Learning Universal Embeddings from Attributes

Ishan Nigam*, Chen Huang*, Deva Ramanan

Abstract

We address the problem of learning a universal embedding space from which different semantic concepts or notions of similarity can originate. Contemporary attribute datasets provide rich multi-label annotations to achieve this goal. Universal embeddings learned from the multi-label attributes would naturally encourage feature sharing, leading to reduced feature redundancy and boosted generalization ability. This paper presents a multi-task framework to learn universal embeddings by mapping them to different subspaces, each corresponds to one particular attribute and is supervised by the triplet similarity. A weighted triplet loss is proposed to combine all the possible triplets for all attributes in a small batch, with weights designed to favor triplets with hard examples by computing their attribute set similarity. This not only eliminates the need for hard triplet mining in batch, and makes the most of available data in a simple way, but also helps to learn the structure of the private and shared features in universal space to encode different triplet similarities across attributes. The learning is made more robust by competing against an adversarial network that perturbs the universal embeddings to increase the loss. The resulting Weighted Triplet-induced Adversarial Universal Embedding (WT-AUE) achieves strong results for attribute prediction, low-shot generalization as well as off-task recognition.

Introduction

Deep networks have revolutionized visual understanding through the ability to learn data-driven representations. Such representations are typically trained for a particular task with massive amounts of labeled data (Deng et al. 2009).

Learning embeddings: A particularly influential variant of representation learning is the problem of learning embedding spaces, which transform input images into a finite-dimensional vector space where images with similar labels are closeby according to some distance (Salakhutdinov and Hinton 2007). Directly learning embeddings has many advantages. Firstly, they are not be tied to category labels, and so can be trained on pairs (or triples) of examples with same/different tags (Hadsell, Chopra, and LeCun 2006). Secondly, they can be used for zero-shot or one-shot learning by applying the learned distance functions to novel class exemplars (Schroff, Kalenichenko, and Philbin 2015). However, embeddings do have limitations. They assume that a single universal notion of distance always holds. But this is not always true. Consider the problem of learning an embedding of faces that is attribute-specific (Fig. 1). A gender-specific embedding will likely need to be invariant to the age attribute, while an age-specific embedding will likely need to be invariant to gender. This suggests it may be hopeless to learn a single “one-size-fits-all” embedding, since an embedding that is invariant to both age and gender seems useless for either task.

Multi-task learning: To reconcile the apparent contradiction above, we appeal to the large literature on multi-task learning (Caruana 1998). Recent work has shown that a shared set of nonlinear features can be jointly learned to serve a variety of different tasks. Examples include joint learning for facial landmark detection, head pose and facial attribute prediction (Zhang et al. 2014b), training a universal Convolutional Neural Network (CNN) for low-, mid-, and high-level vision tasks (Kokkinos 2017), etc. Multi-task networks have also been shown to transfer better to novel tasks because the shared features appear to be more generic (Parisotto, Ba, and Salakhutdinov 2015). From our perspective, the shared feature representation can be thought as as an embedding that is linearly projected into (one or more) one-dimensional spaces representing classifiers for one or more tasks.

Multi-task embeddings: Our formulation can be seen as the natural integration of multi-task learning and non-
linear embeddings: we learn a single universal embedding space that is linearly projected into attribute-specific subspaces. Our overall philosophy is similar to multi-view embeddings (Amid and Ukkonen 2015) and the recent work of conditional similarity networks (CSNs) (Veit, Belongie, and Karaletsos 2017). However, there are several important differences. The former learns separate embeddings without any multi-task sharing, while the latter learns a single universal embedding that is element-wise masked out to produce different embeddings capturing different notions of similarity. In some sense, CSNs learn axis-aligned embedding subspaces, while we learn arbitrary linear projections (consistent with common best-practices in multi-task learning). We demonstrate that our universal embedding space can be used for few-shot learning of attribute-specific embeddings. This has the practical advantage that data (e.g., faces) can be stored using a fixed universal embedding representation that will likely apply to any future attribute of interest.

Training: We also contribute several algorithmic innovations to the problem of learning embeddings specific to our multi-task setting. As in much recent work, we learn with triples of examples, but use multiple attribute labels to speed up convergence and increase accuracy in three distinct ways: (1) When learning on mini-batches, we project each triplet to every attribute-specific subspace, effectively learning from multiple supervisory signals per batch. (2) Unlike the sampling techniques discussed in recent work (Wu et al. 2017), we exhaustively use all triplets in batch. We also find that softly weighting triplets improves convergence. We use multi-view attribute labels to define a notion of semantic similarity between examples, which is in turn used to efficiently construct a meaningful weighting. The weights particularly favor those hard examples, acting as a soft tool of hard triplet mining. (3) We introduce an adversarial network that perturbs the universal embeddings so as to generate harder triplets in each subspace.

Related Work

Multi-Task Embedding Learning

Our universal embedding is learned in a multi-task framework. Example multi-task learning methods learn CNN features for joint facial landmark detection, head pose and facial attribute prediction (Zhang et al. 2014b), or for low-, mid-, and high-level vision tasks (Kokkinos 2017). Another line of multi-task embedding methods learn from multi-label attribute annotations. For example, (Reed et al. 2014) proposed a higher-order Boltzmann machine to disentangle attributes from a manifold with multiplicative interactions. Others use CNN to either learn a single feature embedding with independent attribute classifiers (Liu et al. 2015), or directly learn multiple attribute-specific subspaces (Amid and Ukkonen 2015). However, a single feature embedding cannot capture the contradictory notions of similarity in different attributes. Direct subspace learning will result in high feature redundancy without feature sharing mechanisms. Our method overcomes these issues by jointly learning a universal embedding and the associated subspaces.

Recent methods improve by combining the attribute constraints with a classification loss (Zhang et al. 2016) or even human expertise (Wilber et al. 2015). One closely-related method to ours, called conditional similarity network (Veit, Belongie, and Karaletsos 2017), learns masks on top of a shared embedding to disentangle latent subspaces for different attributes. However their subspaces are independently learned in an axis-aligned manner, which will be shown to underperform our joint learning of arbitrary projections.

Deep Embedding Learning and Data Sampling

Deep embedding methods typically map images into an embedding space, where their distances preserve the relative similarity. The contrastive loss (Hadsell, Chopra, and LeCun 2006) and triplet loss (Schroff, Kalenichenko, and Philbin 2015) are two popular embedding methods, both striving for a similar objective: to bring the same-class images close together while pushing away inter-class images. Contrastive loss chooses to enforce an absolute distance margin to separate the positive image pairs from negative pairs. It is thus not as flexible as triplet loss that only constrains the relative distance relation. Recent improvements are obtained by using more examples within a batch (Sohn 2016; Oh Song et al. 2016) and using an adaptive distance function to evaluate similarities (Huang, Loy, and Tang 2016). However, these embedding methods are only designed to characterize a single notion of semantic similarity. While we learn universal embeddings in a multi-attribute context, (Wu et al. 2017) claimed that the data sampling strategy in batch plays an equal or more important role than the loss function. The authors proposed a distance weighted sampling strategy that samples data uniformly according to their distance. This leads to batch data uniformly spread over the whole distance range, and provides stabler gradients than random sampling, semi-hard negative mining (Schroff, Kalenichenko, and Philbin 2015), and hard negative mining (Simo-Serra et al. 2015).

In (Yuan, Yang, and Zhang 2017), an ensemble model is used to mine hard examples at multiple levels. Despite achieved gains, these sampling strategies suffer from expensive extra costs of distance computations. Our method uses all the $O(n^3)$ triplet data in batch for the first time, and eliminates the need for hard triplet mining by a simple yet effective weighting scheme at negligible cost.

Adversarial Network Training

Adversarial training of neural networks has been shown effective for image generation (Denton et al. 2015) and data augmentation with synthetic rare images (Huang and Ramanan 2017). With similar ideas of generating adversarial images, (Goodfellow, Shlens, and Szegedy 2015) proposed to perturb image pixels to produce erroneous class labels, and many other works followed, e.g., (Zheng et al. 2016). While (Sabour et al. 2016) proposed to manipulate deep feature embeddings instead to generate hard-to-classify examples, which can be easier than image space perturbations. In line with such adversarial embedding learning, (Wang, Shrivastava, and Gupta 2017) proposed two adversarial networks to generate adversarial features that mimic object occlusions and deformations respectively and are hard for a
Figure 2: The network architecture to learn the Weighted Triplet-induced Adversarial Universal Embedding (WT-AUE). The mini-batch contains one (or several) positive pair for each of all $N_a$ attributes (denoted by different colors). Then the exhaustively sampled triplet $t = (i, j, k)$ passes through a CNN and adversarial network to obtain the perturbed universal embeddings for robustness. The universal embeddings are finally mapped to different attribute-specific subspaces, where triplet $t$ can have contradictory notions of similarity. The weighted triplet loss is proposed to aggregate the triplet losses in all subspaces, which simultaneously encourages hard triplet mining and feature interactions across concepts.

detector to classify. A related attributed-guided augmentation method (Dixit et al. 2017) synthesizes new features at a desired attribute strength via a deep encoder-decoder. In comparison, we only employ a two-layered regression network to generate adversarial universal feature embeddings, aimed for violating triplet similarity constraints in different subspaces. This simple adversarial network already works well with our proposed loss function.

**Methodology**

Our goal is to learn a universal embedding function $f(x; \Theta)$ via a CNN with parameters $\Theta$, from image $x$ into a common feature space $\mathbb{R}^d$, such that the embedded features can be shared or disentangled to encode different notions of similarity. For brevity, in the following we will omit $\Theta$ and use $f(x)$ and $f(x; \Theta)$ interchangeably.

**Multi-Task Learning with Attributes**

To achieve the above goal, we append to the universal embedding $f(x)$ one fully-connected layer with weights $W_a \in \mathbb{R}^{b \times d}$ to map to a small subspace (of dimension $b < d$) for each attribute $a$ (see Fig. 2). In each subspace, the attribute-specific similarity is used to supervise the learning of subspace features $W_a f(x)$, and in turn, that of universal embedding $f(x)$. We characterize similarities by the Euclidean distance between features in the corresponding subspace:

$$D_{ij}^a = \|W_a f(x_i) - W_a f(x_j)\|_2,$$

where the feature vector $W_a f(x_i)$ is normalized to have unit length for training stability.

The triplet loss (Schroff, Kalenichenko, and Philbin 2015) is used to learn such Euclidean distances in each subspace. We sample triplets $T_a = \{t_a = (i, j, k)\}$ with the anchor $i$, positive $j$ and negative $k$ examples defined by their labels of attribute $a$. We wish to enforce the relative distance relation $D_{ij}^a < D_{ik}^a$, and arrive to the following triplet loss for multi-task learning:

$$L_{trt} = \frac{1}{N_a |T_a|} \sum_a \sum_{t_a} \sum D_{ij}^a - D_{ik}^a + m \right]_+,$$

where $[.]_+ = \max(0, \cdot)$ is the hinge function and $m$ is the enforced margin. This loss function computes the loss for separately sampled triplets for every attribute in one mini-batch, and sums over all $N_a$ attributes. It can thus jointly train all the subspaces and universal embeddings based on back-propagation. However, two important problems are left unaddressed: how to construct the batch and how to sample triplets in it. We will introduce our specific strategies next, and show that they significantly impact both the convergence rate and performance as found in (Wu et al. 2017).

**Triplet Sampling and Weighted Triplet Loss**

To construct compact but rich mini-batches that have sufficient data for multi-task learning, we follow the method in N-pair loss (Sohn 2016) that permits information recycling. Specifically, our batch is constructed with one (or several) positive pair for each of all $N_a$ attributes, with batch size $n \propto 2N_a$ (see Fig. 2). To form triplets for each positive pair under attribute $a$, it is likely to retrieve the negative example of this attribute from the remaining examples in batch. If we similarly compute one loss for each triplet per attribute, as is done in (Sohn 2016) and many other works, we argue this is a huge waste of batch data in two ways: 1) Only one triplet is used to learn for each attribute, i.e., $|T_a| = 1$ in Eq. (2). 2) A triplet is just considered under one attribute at a time, not under all attribute labels which may capture contradictory similarity notions. It hence loses the chance of leveraging the competition between subspaces over the same data source to feedback to and improve the shared universal embeddings in terms of feature expressiveness. Indeed, Eq. (2) penalizes for various similarity aspects using separately sampled triplets $\{t_a\}$ that are not necessarily overlapped.

To avoid such data waste, we go to the other extreme and enumerate all the $O(n^3)$ triplets $t = (i, j, k)$ from a n-sized batch, where each $t$ is considered under all $N_a$ attributes. The total complexity is $O(n^3 N_a)$. Note our batch with size $n$ is organized much more efficiently than a naive one constructed with $O(n^3 N_a)$ distinct triplets.

However, by making the most of batch data in this way, we are immediately faced with several other issues: 1) It
quickly becomes intractable since the number of gradient computations per batch grows quadruply with respect to the batch size $n$ and attribute number $N_a$. 2) Many triplets will not be valid with exactly one positive pair $(i,j)$ and a negative example $k$. In the binary attribute case, the probability of composing a valid triplet $t$ is $p_t = \frac{2}{n^2} - \frac{1}{4}$. Although our batch already guarantees for each attribute, there will be at least one positive pair to form valid triplets, the invalid ones may still appear frequently and shall not contribute to the loss. 3) The optimization will suffer from slow convergence and poor local optima due to many non-informative triplets that induce (near-) zero loss. Existing works often alleviate this issue via sampling approaches, including semi-hard negative mining (Schroff, Kalenichenko, and Philbin 2015) and distance weighted sampling (Wu et al. 2017). One drawback is that they incur expensive extra costs to evaluate the feature distances for sampling.

We propose a simple yet effective weighting scheme for the $O(n^3)$ triplets, and come to a weighted triplet loss for multi-task learning:

$$L_{wt} = \frac{1}{n^2 N_a} \sum_t \sum_a w_{ijk}^a \left[ D_{ij}^a - D_{ik}^a + m \right], \quad (3)$$

where $w_{ijk}^a$ is the weight for triplet $t$ under attribute $a$, and is normalized in batch. It is defined using the Jaccard index $J(A_i, A_j) = \frac{|A_i \cap A_j|}{|A_i \cup A_j|}$ that measures similarity between the attribute label sets $A_i$ and $A_j$ of images $x_i$ and $x_j$. We have:

$$w_{ijk}^a = J(a_i, a_j) \cdot (1 - J(a_i, a_k)) \cdot (1 - J(A_i \backslash \{a_i\}, A_j \backslash \{a_j\})) \cdot J(A_i \backslash \{a_i\}, A_k \backslash \{a_k\}). \quad (4)$$

where the first two terms actually act as the triplet validity check under the current attribute $a$. They become zero for an invalid triplet and directly rule it out from the weighted loss. This significantly reduces the number of necessary computations of gradients during a backward pass, making its speed more reasonable. The last two terms help to favor those valid triplets with hard positive and negative examples, which respectively have small and large similarities of attribute labels with respect to anchor $i$. Such weighting scheme can be regarded as a soft way of hard triplet mining, but has a negligible cost without expensive example search based on deep feature distance.

**Ablation study:** Table 1 compares our Weighted Triplet loss (WT) in Eq. (3) with various baselines to quantify the efficacy of each component: 1) triplet weighting by $w_{ijk}^a$, 2) reusing each triplet across attributes $\sum_a$, 3) and another dimension of data utilization by exhaustive search of triplets within batch $\sum_t$. We have the following observations:

- Triplet weighting as a soft hard mining scheme improves convergence quality and speed. We found equal weighting with the last two terms in Eq. (4) set to one, hurts performance on the Zappos50k (Yu and Grauman 2014) and CelebA (Liu et al. 2015) datasets. In comparison, weighting with $w_{ijk}^a$ encourages hard triplets while suppressing low-quality ones to prevent poor local optima. It also converges about 3-5 times faster than equal weighting. When compared with traditional hard mining techniques, our soft weighting scheme eliminates the need for the expensive distance evaluations.

- If we do not penalize the same triplet under different attributes (used data size reduced to $O(n^2)$), performance deteriorates too. We argue that by reusing each triplet across attributes, it helps to learn the structure of the private and shared features in universal embedding to encode different triplet similarities. In other words, feature sharing and disentangling are automatically learned in such process.

- Sufficient batch data usage matters. To stress its importance for universal embedding learning, we follow the common practice in literature by only sampling $N_a$ rather than all $O(n^3)$ triplets from batch, but still keep our weighting and triplet reusing schemes. The complexity is thus reduced from $O(n^3 N_a)$ to $O(N_a \cdot N_a)$. We first sampled triplets using each of the $N_a$ positive pairs in batch and a random negative example from the rest. This leads to a significant performance drop in both accuracy and convergence speed. We also experimented with semi-hard sampling (Schroff, Kalenichenko, and Philbin 2015) and distance weighted sampling (Wu et al. 2017) for the negative example, and found slightly better results but at a larger searching cost. The N-pair triplets (Sohn 2016), extending triplets to include all the negative examples in batch rather than one, achieve larger gains verifying again the importance of using sufficient data.

<table>
<thead>
<tr>
<th>Method</th>
<th>Zappos50k</th>
<th>CelebA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal Weighting</td>
<td>9.91</td>
<td>9.88</td>
</tr>
<tr>
<td>$O(n^3)$ not across attributes</td>
<td>10.05</td>
<td>9.75</td>
</tr>
<tr>
<td>$O(N_a^2)$ random negative sampling</td>
<td>10.87</td>
<td>10.72</td>
</tr>
<tr>
<td>$O(N_a^2)$ semi-hard sampling</td>
<td>10.71</td>
<td>10.54</td>
</tr>
<tr>
<td>$O(N_a^2)$ distance weighted sampling</td>
<td>10.43</td>
<td>10.58</td>
</tr>
<tr>
<td>$O(N_a^2)$ N-pair</td>
<td>10.29</td>
<td>10.24</td>
</tr>
</tbody>
</table>

**Adversarial Learning of Universal Embedding**

The above proposed weighted triplet loss is able to jointly learn universal embeddings and the associated subspaces in a data-driven way, which requires a large amount of multi-label annotations of attributes. However, it is often difficult to collect attribute datasets that cover large enough data variance or label contrast to learn expressive universal embeddings. When such data are lacking on existing small- or medium-sized attribute datasets, we found the embedding quality is hindered indeed.

This motivates us to to enrich the data by generating adversarial examples in the feature space.
Figure 3: Adversarial perturbations of the universal embeddings of a triplet $t = (i, j, k)$, and the resulting perturbations in two example subspaces of “heel height” and “gender” of shoes. The triplet is made harder with margin-violating examples (color denotes class label) in each subspace, which improves the embedding quality and convergence speed.

adversarial training works, we propose a simple adversarial regression sub-network to generate hard features for improving embedding quality. As shown in Fig. 2, the adversarial sub-network takes the universal embedding $f(x) \in \mathbb{R}^d$ as input, and generates via two equal-sized fully-connected layers (with parameters $\Theta_{ad}$) a $d$-dimensional perturbation vector, which is then added to $f(x)$ to obtain a perturbed universal embedding $f_{ad}(x)$ with parameters $\Theta$ and $\Theta_{ad}$.

Ideally, perturbations introduced in the universal embedding space should be passed to the following subspaces to make the triplets therein hard, i.e., triplet features may violate the similarity constraint with high loss. Our objective remains the same: to penalize such violation in all subspaces. This makes us come to our final loss for Weighted Triplet-induced Adversarial Universal Embedding (WT-AUE):

$$L_{wtaue} = L_{wt}(f(X)) - \alpha L_{wt}(f_{ad}(X)) + \beta \| f_{ad}(X) - f(X) \|_2,$$  \hspace{1cm} (5)

where $X$ denotes the batch data $\{x_i\}$, the last term penalizes the $L_2$ norm of adversarial perturbations to prevent embedding inflation, and $\alpha$ and $\beta$ are weighting parameters.

Such WT-AUE loss lets the original CNN and its adversarial sub-network compete against each other: if the perturbation of universal embeddings results in violations of triplet constraints in some subspaces, the adversarial loss (second term in Eq. (5)) will be low but the original WT loss (first term) is high. By overcoming the obstacles created by the adversary, our universal embedding will be more noise-resistant. In other words, this helps impose safe margins in the local neighborhoods of both universal space and different subspaces. Fig. 3 shows an example where a triplet is perturbed to be hard with closer positive and negative examples that will violate the margin constraint in two subspaces. Separating these examples with our loss can effectively maintain local margins in all embedding spaces. By walking through the entire embedding spaces along repeated training epochs, we finally achieve consistent discrimination in any local neighborhood.

**Overall training procedure:** We first train on one dataset without adversaries for about 20k iterations, in order to learn the universal embeddings with parameters $\Theta$ and attribute-specific subspaces with $\{W_a\}$. Then we pre-train the adversarial sub-network with $\Theta_{ad}$ for about 10k iterations while fixing $\Theta$. Finally, both $\Theta_{ad}$ and $\Theta$ are jointly learned until convergence. We found the adversarial pre-training is important. It teaches the adversarial sub-network how to generate meaningful perturbations in the initial stage with random initialization. Otherwise, with a well-learned WT embedding, the adversarial learning is likely to be stuck with meaningless local minima having zero adversarial loss almost all time. To this end, we pre-train using the $N_a$ triplets sampled from batch, each adversarially trained only for one attribute rather than $N_a$, with total complexity $N_a$. This generates universal perturbations aimed for one attribute at a time, and starts to learn how to maximally violate the corresponding triplet constraint with high loss. Once sufficient adversary is learned for all attributes, we move to the joint training and the adversarial sub-network improves as the original CNN becomes better and better and vice versa.

**Experiments**

**Datasets and Implementation Details**

**Zappos50k shoe dataset.** The first attribute dataset we use is Zappos50k (Yu and Grauman 2014) with 50k images. The image size $136 \times 102$ is resized to $136 \times 102$ by us. We consider the 4 attribute labels: the shoe type (shoes, boots, sandals or slippers), shoe gender (women, men, girls or boys), the height of the shoes’ heels (numeric values from 0 to 5 inches) and the closing mechanism of the shoes (buckle, pull on, slip on, hook and loop or laced up). For the numeric “heel height” attribute labels, we use their closeness to define the positive and negative examples in triplets. Following (Veit, Belongie, and Karaletsos 2017), we split the dataset into three parts: 70% for training, 10% for validation and 20% for testing. We similarly have 40k testing triplets of attribute $\{t_a = (i, j, k)\}$ for each attribute $a$. The testing criteria is to evaluate the validity of each $t_a$ by the corresponding subspace features: whether the feature distance between the positive pair $(i, j)$ is smaller than that between the negative pair $(i, k)$ of attribute $a$. The error rate is 50% by chance. The extra shoes’ brand information is used to perform off-task classification to show the generalization ability of our universal embedding.

**CelebA dataset.** The face attribute dataset CelebA (Liu et al. 2015) contains about 202k face images of 10k identities. Each image is annotated with 40 binary attributes like “male”, “oval face” and “wavy hair”. The dataset is split into 162k training, 20k validation and 20k testing images. There are no identity overlap between these splits. The 5 key
points for each face image are used to align and crop image to 55 × 47 pixels. Both the original and cropped images are used to report our results, in terms of the average classification error across attributes. To classify each attribute for a testing image, we follow (Liu et al. 2015) to train a SVM classifier in the corresponding attribute subspace on validation set.

**Implementation details.** For fair comparisons with recent works, we similarly use the 18 layer deep residual network (He et al. 2016) and the 12-layer deep fully convolutional neural network in (Kalayeh, Gong, and Shah 2017) on Zappos50k and CelebA datasets, respectively. We use the last global average pooling layer of the two networks as our universal embedding \( f(x) \), with dimensions \( d = 64 \) and \( d = 1024 \). The respective subspace dimensions are \( b = 5 \) and \( b = 20 \). To set a proper batch size \( n \propto 2N_a \) for two datasets with attribute numbers \( N_a = 4 \) and \( N_a = 40 \), we balance the computational cost against data adequacy in batch. Since Zappos50k has much fewer attributes to consider than CelebA, we sample more positive pairs per attribute in batch for Zappos50k than for CelebA. In practice, we sample 5 and 1 positive pairs per attribute for the two datasets, having batch size \( n = 40 \) and \( n = 80 \). For all our experiments, the initial learning rate is 0.001, and \( \alpha = 1, \beta = 0.0005 \) in Eq. (5).

**Attribute and Subspace Evaluations**

We start with the evaluations of attribute prediction performance in difference subspaces. Note on Zappos50k dataset, the attribute is directly evaluated by checking the subspace features under triplet constraints; on CelebA dataset, we classify attributes based on subspace features. These experiments are to test how well our universal embedding can factorize subspaces with different attribute concepts.

Table 2 compares our method with several important baselines on Zappos50k. The single triplet embedding is learned from all available triplets as if they all come from a common space. It leads to a high error rate of 23.72%, showing that multiple attribute notions cannot be captured in a single space. Training a set of attribute-specialized triplet embeddings can be an immediate remedy, reducing the error rate to 11.35%. However, this comes at the cost of learning a separate universal embedding supervised by only \( N_a \) times more network parameters. Conditional Similarity Network (CSN) (Veit, Belongie, and Karaletsos 2017) achieves efficiency by sharing a common embedding and learning masks to attend to relevant dimensions for various attributes. CSN even outperforms training a set of embeddings in accuracy. Our Weighted Triplet loss (WT) achieves a drastic improvement (9.62%) by mapping from a universal embedding to attribute-specific subspaces, and performing a weighted combination of the exhaustively sampled triplets across all subspaces. This indicates the importance of maximal and discriminatory data usage for a collaborative subspace learning. With adversarial perturbations that robustify embedding learning, our WT-AUE achieves the lowest error rate of 9.34%.

Table 3 compares our method with the previous FaceTracer (Kumar, Belhumeur, and Nayar 2008) and PANDA (Zhang et al. 2014a) as well as state-of-the-art face prediction methods like SSP+SSG (Kalayeh, Gong, and Shah 2017). Our WT-AUE outperforms most prior arts by a large margin, and is comparable to the SSP+SSG method that employs another semantic segmentation network to improve attribute prediction. Unlike most of these single-embedding-based methods, our WT-AUE excels by joint learning of a universal embedding and different subspaces, where adversarial learning helps a lot too (in comparison to WT).

**Generalization to Unseen Attributes**

One benefit of our joint training of universal embedding and attribute-specific subspaces is that, this fosters automatic feature interactions (e.g., sharing or disentangling) in the universal embedding space to generate different attribute notions or even new ones. This suggests the potential of feature generalization from reduced feature redundancy. To prove this hypothesis empirically, we conduct a generalization experiment in the leave-one-attribute-out style: each time we learn the universal embedding supervised by only \( N_a - 1 \) attributes, and then freeze the universal embedding parameters to learn the subspace \( W_a \) for the left out (thus new) attribute. Note both learning stages use the WT-AUE loss. Our goal is to test how well such learned universal embedding can generalize to unseen attribute concepts.

Considering the Zappos50k dataset (50k images) is smaller than CelebA (202k images), we use Zappos50k to better examine our embedding’s generalization ability with a small data size. Moreover, we further mimic low-shot learning for the left-out attribute with a decreasing amount of training data for it (80%, 40%, 10%). Table 4 lists the triplet prediction error rate averaged across \( N_a = 4 \) attributes under such generalization test. We observe a graceful performance degradation with fewer and fewer training data of un-

---

**Table 2:** Average triplet prediction error (%) of 4 attributes on Zappos50k dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single triplet embedding</td>
<td>23.72</td>
</tr>
<tr>
<td>Set of specialized triplet embeddings</td>
<td>11.35</td>
</tr>
<tr>
<td>Conditional similarity network</td>
<td>10.73</td>
</tr>
<tr>
<td>WT</td>
<td>9.62</td>
</tr>
<tr>
<td>WT-AUE</td>
<td>9.34</td>
</tr>
</tbody>
</table>

**Table 3:** Average classification error (%) of 40 attributes on the CelebA original and cropped image sets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Original</th>
<th>Cropped</th>
</tr>
</thead>
<tbody>
<tr>
<td>FaceTracer</td>
<td>18.88</td>
<td>-</td>
</tr>
<tr>
<td>PANDA</td>
<td>15.00</td>
<td>-</td>
</tr>
<tr>
<td>(Liu et al. 2015)</td>
<td>12.70</td>
<td>-</td>
</tr>
<tr>
<td>(Wang, Cheng, and Feris 2016)</td>
<td>12.00</td>
<td>-</td>
</tr>
<tr>
<td>(Zhong, Sullivan, and Li 2016)</td>
<td>10.20</td>
<td>-</td>
</tr>
<tr>
<td>SSP+SSG</td>
<td>8.84</td>
<td>8.20</td>
</tr>
<tr>
<td>WT</td>
<td>9.58</td>
<td>9.19</td>
</tr>
<tr>
<td>WT-AUE</td>
<td>9.02</td>
<td>8.46</td>
</tr>
</tbody>
</table>
seen attributes. More interestingly, using only 10% training data of them, we achieve comparable results to the state-of-the-art conditional similarity network (Veit, Belongie, and Karaletos 2017). This indeed validates the superior generalization ability of our universal embedding.

**Transfer Learning for Off-Task Recognition**

Another way to evaluate the generalization ability of our universal embedding is to use it for transfer learning under a different task. We use Zappos50k dataset again for its small size. Following (Veit, Belongie, and Karaletos 2017), we select the 30 shoe brands with the most examples in Zappos50k for standard multi-way classification. During training, we first learn the universal embedding supervised by 4 attributes with the WT-AUE loss, then fix the embedding and learn one fully connected layer with Softmax loss for the 30 brand classes.

Table 5 shows our method attains the highest classification accuracy. Single triplet embedding hurts the performance since it learns contradictory similarity notions and hinders the embedding quality. Our method outperforms the conditional similarity network (Veit, Belongie, and Karaletos 2017) and achieves gains over the ImageNet pre-trained model. This indicates the strong supervision provided by our WT-AUE loss boosts embedding generalization.

**Conclusion**

We present a multi-task learning method to learn universal feature embeddings from multi-label attributes. By mapping universal embeddings to various attribute-specific subspaces, we propose a weighted triplet loss to learn from all the triplets, especially those hard ones in all subspaces during each mini-batch training. Adversary is also introduced to robustify the learning by competing against the universal embedding perturbations. We show this effectively encourages feature sharing and disentangling in universal space, which leads to reduced feature redundancy and boosted generalization ability. Experiments not only demonstrate the superior attribute prediction results by our learned subspaces, but also validate the generalization ability of our universal embedding in the tasks of low-shot learning for unseen attributes and off-task recognition.

**References**


Rudd, E. M.; Günther, M.; and Boult, T. E. 2016. MOON: A mixed objective optimization network for the recognition of facial attributes. In ECCV.

Sabour, S.; Cao, Y.; Faghri, F.; and Fleet, D. J. 2016. Adversarial manipulation of deep representations. In ICLR.


Sohn, K. 2016. Improved deep metric learning with multi-class N-pair loss objective. In NIPS.


Wang, J.; Cheng, Y.; and Feris, R. S. 2016. Walk and learn: Facial attribute representation learning from egocentric video and contextual data. In CVPR.


Zhang, X.; Zhou, F.; Lin, Y.; and Zhang, S. 2016. Embedding label structures for fine-grained feature representation. In CVPR.


Zhong, Y.; Sullivan, J.; and Li, H. 2016. Face attribute prediction with classification CNN. In ICIP.