Jointly Learning Multiple Perceptual Similarities

Liwen Zhang
The University of Chicago
liwenz@cs.uchicago.edu

Subhransu Maji
UMass Amherst
smaji@cs.umass.edu

Ryota Tomioka
Toyota Technological Institute at Chicago
tomioka@ttic.edu

Abstract

Perceptual similarity between objects is multi-faceted and it is easier to judge similarity when the focus is on a specific aspect. We consider the problem of mapping objects into view specific embeddings where the distance between them is consistent with the similarity comparisons of the form “from the t-th perspective, object A is more similar to B than to C”. Our framework jointly learns view specific embeddings and can exploit correlations between views if they exist. Experiments on a number of datasets, including a large dataset of multi-view crowdsourced comparison on bird images, show the proposed method achieves lower triplet generalization error and better grouping of classes in most cases, when compared to learning embeddings independently for each view. The improvements are especially large in the realistic setting when there is limited triplet data for each view.

1 Introduction

Measure of similarity plays an important role in applications such as content-based recommendation, image search and speech recognition. While the nature of similarity between objects can be abstract, it is typically captured by either (a) representing each object as a fixed length vector (explicit parameterization) or (b) representing the distances or inner products between objects as a distance matrix or a Gram matrix (implicit parameterization) and learn them so that they conform to the similarity observation. Explicit embeddings are attractive because they can be used directly in a number of methods that required fixed length representations of objects. This allows simple algorithms based on vector operations to capture complex relations between objects, as has been demonstrated by the applications of word embeddings \cite{mikolov2013distributed} in language.

A number of techniques to learn embeddings from single view have been proposed \cite{hinton2012deep, roweis2000nonlinear, weinberger2009distance}. Similarity comparison of the form “object A is more similar to B than to C”, which we call triplet comparison, is a commonly used form of supervision for learning such embeddings \cite{bromley1993signature, li2013metric, luo2014triplet}. Such comparisons may be derived from class labels, or any external source of information. For example, the occurrence of words in proximity in a sentence was used to learn word embeddings.

We focus on the case when such similarity comparisons are provided by a human so that the learned embeddings reflect a perceptual similarity. Although judging similarity comparisons is easier than rating similarity on an absolute scale \cite{ghazanfar2002categorization}, there can be considerable ambiguity. Consider the problem of comparing three birds as seen in Fig. 1. Most annotators will say that the head of bird A is more similar to the head of B while the back of A is

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Ambiguity in similarity. A is more similar to C than B when focussing on the back (middle row), but is more similar to B than C when focussing on the head (bottom row).}
\end{figure}
more similar to $C$. Such ambiguity often leads to noise in annotation resulting in poor embeddings.

A better approach would be to tell the annotator the desired view or the perspective of the object to use for measuring similarity. Such view-specific comparisons are not only easier for annotators, they can also enable precise feedback for human “in the loop” tasks, such as, interactive fine-grained recognition [21], thereby reducing the human effort. The main drawback of learning view interactive fine-grained recognition [21], thereby reducing the human effort. The main drawback of learning view specific embeddings independently is that the number of similarity comparisons scales linearly with the number of views. This is undesirable as even learning a single embedding of $N$ objects may require $O(N^3)$ triplet comparisons [6] in the worst case.

We propose a method for learning embeddings jointly that addresses this drawback. Our method exploits underlying correlations that may exist between the views allowing a better use of the training data. For instance, photos of a car taken from different angles are correlated, as they can be considered as projections of a 3D model onto different planes. In a symphony, musics played by individual instruments in the orchestra could be related, as they may share some of the melody or rhythm. Our method models the correlation between views by assuming that each view is a low-rank projection of a common embedding. We show that the resulting optimization problem can be solved using an iterative procedure that alternates between updating the low-rank projection matrices and the common embedding.

We experiment with a synthetic dataset and three realistic datasets from different domains, namely, poses of airplanes, features from faces (PubFig dataset [10]), and crowd-sourced similarities collected on different body parts of birds (CUB dataset [23]). Our conclusions are that for a given amount of training data per view, the proposed joint learning approach obtains lower triplet generalization error compared to the naive independent learning approach on most datasets. The proposed joint learning approach also tends to obtain better cluster structures at the category level, which are measured through the leave-one-out classification error. In the more realistic setting where there is little training data for each view the joint embeddings are significantly better.

## 2 Related Work

Different techniques have been studied for embedding data points based on similarity triplets. Agarwal et al. [11] sought to find an embedding where pair-wise Euclidean distances have the same ordering as a given set of dissimilarities. McFee and Lanckiet [11, 12] studied the same setting using multiple kernel learning. Tamuz et al. [17] proposed a method to learn an embedding from crowd sourced data alone where the training set consists of triplet relations. Van der Maaten et al. [19] proposed a technique called t-Distributed Stochastic Embedding (t-STE) in the triplet setting using Student-t kernel to model the similarity so that the resulting embedding is more compact and clustered.

While the studies above deal with relative similarity measure over three or four objects, plenty of work has focused on class labels as a source of supervision. For example, Weinberger and Saul [22] aimed at maximizing the margin between input vectors from different classes so that, under the learned metric, objects from the same class are clustered together. Parameswaran and Weinberger [14] extended this method to a multiple-task setting and consider the problems where data could be labelled under different concepts.

However, the problem we consider here is different from work mentioned above in two folds: 1) Existing work that deals with triplets comparisons learns a single similarity metric. Instead, we consider multiple similarity measures and we aim to jointly learn embeddings that are consistent with the multi-view observations. 2) instead of using class labels we use triplet comparisons, which can be obtained in much larger scale [17, 21] and more suitable for capturing perceptual similarities.

Finally, our approach is complementary to methods that seek to minimize user effort by actively collecting triplet comparisons [6, 17] and better interfaces for triplet collection [24].

## 3 Formulation

In this section, we first review the single metric triplet embedding problem considered in [11, 17, 19]. Then we formulate its extension to the case where there exist multiple measures of similarity.

### 3.1 Triplet Embedding

Given a set of triplets $S = \{(i, j, k) \mid z_i$ is more similar to $z_j$ than $z_k\}$, we aim to find an embedding $\{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}^D$, for some $D \ll N$, such that the a pair-wise comparison of Euclidean distances (approximately) agrees with $S$, i.e., $(i, j, k) \in S \Rightarrow ||x_i - x_j||^2 < ||x_i - x_k||^2$. This is the setting studied by van der Maaten and Weinberger [19] and it is a special case of the paired comparison setting proposed in Agarwal et al. [11], which originates from the work of Shepard and Kruskal [15, 16, 8, 9].

To this end, a commonly employed strategy is to define a loss function $\ell$ that measures how well the embedding
models a triplet \((i, j, k)\) and solve a minimization problem of the form

\[
\min_{X \in \mathbb{R}^{N \times D}} \quad C \cdot \sum_{(i,j,k) \in S} \ell(d(i,j), d(i,k)) + \|X\|_F^2,
\]

where \(\| \cdot \|_F\) is the Frobenius norm and \(C > 0\) is a regularization parameter. When the hinge loss \(\ell(d(i,j), d(i,k)) = \max(1 + d(i,j) - d(i,k), 0)\) is used, the above framework is known as generalized non-metric multidimensional scaling (GNMDS; [11]). Other choices of loss functions lead to crowd kernel learning [17] and \(t\)-distributed stochastic triplet embedding (t-STE) [19].

Since the squared Euclidean distance \(\|x_i - x_j\|^2\) can be expressed as \(\|x_i - x_j\|^2 = k_{ii} - 2k_{ij} + k_{jj}\) using a positive semidefinite Gram matrix \(K \succeq 0\), minimization problem (1) can be equivalently rewritten as follows:

\[
\min_{K \in \mathbb{R}^{N \times N}} \quad C \cdot \sum_{(i,j,k) \in S} \ell(d_K(i,j), d_K(i,k)) + \text{tr}(K),
\]

s.t. \(K \succeq 0\),

where \(d_K(i,j) = k_{ii} - 2k_{ij} + k_{jj}\). Penalizing the trace of the Gram matrix can be seen as a convex surrogate for penalizing the rank of the Gram matrix [1][4]. After the optimal \(K\) is learned, we recover the embedding \(X = [x_1, \ldots, x_N]^\top\) by the decomposition \(K = XX^\top\) up to rotation.

Since optimization problems (1) and (2) are equivalent up to rotation, we will interchangeably say “learning the embedding \(X\)” and “learning the metric \(K\”). However, optimization problem (2) is convex if the loss function \(\ell\) is convex, while this is not true for (1).

### 3.2 Multiple-metric Triplet Embedding

While previous studies deal with the situation where there is one underlying similarity measure, we would like to consider the case that objects are compared from a few different aspects, \(i.e.,\) there exists multiple measures of similarity.

More precisely, we assume that \(T\) sets of triplets \(S_1, \ldots, S_T\) are obtained by asking annotators to focus on a specific aspect when making pair-wise comparisons. In addition, a set of triplets \(S_0\) corresponding to global (unspecific) notion of similarity may be available. Our goal is to obtain an embedding for the global notion of similarity as well as embeddings corresponding to local views. To this end, we take a hybrid approach that combines (1) and (2) and use explicit parametrization (1) for the global embedding of each object and Gram-matrix-based parametrization (2) for modeling each view.

Let \(x_1, \ldots, x_N \in \mathbb{R}^D\) be vectors corresponding to the global embedding of the objects as above. Each local view is associated with a Gram matrix \(M_t\) and the underlying metric is defined as a (squared) Mahalanobis distance \(d_{M_t}(i,j) := (x_i - x_j)^\top M_t (x_i - x_j)\). We formulate the learning problem as follows:

\[
\min_{M_1, \ldots, M_T} \sum_{(i,j,k) \in S_0} \ell \left( \|x_i - x_j\|^2, \|x_i - x_k\|^2 \right) + \sum_{t=1}^T \sum_{(i,j,k) \in S_t} \ell \left( (1 + d_{M_t}(i,j)) - d_{M_t}(i,k) \right) + \sum_{t=1}^T \gamma \text{tr}(M_t) + \beta \|X\|_F^2,
\]

s.t. \(M_t \succeq 0 \quad (t = 1, \ldots, T)\).

where \(\ell\) is a loss function as above. We use the hinge loss in the experiments in this paper, but the proposed framework readily generalizes to other loss functions proposed in literature [17][19].

We employ regularization terms for both the local metric \(M_t\) and the global embedding vectors \(x_i\) in (3). We add constraints on \(\text{tr}(M_t)\) in the optimization objective, as the trace norm is known to be effective in producing low-rank solutions [1][4]. The regularization term on the norm of \(x_i\) is necessary because of the scale ambiguity; the trace of \(M_t\) can be reduced by scaling down \(M_t\) while scaling up \(x_i\)’s.

Although objective function (3) is non-convex, if we fix the value of \(x_i\)’s and choose a convex loss function, \(e.g.,\) hinge-loss, then it becomes a convex problem with respect to \(M_t\)’s and \(M_t\)’s can be learned independently since they appear in disjoint terms.

To minimize the objective, we update \(M_t\)’s and \(x_i\)’s alternately. When \(M_t\)’s are fixed, \(x_i\)’s are updated directly via gradient descent (Algorithm 1). When \(x_i\)’s are fixed, \(M_t\)’s can be updated independently via projected gradient descent, \(i.e.,\) by iteratively taking a sub-gradient step and projecting the resulting \(M_t\) onto the positive semidefinite cone (Algorithm 2). The number of dimension \(D\) is left as a hyper parameter. The algorithm is summarized in Algorithm 3.

#### Effective regularization term

Considering the scale ambiguity between \(x_i\) and \(M_t\) more carefully, we can reduce the sum of trace norm and squared Frobenius norm terms in (3) into a single effective regularization term with only one hyperparameter.

In fact, for any \(\alpha > 0\), we can rescale \(X\) and \(M_t\) as \(\alpha X\) and \(\frac{M_t}{\alpha^2}\), respectively, without affecting the loss term in (3). Then due to the arithmetic-mean-geometric-mean
inequality, we have
\[
\beta \alpha^2 \|X\|_F^2 + \sum_{t=1}^{T} \frac{\gamma}{\alpha^2} \text{tr}(M_t) \geq 2 \sqrt{\beta \gamma \|X\|_F^2} \sum_{t=1}^{T} \text{tr}(M_t),
\]
(4)
in which the lower bound can be reached by setting \(\alpha\) to the value:
\[
\alpha = \left( \frac{\gamma \sum_{t=1}^{T} \text{tr}(M_t)}{\beta \|X\|_F^2} \right)^{\frac{1}{4}}.
\]
This shows that, the effective regularization only depends on the product \(\beta \gamma\) of the two hyperparameters.

**Number of parameters**

Why is it a good idea to have a shared global view? A simple parameter counting argument tells us that independently learning \(T\) views requires to fit \(O(NDT)\) parameters, where \(N\) is the number of objects, \(D\) is the number of dimensions, and \(T\) is the number of views. On the other hand, our joint learning model has only \(O(ND + D^2T)\) parameters. Thus when \(D < N\), our model has much fewer parameters and enables better generalization, especially when the number of triplets is limited.

**Explicit vs. implicit parametrization**

In the above formulation we employed explicit parametrization for the global view and implicit (Gram matrix based) parametrization for the local views. Since the cost of eigen decomposition required in Algorithm 2 scales with the dimension of the Gram matrix, even if we take \(D = 100\), our algorithm is very fast. On the other hand, if we take the other option (Gram matrix based parametrization for the global view and explicit parametrization for the local views), the algorithm would not be scalable, because we need to solve eigen problem of size \(N \times N\) in each iteration.

### 4 Experiments

In this section, we test our algorithm on both synthetic data and real datasets. One of the real dataset consists of images of airplanes with different poses where similarity between two poses is defined rigorously. Another real dataset uses triplets sampled from a few kernel matrices which are computed from attributes of facial images. The other dataset contains images of 200 species of birds where similarity triplets among the birds are “crowd-sourced”. We first give a brief explanation on the design of the experiments and introduce the datasets. Experimental results follow.

**Algorithm 1:** Update \(x_i\)’s

**Input:** \(x_i, i = 1, 2, \ldots, N; M_t, t = 1, 2, \ldots, T; \beta\).
**Output:** \(x_i, i = 1, 2, \ldots, N\).
**Initialization:** choose a proper initial step size \(\eta_0\); set \(M_0 := I_D;\) set \(\eta \leftarrow \eta_0;\) set iteration counter \(m = 1;\)
**while not converged do**
\[
g_n = 2\beta x_n + \sum_{t=0}^{T-1} \sum_{(i,j,k) \in S_t, \ n \in (i,j,k)} \nabla x_n \ell(d_{M_t}(i,j), \ d_{M_t}(i,k))
\]
**Update**
\[
x_n \leftarrow x_n - \eta g_n \quad (n = 1, \ldots, N);
\]
**Update counter:** \(m \leftarrow m + 1;\)
**Update step size:** \(\eta \leftarrow \eta_0/\sqrt{m};\)
**end**

**Algorithm 2:** Update \(M_t\)

**Input:** \(x_i, i = 1, 2, \ldots, N; M_t; \gamma\).
**Output:** \(M_t\).
**Initialization:** choose a proper initial step size \(\eta_0\); set \(\eta \leftarrow \eta_0;\) set iteration counter \(m = 1;\)
**while not converged do**
\[
G \leftarrow \sum_{(i,j,k) \in S_t} \nabla M_t \ell(d_{M_t}(i,j), \ d_{M_t}(i,k)) + \gamma I_D;
M_t \leftarrow M_t - \eta G;
\]
Find the eigenvalue decomposition \(M_t = V \Lambda V^T;\)
**Project** \(M_t\) to PSD cone:
\[
M_t \leftarrow V \max(\Lambda, 0)V^T;
\]
**Update counter:** \(m \leftarrow m + 1;\)
**Update step size:** \(\eta \leftarrow \eta_0/\sqrt{m};\)
**end**

**Algorithm 3:** Multiple-metric Learning

**Input:** \(N,\) the number of objects; \(D,\) dimension of embedding; \(S_t, \ t = 0, 1, 2, \ldots, T,\) triplet constraints. Regularization parameters \(\beta, \gamma.\)
**Output:** Embedding \(\{x_i\}_{i=1}^N,\) PSD matrices \(\{M_t\}_{t=1}^T\)
**Initialization:** initialize \(x_i\)’s randomly, initialize \(M_t\) as identity matrices;
**while not converged do**
\[
\text{Update } x_i\text{'s by using Algorithm 1;}
\]
for \(t \in \{1, 2, \ldots, T\}\) do
\[
\text{Update } M_t\text{'s by using Algorithm 2.}
\]
**end**
4.1 Experimental Setup

On each dataset, we inspect the quality of embeddings learned from training sets with increasing sizes. We draw about equal numbers of training triplets from all views and check the quality of embedding. In addition, we inspect how the similarity knowledge on existing views could be “transferred” to a “new” view where the number of similarity comparisons is small. We did this by conducting an experiment in which we draw a small set of training triplets from one view but use large numbers of training triplets from the rest views. The quality of embeddings is measured from the following aspects:

1. Triplet generalization error. We split triplets randomly into a training set and a test set. Triplet generalization error is defined as the percentage of test triplets whose triplet relations are not correctly modelled by the learned embedding.

2. Leave-one-out classification error. We held out all information about objects’ class label during the training stage. Only triplet constraints are used for learning the embedding. Then, at the test stage, we choose one embedded object as target and predict its label by revealing the labels of its neighbours. We do this prediction for every objects in turn. The leave-one-out classification error is the percentage of objects whose labels are not correctly predicted. Throughout the experiments, we use a 3-nearest-neighbour classifier to test classification error.

In [19], van der Maaten et al. showed that reducing triplet generalization error often leads to small nearest-neighbour classification error, although small triplet generalization does not necessarily guarantee low classification error. We also inspect embeddings obtained by simultaneously learning multiple metrics from this aspect.

In the experiments, we use hinge loss as the loss function. As a baseline, we conduct triplet embedding on every view as if they are independent tasks. We adopt the problem (1) as the objective and solve it by using the package provided by the author of [19]. To tune the regularization parameters, we further split the training sets for a 5-fold cross-validation and swept over \( \{10^{-5}, 10^{-4}, \ldots, 10^{5}\} \) for all parameters.

4.2 Datasets

Here are the details of the datasets:

Synthetic Data

Two synthetic datasets are generated. One consists of 200 points uniformly sampled from a 10 dimensional unit hypercube, while the other dataset have 200 objects from a mixture of four Gaussian with variance 1 whose centers are randomly chosen in a hypercube with side length 10. Six views are generated on each dataset. Each view is produced by projecting data points onto a random five dimensional subspace. Training and test triplets \((i,j,k)\) are randomly sampled from all possible triplets on every views.

Poses of Airplanes

This dataset is constructed from 200 images of airplanes from the PASCAL VOC dataset [3] which are annotated with 16 landmarks such as nose tip, wing tips, etc [2]. We use these landmarks to construct a pose-based similarity. Given two planes and the positions of landmarks in these images, pose similarity is defined as the residual error of alignment between the two sets of landmarks under scaling and translation. We generated 5 views each of which is associated with a subset of these landmarks as seen in Fig. 2 which shows three annotated images from the set. The planes are highly diverse ranging from passenger planes to fighter jets, varying in size and form which results in a slightly different similarity between instances for each view. However, there is a strong correlation between the views because the underlying set of landmarks are shared.

Additionally, we categorize the planes into five classes: “left-facing”, “right-facing”, “pointing-up”, “pointing-down” and “facing out or facing away” to evaluate classification error. This produces classes with unbalanced number of members. About 80% of the images belongs to one of these three classes: “left-facing”, “right-facing” and “facing out or facing away”.

![Figure 2: View specific similarities between poses of planes are obtained by considering subsets of landmarks shown by different colored rectangles and measuring their similarity in configuration up to a scaling and translation. For e.g. view 1 consists of all landmarks.](image-url)
triplet constraints on associated species, \(|K_i,j,l|K_{\text{sim}}\) containing birds considered similar to the target and \(K_{\text{dissim}}\) having the ones considered dissimilar. Such a partition is broadcast to an equivalent set of triplet constraints on associated species, \(\{(i,j,l)\mid j \in K_{\text{sim}}, l \in K_{\text{dissim}}\}\). Therefore, for each user response, \(|K_{\text{sim}}|K_{\text{dissim}}|\) triplet constraints are acquired.

In the setting of multiple metric embedding, different views of birds are obtained cropping regions around various parts of the bird as shown in Fig. 3, and then using the same procedure as before to collect triplet comparisons. In this dataset, there are about 100,000 triplets on similarities from comparisons made on the whole birds, while there are other 5 views each of which is cast on localized region of the birds (e.g. beak, breast, wing). Number of triplets obtained from these views range from about 4,000 to 7,000. For testing classification error, we use a taxonomy of the bird species obtained from the authors of [23]. To make the number of objects in all classes balanced, we manually grouped some of the classes to get 6 super classes in total.

This data set is more challenging than the other data sets in a sense that it emerges in a more realistic situation where not necessarily all triplet relations are available due to the nature of crowdsourcing.

We note that these global and local similarity triplets were used in [20] and [21] as a way to incorporate human feedback during recognition. However, our focus is to learn better embeddings of the data itself by combining information across different views.

4.3 Results

Synthetic data

We first conduct experiments on synthetic dataset. Embedding is learned in a 10 dimensional space. Triplet generalization errors and leave-one-out 3-nearest-neighbour classification errors are plotted in Fig. 4. Our algorithm achieves smaller triplet generalization error on both clustered and uniformly distributed dataset. The independent learning requires about 3–40,000 triplets to obtain triplet generalization error less than 0.1, whereas the joint learning achieves the same error around 10,000 triplets. In terms of classification error, after seeing 5,000 triplets, the joint learning obtains zero error, whereas independent learning requires about 10 times of that to obtain a comparable error.

Poses of Airplanes

The airplanes are embedded into a 10 dimensional space based on a training setting which consists of 3,000 triplets from every view. As an illustration, we project the learned global view of the objects onto their first two principle dimensions via SVD and illustrated the embedding in Fig. 5. The visualization shows that objects roughly lies on a circle corresponding to the left-right and up-down orientation.

Fig. 6 shows the triplet generalization errors and classification errors of the learned embeddings. Errors on local views are shown in small plots while the large plots...
show the average across views. In terms of the average triplet generalization across all the views, the proposed joint model is clearly better than the independent learning up to 10,000 triplets or so when the difference becomes indistinguishable. This is not only in average but also uniformly for each view (see the small plots). The 3-NN classification error shows that the proposed joint learning results in a more clustered embedding than independent learning.

Public Figures Face Dataset

The 200 images are embedded into 5, 10, and 20 dimensional spaces. We draw triplets randomly from the ground truth similarity measure to form training and test sets. Triplet generalization errors and classification errors are shown in Fig. 7.

In terms of the triplet generalization error, the joint learning reduces the error faster than the independent learning up to around 10,000 triplets where the decrease slows down. Since the error in this regime reduces monotonically with increasing number of dimensions, this can be understood as a bias induced by the joint learning. On the other hand, when we have less than 10,000 triplets, the error of the joint learning increases (but not as large as the independent learning) as dimension increases; this can be understood as a variance. When embedding in a 20 dimensional space, the joint learning has lower or comparable error to independent learning even when $10^5$ triplets are available. In terms of the leave-one-out classification error, joint learning continues to be better even when the number of triplets are very large.

CUB-200 Birds Data

We learn the embedding both in a 10 dimensional space and a 60 dimensional space. An illustration of embedding learned in 60 dimensional space can be found in Fig. 8. The joint embedding seems to group some clusters (e.g., Charadriiformes) tighter than the independent embedding.

In the birds dataset, more triplets are available from the first view compared to other views. Therefore, we first sample equal numbers of triplets in each view up to 3,000 triplets in each view. Afterwards, we add triplets only to the first view. The triplet generalization error is measured using remaining triplets in each view.

Figure 7 shows the triplet generalization errors and the leave-one-out 3-nearest-neighbour classification errors. The solid vertical line shows the point that we start to add training triplets only to the first view. Comparing joint learning in 10 dimensions and 60 dimensions, we see that the higher dimension gives the lower error; thus...
Figure 6: Results on poses of planes dataset. Embedding is learned in a 10 dimensional space. Left: triplet generalization error. Right: leave-one-out 3-nearest-neighbor classification error. The small figures shows errors on individual views and the large figures show the average.

Figure 7: Results on public figures faces dataset. Embeddings are learned in a 5 dimensional space, a 10 dimensional space and a 20 dimensional space. Left: triplet generalization error. Right: leave-one-out 3-nearest-neighbor classification error. The small figures shows errors on individual views and the large figures show the average.

Figure 9: Results on CUB-200 birds dataset. Left: triplet generalization error. Right: leave-one-out 3-nearest-neighbor classification error. The small figures shows errors on individual views and the large figures show the average.

Figure 10: Learning a new view on CUB-200 birds dataset. Training data contains 100 triplets from the second local view and 3,000 triplets from other 5 views. Embeddings are learned in a 10 dimensional space and then further embedded in a 2 dimensional plane by using tSNE [18] for the purpose of visualization. Left: triplet generalization error on the second local view. Middle: embedding learned independently. Right: embedding learned jointly.
we suspect that the triplet generalization error is dominated by the bias component. The error of independent learning is comparable to joint learning for small number of triplets and becomes smaller as the number of triplets increases. Interestingly the error of joint learning in 60 dimensions coincides with that of independent learning in 10 dimensions after seeing 6,000 triplets. This can be explained by the fact that with 6 views, the two models have comparable complexity (see discussion at the end of Section 3.2) and thus the same asymptotic variance. In terms of leave-one-out classification error, our method obtains lower errors on all views except for the first view.

Learning a New View

On the CUB-200 birds dataset, we simulated the setting of learning a new view (or zero-shot learning). We draw a training set that contains 100 triplets from the 2nd local view and 3,000 triplets from all other 5 views. We investigate how joint learning helps in getting a good embedding on a new view with extremely small number of samples. The learned embeddings are shown in Fig. 10. It is evident that, embedding learned purely out of 100 triplets from that view has objects from different classes completely mixed together, while the local view from the embedding learned jointly group objects from some classes better. For example, members in the class Passeriformes(Emberizidae) is more separated from members in Passeriformes(Icteridae). The triplet generalization errors of both approaches (left panel in Fig. 10) is consistent with the visualization. The triplet generalization error of the proposed joint learning is lower than that of the independent learning up to around 700 triplets.

4.4 Relating the performance gain with the triplet consistency

In Table 1 we relate the performance gain we obtained for the joint learning approach compared to the independent learning approach to the underlying between-task similarity. The performance gain is measured by the difference between the area under the triplet generalization errors normalized by that of the independent learning. The between-task similarity is measured by triplet consistency between different views averaged over all task pairs. For the CUB-200 dataset in which only a subset of valid triplet constraints are available, we take the independently learned embeddings with the largest number of triplets and use those to compute the triplet consistency.

We can see that in both synthetic and poses of airplanes datasets where we saw significant gain by using the proposed joint approach, the triplet consistency is high. On the other hand, the consistency is lower or close to ran-


dom (0.5) for the other two datasets possibly explaining why the performance gain was not as significant as in the first two datasets.

|                  | Synthetic (clustered) | Airplanes | PubFig | CUB-200 |
|------------------|-----------------------|-----------|--------|---------|
| Average triplet consistency | 0.78                  | 0.85      | 0.59   | 0.53    |
| Performance Gain (%)  | 52.37                 | 23.97     | 9.89   | 0.44    |

5 Discussion

We have proposed a model for jointly learning multiple measures of similarities from multi-view triplet observations. The proposed model consists of the global view, which represents each object as a fixed dimensional vector, and local views, which specifies each view-specific Mahalanobis metric as a positive semidefinite matrix.

Experiments on both synthetic and real datasets have demonstrate that in most cases our proposed joint model outperforms the conventional independent learning model. Although the results are presented for the hinge loss (i.e., GNMDS [1]), the proposed algorithm easily generalizes to other loss functions, e.g., the t-STE loss [19].

We have explained the performance gain in terms of a variance-bias trade-off; that is, the assumption of the shared global view leads to a significant reduction of sample complexity. Since in many real applications, similarity triplets can be expensive to obtain, jointly learning similarity metrics is preferable as it can recover the underlying structure using relatively small number of training data. Moreover, we have empirically calculated the triplet consistency between pairs of views on all datasets and showed that the more similar the views are, the greater the performance gain becomes.

Experimental results also showed that jointly learning multiple metrics performs better in terms of 3-nearest-neighbour classification error on all datasets, which implies that it has a potential to recover the category level structure of the data. As a future work, we aim to study how to use embeddings learned from similarity comparisons for classification tasks where data is partially labelled.
References

[1] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. J. Kriegman, and S. Belongie. Generalized non-metric multidimensional scaling. In International Conference on Artificial Intelligence and Statistics, pages 11–18, 2007.

[2] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In European Conference on Computer Vision (ECCV), 2010.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International Journal of Computer Vision, 88(2):303–338, June 2010.

[4] M. Fazel, H. Hindi, and S. P. Boyd. A Rank Minimization Heuristic with Application to Minimum Order System Approximation. In Proc. of the American Control Conference, 2001.

[5] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008.

[6] K. G. Jamieson and R. D. Nowak. Low-dimensional embedding using adaptively selected ordinal data. In Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on, pages 1077–1084. IEEE, 2011.

[7] M. G. Kendall and J. D. Gibbons. Rank correlation methods. Edward Arnold. Oxford University Press, 1990.

[8] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika, 29(1):1–27, 1964.

[9] J. B. Kruskal. Nonmetric multidimensional scaling: a numerical method. Psychometrika, 29(2):115–129, 1964.

[10] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In Computer Vision, 2009 IEEE 12th International Conference on, pages 365–372. IEEE, 2009.

[11] B. McFee and G. Lanckriet. Partial order embedding with multiple kernels. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 721–728. ACM, 2009.

[12] B. McFee and G. Lanckriet. Learning multi-modal similarity. The Journal of Machine Learning Research, 12:491–523, 2011.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[14] S. Parameswaran and K. Q. Weinberger. Large margin multi-task metric learning. In Advances in neural information processing systems, pages 1867–1875, 2010.

[15] R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. I. Psychometrika, 27(2):125–140, 1962.

[16] R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. II. Psychometrika, 27(3):219–246, 1962.

[17] O. Tamuz, C. Liu, S. Belongie, O. Shamir, and A. T. Kalai. Adaptively learning the crowd kernel. arXiv preprint arXiv:1105.1033, 2011.

[18] L. van der Maaten and G. Hinton. Visualizing data using t-sne. Journal of Machine Learning Research, 9(2579-2605):85, 2008.

[19] L. van der Maaten and K. Weinberger. Stochastic triplet embedding. In Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on, pages 1–6. IEEE, 2012.

[20] C. Wah, G. V. Horn, S. Branson, S. Maji, P. Perona, and S. Belongie. Similarity comparisons for interactive fine-grained categorization. In Computer Vision and Pattern Recognition, 2014.

[21] C. Wah, S. Maji, and S. Belongie. Learning localized perceptual similarity metrics for interactive categorization. In IEEE Winter Conference on Applications of Computer Vision, WACV, 2015.

[22] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. The Journal of Machine Learning Research, 10:207–244, 2009.

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

[24] M. J. Wilber, I. S. Kwak, and S. J. Belongie. Cost-effective hits for relative similarity comparisons. arXiv preprint arXiv:1404.3291, 2014.
Supplementary Material

A Views of Poses of Airplanes Dataset

Each of the 200 airplanes were annotated with 16 landmarks namely,

01. Top_Rudder  05. L._WingTip  09. Nose_Bottom  13. Left_Engine_Back
02. Bot_Rudder  06. R._WingTip  10. Left_Wing_Base  14. Right_Engine_Front
03. L._Stabilizer  07. NoseTip  11. Right_Wing_Base  15. Right_Engine_Back
04. R._Stabilizer  08. Nose_Top  12. Left_Engine_Front  16. Bot_Rudder_Front

This is also illustrated in the Figure A.1. The five different views are defined by considering different subsets of landmarks as follows:

1. \( \text{all} \in \{1, 2, \ldots, 20\} \)
2. \( \text{back} \in \{1, 2, 3, 4, 16\} \)
3. \( \text{nose} \in \{7, 8, 9\} \)
4. \( \text{back+wings} \in \{1, 2, \ldots, 6, 10, 11, \ldots, 16\} \)
5. \( \text{nose+wings} \in \{5, 6, \ldots, 15\} \)

For triplet \((A, B, C)\) we compute similarity \(s_i(A, B)\) and \(s_i(A, C)\) by aligning the subset \(i\) of landmarks of \(B\) and \(C\) to \(A\) under a translation and scaling that minimizes the sum of squared error after alignment. The similarity is inversely proportional to the residual error. This is also known as “procrustes analysis” commonly used for matching shapes.

B Attributes of Public Figures Face Dataset

Each image in the Public Figures Face Dataset (Pubfig)\(^1\) is characterized by 75 attributes. We used 39 of the attributes in our work and categorized them into 5 groups according to the aspects they describe. Here is a table of the categories and attributes:

| Category      | Attributes                                                                 |
|---------------|---------------------------------------------------------------------------|
| Hair          | Black Hair, Blond Hair, Brown Hair, Gray Hair, Bald, Curly Hair, Wavy Hair, Straight Hair, Receding Hairline, Bangs, Sideburns. |
| Age           | Baby, Child, Youth, Middle Aged, Senior                                    |
| Accessory     | No Eyewear, Eyeglasses, Sunglasses, Wearing Hat, Wearing Lipstick, Heavy Makeup, Wearing Earrings, Wearing Necktie, Wearing Necklace. |
| Shape         | Oval Face, Round Face, Square Face, High Cheekbones, Big Nose, Pointy Nose, Round Jaw, Narrow Eyes, Big Lips, Strong Nose-Mouth Lines. |
| Ethnicity     | Asian, Black, White, Indian                                                |

Table B.1: List of Pubfig attributes that were used in our work.

\(^1\)Available at [http://www.cs.columbia.edu/CAVE/databases/pubfig/](http://www.cs.columbia.edu/CAVE/databases/pubfig/)
Figure A.1: Landmarks illustrated on the several planes