feature space.
```

Algorithm 2 Relative Ordering Estimation (ROE)

```
1. For given $\boldsymbol{\Delta_m}$ and $\boldsymbol{b_m}$, and initial relative ordering of $S$ classes $c_{s_1} \prec c_{s_2} \prec c_{s_3} \cdots \prec c_{s_S}$, form a search tree with the root node as $s_{\lfloor S/2 \rfloor}$.
2. Let $L$ denote a level of the tree. Initialize $L = 1$ (root node).
3. At a level $L$, let each (parent) node be $pa_k; k = 1, \ldots, K$ of the tree, all its children be $ch_{r_k}; r_k = 1, \ldots, R_k$, and all left children be $lch_{e_k}; e_k = 1, \ldots, E_k$. Note that $lch_{(.)} \subset ch_{(.)}$. Let a swapping of $pa_k$ with $ch_{r_k}$ be denoted by $v_{k,r_k} = \{pa_k \leftrightarrow ch_{r_k}\}$. Consider all $v_{k,r_k}, k = 1, \ldots, K$ such that $\forall pa_k = q_m, lch_{e_k} \neq p_m$ & parent $(lch_{e_k}) \neq p_m$, and find $r_k$ which $\forall i \in c_{s_1}, j \in c_{s_2},\ c_{s_1} \succ c_{s_2}; s_1, s_2 = 1, \ldots S_s$ minimizes

   $$C_1 \sum \left(\varphi((\boldsymbol{b_m^T} + \boldsymbol{\Delta_m^T}), \boldsymbol{x_i^{s_1}}) - \varphi((\boldsymbol{b_m^T} + \boldsymbol{\Delta_m^T}), \boldsymbol{x_j^{s_2}})\right)^2 + C_2 \sum_{i \in c_{p_m}; j \in c_{q_m}} \Big[\varphi((\boldsymbol{b_m^T} + \boldsymbol{\Delta_m^T}), \boldsymbol{x_i^{p_m}}) - \varphi((\boldsymbol{b_m^T} + \boldsymbol{\Delta_m^T}), \boldsymbol{x_j^{q_m}}) - (r_{p_m} - r_{q_m}) - \delta_m\Big]^2 \quad (8)$$

   If $L$ happens to be the penultimate level in the tree, also allow swapping between the children of a parent node.
4. Increment $L$ and go to Step (3), and stop if the last level of the tree is reached.
```

3.2. Relative Classification

Relative classification refers to the task of categorizing a test image relative to any two given training images. Given a test image I_t and its associated score for a given attribute a_m, we randomly select one image I_l from the training set whose attribute score is less than that of I_t, and an image I_u whose attribute score is greater than that of I_t. We then see the classes of I_t, I_l and I_u as c_t, c_l and c_u respectively. If for attribute a_m, $c_l \prec c_t \prec c_u$, the relative classification is correct, else not.

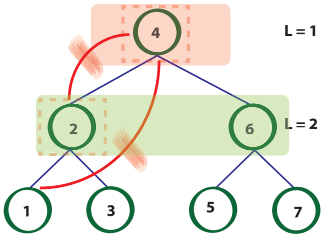


Figure 2: An illustration of the tree structure of classes ordered as $1 \prec 2 \prec 3 \prec 4 \prec 5 \prec 6 \prec 7$ constrained by $2 \prec 4$. It aids in estimation of the relative ordering according to Algorithm 2. Since $2 \prec 4$, swappings $4 \leftrightarrow 2$ and $4 \leftrightarrow 1$ are not allowed for this tree structure. *Best viewed in color.*

3.3. Absolute Classification

Absolute classification refers to the task of categorizing a test image to its correct class. For this we form a generative model for each class c_s in $\mathbb{R}^M$ similar to [19], once the attribute scores for the training images are obtained and the best relative ordering of classes are learnt. We use a Gaussian distribution and estimate the mean $\boldsymbol{\mu_s} \in \mathbb{R}^M$ and $M \times M$ covariance matrix Σ_s from the attribute scores of the training images from class c_s, so that each class c_s is represented by $\mathcal{N}(\boldsymbol{\mu_s}, \Sigma_s); s = 1, \ldots, S$ for S classes. Given a test image with the feature vector $\boldsymbol{x}_t$, it is assigned to a class c_{s*} that produces the highest likelihood:

$$s* = \underset{s \in 1, \ldots, S}{\arg\max}\ P(\boldsymbol{x}_t | \boldsymbol{\mu_s}, \Sigma_s) \quad (9)$$

3.4. Zero-shot Classification

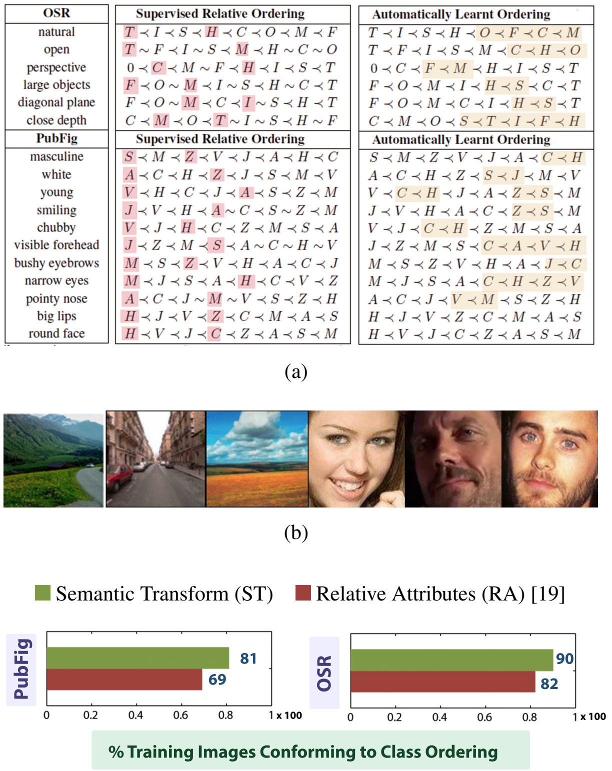
For zero-shot classification [19], we have S_s number of seen classes and S_u number of unseen classes from a total of S classes. For all the seen classes S_s, we learn the relative ordering as specified in Sec 3.1. Similar to [19], we assume that the relative ordering of the unseen class is known with respect to the seen classes. Then, a generative model for the unseen classes can be built using that of the seen classes [19]. The generative model for the seen classes is built according to the procedure specified in Sec 3.3. For two seen classes c_{s_1} and c_{s_2} and given an attribute a_m, the parameters of the generative model are specified as follows [19]: if $c_{s_1} \prec c_u \prec c_{s_2}$, $\boldsymbol{\mu_{u,m}} = \frac{1}{2}(\boldsymbol{\mu_{s_1,m}} + \boldsymbol{\mu_{s_2,m}})$; if $c_u \prec c_{s_1}$, $\boldsymbol{\mu_{u,m}} = \boldsymbol{\mu_{s_1,m}} - d_m$ where d_m is the average distance between the sorted mean attribute scores for all seen classes corresponding to attribute a_m; if $c_u \succ c_{s_2}$, $\boldsymbol{\mu_{u,m}} = \boldsymbol{\mu_{s_2,m}} + d_m$. In all cases, $\Sigma_u = \frac{1}{S_s} \sum_{s=1}^{S_s} (\Sigma_s)$. If an attribute a_m is not used to define an unseen class c_u, μ_{u_m} is taken to be the mean of the scores of all training images for attribute a_m, and the m^{th} diagonal entry of $\Sigma_u((\Sigma_u)_{m,m})$ to be the variance of the same. With the generative model estimated for the unseen classes, the classification can be done using Eqn (9).

4. Results and Discussion

We evaluate our approach on the datasets of faces and natural scenes. For natural scenes, we consider the **Outdoor Scene Recognition (OSR) Dataset** [17] containing 2688 images from 8 classes. For faces, we consider the **Public Figure Face Database (PubFig)** [11] containing 800 images from 8 random identities with 100 images each. Each of those identities corresponds to a class. Similar to [19, 15], we use the 512-dimensional gist [18] descriptor as our visual image features for the *OSR* dataset, while we use a concatenation of the gist descriptor and a 45-dimensional Lab color histogram as our visual feature descriptors for the *PubFig* Dataset. Some images from the *OSR* and the *PubFig* datasets are shown in Fig 3(b).

For each dataset, we use 240 images for training, with 30 images from each class. Note that both the datasets have 8 classes (categories) as shown in Fig 3(a). The rest of the images are used for testing. For relative and absolute classification tasks, all of the classes are seen, i.e. for all the 8 classes, 30 images are used for training.

Our approach is a weakly supervised one, i.e. the relative ordering of the classes for each attribute are not known, but instead need to be learnt using minimal prior relative information. We compare our automatically learnt class relations with the human annotated ones of [19] in Fig 3(a). It can be seen that our learnt class relations are normally the same as that of the human-annotated ones, except for closely related or similarly related classes. This is justifiable, since the closely related human-annotated classes can go either ways if more training data is available. Also, since our learning approach uses semantic supervision instead of just the relational supervision, our learnt weighting curve is semantically more correct than the one learnt with [19]. The weighting curve of [19] does not know what should roughly be the semantic score of a class for a given attribute, unlike ours where we know for atleast 2 classes per-attribute. To be more precise, note that for a given/learnt relative ordering, the condition specified by equation (1) should hold true for all training images. However, in reality, this is not the case. For relational supervision as considered in [19], the number of times the relations are satisfied by the training images are considerably lesser than what we get with our approach. Fig 3(c) depicts the numbers. Thus, for the same latent feature space model, our approach is able to produce better classification. Fig 5(a), 5(b) shows how this affects the relative separation of the classes, and the variance of the attribute scores of the training images. The effectiveness of this notion is depicted through the empirical results presented in the subsequent sections. Note that in Fig 3(a), the red shaded areas (under *Supervised Relative Ordering*) represent those classes (per-attribute) that were considered for supervision. There were only two classes chosen (per-attribute) for supervision. Moreover, the classes were so chosen that for any attribute, they were neither the same, nor closely related, in order to have meaningful and discriminative semantic response variables r_{p_m}, r_{q_m}. The response variables are set on a scale of $(0, 1)$ as follows: For 8 classes, in each dataset, we assume equally spaced attribute scores according to the *Superlative Relative Ordering*. These are rough scores that we set for our experiments, but note that we only use response variables for c_{p_m}, c_{q_m}.



OSR	Supervised Relative Ordering	Automatically Learnt Ordering
natural	$T \prec I \prec S \prec H \prec C \prec O \prec M \prec F$	$T \prec I \prec S \prec H \prec O \prec F \prec C \prec M$
open	$T \sim F \prec I \sim S \prec M \prec H \sim C \sim O$	$T \prec F \prec I \prec S \prec M \prec C \prec H \prec O$
perspective	$0 \prec C \prec M \sim F \prec H \prec I \prec S \prec T$	$0 \prec C \prec F \prec M \prec H \prec I \prec S \prec T$
large objects	$F \prec O \sim M \prec I \sim S \prec H \sim C \prec T$	$F \prec O \prec M \prec I \prec H \prec S \prec C \prec T$
diagonal plane	$F \prec O \sim M \prec C \prec I \sim S \prec H \prec T$	$F \prec O \prec M \prec C \prec I \prec H \prec S \prec T$
close depth	$C \prec M \prec O \prec T \sim I \sim S \prec H \sim F$	$C \prec M \prec O \prec S \prec T \prec I \prec F \prec H$
PubFig	**Supervised Relative Ordering**	**Automatically Learnt Ordering**
masculine	$S \prec M \prec Z \prec V \prec J \prec A \prec H \prec C$	$S \prec M \prec Z \prec V \prec J \prec A \prec C \prec H$
white	$A \prec C \prec H \prec Z \prec J \prec S \prec M \prec V$	$A \prec C \prec H \prec Z \prec S \prec J \prec M \prec V$
young	$V \prec H \prec C \prec J \prec A \prec S \prec Z \prec M$	$V \prec C \prec H \prec J \prec A \prec Z \prec S \prec M$
smiling	$J \prec V \prec H \prec A \sim C \prec S \sim Z \prec M$	$J \prec V \prec H \prec A \prec C \prec Z \prec S \prec M$
chubby	$V \prec J \prec H \prec C \prec Z \prec M \prec S \prec A$	$V \prec J \prec C \prec H \prec Z \prec M \prec S \prec A$
visible forehead	$J \prec Z \prec M \prec S \prec A \sim C \sim H \sim V$	$J \prec Z \prec M \prec S \prec C \prec A \prec V \prec H$
bushy eyebrows	$M \prec S \prec Z \prec V \prec H \prec A \prec C \prec J$	$M \prec S \prec Z \prec V \prec H \prec A \prec J \prec C$
narrow eyes	$M \prec J \prec S \prec A \prec H \prec C \prec V \prec Z$	$M \prec J \prec S \prec A \prec C \prec H \prec Z \prec V$
pointy nose	$A \prec C \prec J \sim M \sim V \prec S \prec Z \prec H$	$A \prec C \prec J \prec V \prec M \prec S \prec Z \prec H$
big lips	$H \prec J \prec V \prec Z \prec C \prec M \prec A \prec S$	$H \prec J \prec V \prec Z \prec C \prec M \prec A \prec S$
round face	$H \prec V \prec J \prec C \prec Z \prec A \prec S \prec M$	$H \prec V \prec J \prec C \prec Z \prec A \prec S \prec M$

(a)

(b)

Semantic Transform (ST)
Relative Attributes (RA) [19]
PubFig
81
69
OSR
90
82
0
0.2
0.4
0.6
0.8
1 x 100
% Training Images Conforming to Class Ordering

(c)

Figure 3: (a) **Automatically Learnt Relative Ordering with our ST approach.** The *OSR* dataset includes images from the classes (categories): coast(C), forest (F), highway (H), inside-city (I), mountain (M), open-country(O), street (S) and tall-building (T). The *PubFig* dataset includes images of classes: Alex Rodriguez (A), Clive Owen (C), Hugh Laurie (H), Jared Leto (J), Miley Cyrus (M), Scarlett Johansson (S), Viggo Mortensen (V) and Zac Efron (Z). *The yellow shaded areas represent differences in our learnt ordering as compared to [19].* The red shaded areas are the classes c_{p_m}, c_{q_m} (per-attribute) that were considered for supervision. (b) **An illustration of some of the images from the *OSR* and the *PubFig* datasets.** While the *OSR* dataset contains the images of natural scenes, *PubFig* dataset consists primarily of people's faces. (c) **Mean (taken across attributes) percentage of the number of training images conforming to the supervised/learnt class ordering** for RA and ST. ST performs better since the weighting curves encompass semantic information. *Best viewed in color and with zoom.*

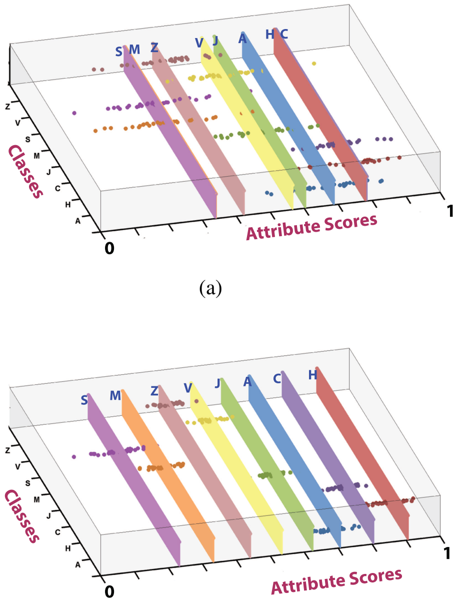
Relative Classification: We follow the relative classification procedure as stated in Sec 3.2. We compare the results of our ST approach outlined in Sec 3 with the supervised relative attribute learning (RA) approach of [19] and supervised relative attribute forest (RF) learning approach of [14]. The results with the mean recognition scores are presented for both the *OSR* and the *PubFig* datasets in Fig 4(a). It can be seen that our approach gives the best results as compared to the the state-of-the-art methods which consider relative attributes. Note that unlike [19, 14], we also learn the relative ordering of classes in conjunction with the weighting curves.

One of the critical points to consider here is that how does one choose I_l and I_u (Sec 3.2)? Typically, it is suggested that one should choose I_l and I_u such that the attribute scores for them are not too far nor too close to that of I_t. The previous works of [19, 14] do not give any account of the relative separation of the attribute scores while doing classification on a relative basis. Utilizing the code of [19], we see that the authors have considered a relative separation of around 0.75 if the attribute scores for all of the training images (corresponding to the seen classes) are within a normalized range of 0 to 1. It is reasonable to assume that the authors of [14] have assumed a similar separation scale. All the results presented in Fig 4(a) are on a separation scale of 0.75.

A separation scale of 0.75 is somewhat large for relative classification, since a lot of attribute scores tend to fall within this scale. We evaluate our ST approach with varying separation scales (within a reasonable range), and compare it with the corresponding results for the work of [19]. The mean relative classification values for both the *OSR* and the *PubFig* datasets are shown for varying separation scales in Fig 4(d). While the classification accuracy for [19] falls off significantly with lower separation scales, our model proves robust in the same scenario. This can be attributed to the fact that weighting curves learnt from ST have semantic information embedded in them.

Absolute Classification: This refers to the task of assigning a test image to its correct class. The classification procedure used is the one outlined in Sec 3.3. The mean recognition results are presented in Fig 4(b) for our ST approach, for the approach of [19] (RA), the unsupervised relative attribute (UR) learning approach of [15] and the approach of [19] (RF). It can be seen that we significantly outperform the other methods for the same reasons as outlined in the previous section.

Zero-shot classification: This is the task of class-labelling a test image when no images of that class have been used for training. For zero-shot learning, out of 8 classes per dataset, we normally choose 2 unseen classes and consider the remaining 6 classes as seen. We choose 20 such seen/unseen combinations randomly for each dataset. For the seen classes, we use 30 images per class for training. During testing, we use all of the images of the unseen classes. The generative model outlined in Sec 3.4 is used for zero-shot classification. We present the results of our ST approach, the RA [19] method and the UR model of [15]. The average recognition scores for both the datasets are presented in Fig 4(c).



(a)

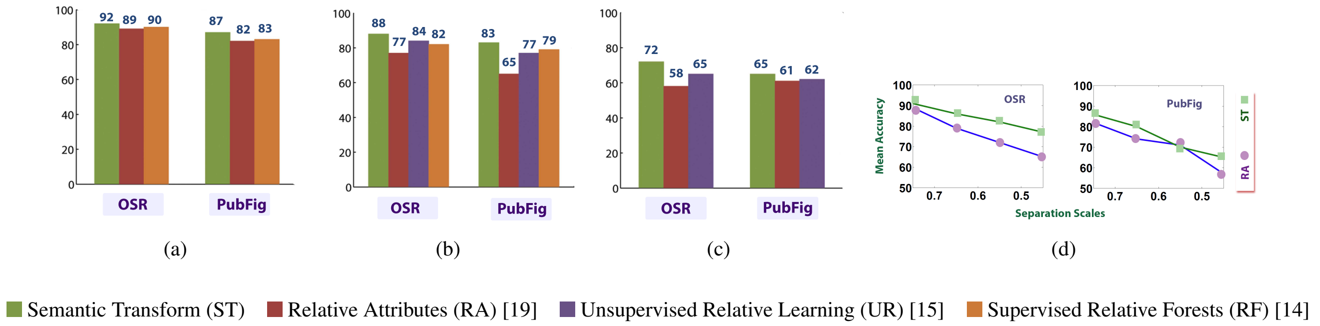
(b)

Figure 5: (a) The attribute scores (for training images) with per-class means using [19] for a human-annotated relative order $S \prec M \prec Z \prec V \prec J \prec A \prec H \prec C$. The classes S, M and H, C have confusing scores which affects the classification accuracy. (b) The attribute scores (with per-class means) using our approach (ST). The classes are now well separated with our optimization procedure. The learnt relative order $S \prec M \prec Z \prec V \prec J \prec A \prec C \prec H$ is slightly different from the human-annotated one (C and H swapped), but only differs for closely related classes to achieve maximal separability, as optimized by the adapted weighting curve inherently encompassing some semantic information. The results are simulated for the attribute *masculine* on *PubFig* Dataset. For a full description of classes and attributes, please see caption of Fig 3(a). *Best viewed in color and with zoom.*

It is clear from Fig 4(c) that our *Semantic Transform* procedure outperforms other recent methods for zero-shot classification. The better performance of ST can be attributed to reasons similar to those as in the previous subsections.

5. Conclusions and Future Work

We have introduced the **Semantic Transform**, which under minimal supervision, can adaptively find a semantic feature space along with a class ordering that is related in the best possible way. To relate the classes under weak supervision, the class ordering needs to be refined according to a cost function in an iterative procedure, for which we have proposed a constrained search tree formulation. We have shown that *Semantic Transform* has the ability to learn a given feature space model in a more semantically correct



(a)

(b)

(c)

(d)

Figure 4: (a) **Mean Relative Classification Accuracy** with ST, RF, RA. (b) **Mean Absolute Classification Accuracy** for ST, RF, UR, RA. (c) **Mean Zero-Shot Classification Accuracy** for ST, UR and RA. In all cases, ST performs the best. (d) **Mean Relative Classification Accuracy with varying separation scales** for RA (blue circle), ST (green square). Our ST approach shows better stability. *Best viewed in color.*

way, which has led to results better than the state-of-the-art for all of the tasks of relative, absolute and zero-shot classification on two popular datasets.

The notion of *Semantic Transform* has many future research avenues. Various latent space feature models with different types of perturbation models (including the ones related to neural networks and topic models) can be tried for big datasets. Use of advanced evolutionary algorithms with an analysis of the amount of supervision required is a plausible research area. *Very broadly, with Semantic Transform, we aim to find a latent feature space (with limited/feasible supervision) for all types of multimedia entities on the web, through which the presently available data classes can be efficiently ranked; and with new data class additions, the feature space can keep evolving naturally.*

References

[1] T. Bernecker, H.-P. Kriegel, M. Renz, and A. Zuefle. Probabilistic ranking in uncertain vector spaces. In *Database Systems for Advanced Applications*, 2009.

[2] D. M. Blei and J. D. McAuliffe. Supervised topic models. *arXiv preprint arXiv:1003.0783*, 2010.

[3] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5), 2007.

[5] W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. In *JMLR*, pages 1019–1041, 2005.

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

[7] D. E. Goldberg and J. H. Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[8] J. Guiver and E. Snelson. Learning to rank with softrank and gaussian processes. In *ACM SIGIR*, 2008.

[9] M. A. Hearst, S. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4), 1998.

[10] T. Joachims. Optimizing search engines using clickthrough data. In *ACM SIGKDD*, 2002.

[11] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009.

[12] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[13] P. Larrañaga and J. A. Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*, volume 2. Springer, 2002.

[14] S. Li, S. Shan, and X. Chen. Relative forest for attribute prediction. In *ACCV*, 2012.

[15] S. Ma, S. Sclaroff, and N. Ikizler-Cinbis. Unsupervised learning of discriminative relative visual attributes. In *ECCV Workshop on Parts and Attributes*, 2012.

[16] C. Z. Mooney, R. D. Duval, and R. Duvall. *Bootstrapping: A nonparametric approach to statistical inference*. Number 94-95. Sage, 1993.

[17] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3), 2001.

[18] A. Oliva, A. Torralba, et al. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155, 2006.

[19] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.

[20] M. Rastegari, A. Farhadi, and D. Forsyth. Attribute discovery via predictable discriminative binary codes. In *ECCV*. 2012.

[21] O. Russakovsky and L. Fei-Fei. Attribute learning in large-scale datasets. In *Trends and Topics in Computer Vision*. 2012.

[22] B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization and beyond*. the MIT Press, 2002.

[23] A. Shrivastava, S. Singh, and A. Gupta. Constrained semi-supervised learning using attributes and comparative attributes. In *ECCV*. 2012.

[24] P. Sinha. Recognizing complex patterns. *Nature Neuroscience*, 5:1093–1097, 2002.

[25] M. A. Soliman and I. F. Ilyas. Ranking with uncertain scores. In *ICDE*, 2009.

[26] G. Wang and D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *ICCV*, 2009.

[27] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. In *ECCV*, 2010.

[28] X. Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999.