Learning Multiple Maps from Conditional Ordinal Triplets

Dung D. Le and Hady W. Lauw
School of Information Systems, Singapore Management University, Singapore
{ddle.2015, hadywlauw}@smu.edu.sg

Abstract

Ordinal embedding seeks a low-dimensional representation of objects based on relative comparisons of their similarities. This low-dimensional representation lends itself to visualization on a Euclidean map. Classical assumptions admit only one valid aspect of similarity. However, there are increasing scenarios involving ordinal comparisons that inherently reflect multiple aspects of similarity, which would be better represented by multiple maps. We formulate this problem as conditional ordinal embedding, which learns a distinct low-dimensional representation conditioned on each aspect, yet allows collaboration across aspects via a shared representation. Our geometric approach is novel in its use of a shared spherical representation and multiple aspect-specific projection maps on tangent hyperplanes. Experiments on public datasets showcase the utility of collaborative learning over baselines that learn multiple maps independently.

1 Introduction

Increasingly, there are more scenarios where we know some relative comparisons – which object is more similar to another, even as their exact similarities are not known. For instance, [Agarwal et al., 2007; Wills et al., 2009] investigated human perception of “gloss” by studying how human subjects compared images. It is now commonplace to employ human intelligence tasks to generate categorization labels for images [Gomes et al., 2011; Wilber et al., 2014]. [Yue et al., 2014] modeled how different users organized attractions.

Such observations can be represented as object triplets. Observing a triplet $(i,j,k)$ indicates the reference (center) object $j$’s greater similarity to the first-mentioned $i$ than to $k$. The problem of interest is to arrive at object coordinates in a low-dimensional space – effectively a map as the output, such that their relative distances would preserve the observed triplets. This problem is known as ordinal embedding.

The output representation is useful for various applications such as estimation of relative similarities for unseen triplets or “features” for other machine learning tasks. Another important application that we focus on here is visualization on a scatterplot. Without loss of generality, in the subsequent discussion, we will assume 2D for ease of illustration.

Previous works [Terada and Luxburg, 2014; Van der Maaten and Weinberger, 2012] mostly output one visualization map, reflecting a singular similarity perception. However there could be more than one similarity perceptions. For instance, when the triplets have been generated by different human subjects, there may be natural “disagreements” on some triplets. Classically, such disagreements are assumed to be noisy conflicts to be removed in order to uncover the one map.

We postulate that these triplets may be expressing multiple similarity perceptions. The disagreements among triplets reflect idiosyncratic perceptions of similarity. The varying perceptions are valid, and should be preserved by the embeddings. A single visualization map is insufficient to accommodate the different points of view simultaneously. It would be more appropriate to learn multiple maps, each of which reflects a particular perception of similarity.

Hence, we are dealing not with ordinal triplets per se, rather with conditional ordinal triplets of the form $(i,j,k)_t$, expressing relative comparison conditioned on an “aspect” $t$. We refer to the problem of learning multiple maps from such conditional ordinal triples as conditional ordinal embedding, dealing with several ordinal embedding tasks concerning the same universe of objects. As input, we are given conditional ordinal triplets where the associations among triplets to aspects are known. As output, we seek to learn multiple low-dimensional Euclidean maps, one for each aspect.

Contributions. As the first contribution, we propose a collaborative approach to learning multiple maps from conditional ordinal triplets by considering the aspects jointly via a shared representation, while still respecting aspect-specific representations. While the concept of multiple maps has been introduced in different contexts [Van der Maaten and Hinton, 2012; Amid and Ukkonen, 2015], our framework with shared representation is novel. As the second contribution, as a concrete manifestation of the shared representation, we design a novel geometric framework Spherical Conditional Ordinal Embedding or SCORE, based on a spherical representation shared among aspects, while allowing multiple aspect-specific maps as tangent hyperplanes on the sphere. We also describe the learning algorithm and validate our hypothesis via comprehensive experiments on public datasets.
2 Related Work

Existing works in ordinal embedding focus on a single aspect, e.g., Soft Ordinal Embedding or SOE [Terada and Luxburg, 2014], GNMDS [Agarwal et al., 2007], t-Stochastic Triplet Embedding or tSTE [Van der Maaten and Weinberger, 2012], Crowd Kernel Learning [Tamuz et al., 2011]. In contrast, we focus on multi-aspect, and will compare to the latest models SOE and tSTE. [Le and Lauw, 2016] derives one map for multiple types of objects. Riemannian manifold embedding [Wilson et al., 2010], [Wilson and Hancock, 2010] preserves the input distances as geodesic distances on the sphere. We rely on ordinal triplets, not on pairwise distances.

Similarity learning mainly assumes feature vectors are known [Yang, 2007], while we learn only from triplets. Our ordinal embedding formulation enables visualization as one use case. Similarity learning’s use cases are primarily clustering [Yue et al., 2014] or classification [Weinberger and Saul, 2009]. [McFee and Lanckriet, 2011] uses triplets as side information, and still primarily relies on features. [Cheng, 2013] considers similarity between two domains of objects. Conditional Similarity Networks (CSN) [Veit et al., 2017] induces embedding for different similarity notions, but its embedding is learnt from images features, which is unknown to us. [Le and Lauw, 2018] introduces a multiperspective graph-theoretic similarity measure, where the inputs are similarity graphs rather than ordinal triplets.

If we broadly interpret embedding for an aspect as a “task”, our problem is a distinct formulation of multi-task learning [Caruana, 1997]. Other formulations include learning for nearest neighbor [Parameswaran and Weinberger, 2010],[Yang et al., 2013], feature selection [Argyriou et al., 2007], and clustering [Yue et al., 2014].

The term “embedding” is also used in contexts of representation learning such as distributed representation [Bengio et al., 2013] or distributional representation [Blei et al., 2003]. We do not rely on features; rather we learn from similarities or distances (or ordinal comparisons thereof). Ours are low-dimensional Euclidean coordinates that directly support visual analysis, whereas the above works would require a separate method for visualization.

3 Overview

3.1 Problem Formulation

Input. The set of objects of interest is denoted I e.g., images, documents, items, and the set of aspects T. For generality, we assume no feature for an object beyond its identity. An aspect could be a human subject, an attribute, etc., whose perception of similarity is to be modeled. Each aspect t ∈ T observes conditional ordinal triplets in the form of (i, j, k), where (i ≠ j ≠ k) ∈ I. Such a triplet indicates that conditioned on aspect t, j is more similar to i than to k. The set of observed triplets for an aspect t is: N_t = {(i, j, k)|i ≠ j ≠ k ∈ I}. The input is thus N, the union of triplets of all aspects.

Output. For each aspect t ∈ T, we derive an embedding map of all objects. For the map associated with aspect t, every object i ∈ I is associated with a coordinate y_i^t ∈ R^K, where K is the desired dimensionality of the target representation. For visualization purpose, we assume K = 2 in this paper. The objective is to satisfy the following condition for as many triplets in N_t specifically, and N generally, as possible:

\[ \langle i, j, k \rangle_t \in N \iff ||y_j^t - y_i^t|| < ||y_j^t - y_k^t|| \] (1)

3.2 Proposed Methodology

Fig. 1 outlines two approaches for conditional ordinal embedding problem (Eq. 1). The straightforward approach is disjoint learning, i.e., deriving a map for each aspect independently. Specifically, the map m_{t1} is learnt from only aspect t_1’s triplets, and the various maps m_{t1} to m_{tk} are not related (Fig. 1 left).

We believe that the aspects are potentially related as they concern the same set of objects. Their latent relationships could render significant advantage when aspects are sufficiently related, and yet each aspect is under-sampled. In practice, we do not necessarily observe all possible triplets, but a subset. For sparse data, an aspect may have insufficient information. Furthermore, the triplets of any one aspect may not cover all objects [Agarwal et al., 2007].

We propose a collaborative approach (Fig. 1 right). The challenge is to design a shared representation that allows “sharing” across aspects, and yet still allows each aspect to remain distinct. Here, we adopt a well-known instance of Riemannian manifold [Gilkey and others, 1975], namely: hypersphere. Every aspect has a coordinate on this shared hypersphere and its embedding is expressed on the hyperplane tangent at that aspect’s coordinate. Every object also has a coordinate on this shared hypersphere. The projection of the objects’ spherical coordinates onto the tangent hyperplane of a aspect constructs a map that reveals that aspect’s distinct “point of view” or perception of similarity. We elaborate this modeling in the next section.

4 Spherical Conditional Ordinal Embedding

Each aspect t ∈ T and object i ∈ I are respectively associated with a spherical coordinate x_i, y_i ∈ S^K, where S^K = {p ∈ R^{K+1} : ||p|| = 1}. The output coordinate y_i^t ∈ R^K is the projection of y_i onto task-specific K-dimensional hyperplane defined by x_i. The intuition for a sphere as the shared representation is that it allows greater flexibility for each aspect to model its own similarity perception, while still being embedded within the same hyperspace and sharing the same geometric shared representation.
4.1 Model
To arrive at shared $Y = \{y_i : i \in I\}$, while accommodating variances among aspects, we turn to probabilistic modeling.

**Generative Process**
Let us first consider an individual conditional ordinal triplet $(i, j, k) \in \mathcal{N}$. We associate an aspect $t$ and three objects $i$, $j$, and $k$ with a binary-valued random variable $c_{ijk}^t$. When $c_{ijk}^t = 1$, we generate the triplet $(i, j, k) \in \mathcal{N}$, i.e., $t$ considers $j$ to be more similar to $i$ than to $k$. If $c_{ijk}^t = 0$, opposing triplet $(k, j, i) \in \mathcal{N}$ is generated.

There are two views of relative proximity, which determines the outcome of $c_{ijk}^t$. First, there is the aspect-specific view of an aspect $t$, based on the projected coordinates on $t$’s tangent hyperplane, for which the probability is $P_t(c_{ijk}^t = 1 | y_i^t, y_j^t, y_k^t)$. Second, there is the global view, based on coordinates on the shared sphere, for which the probability is $P_s(c_{ijk}^t = 1 | y_i, y_j, y_k)$. We assume that some triplets are aspect-specific, generated according to $P_t$, while other triplets are generic, generated according to $P_s$. The balance between the two is modeled by parameter $\delta_t \in [0, 1]$.

Now we describe the generative process for triplets in $\mathcal{N}$:

1. For each task $t \in \mathcal{T}$:
   - Draw $t$’s coordinate: $x_t \sim vMF (\mu_t, \kappa_t)$
   - Draw $t$’s parameter $\delta_t \sim U(0, 1)$
2. For each object $i \in \mathcal{I}$:
   - Draw $i$’s coordinate: $y_i \sim vMF (\mu_I, \kappa_I)$
3. For objects $i, j, k \in \mathcal{I}$, $i < k$, $j \neq i, k$:
   - If a draw from Bernoulli ($\delta_t$) turns up 1, then:
     
     $c_{ijk}^t \sim Bernoulli \left(P_t(c_{ijk}^t = 1 | y_i^t, y_j^t, y_k^t)\right)$
     Else: $c_{ijk}^t \sim Bernoulli \left(P_s(c_{ijk}^t = 1 | y_i, y_j, y_k)\right)$
   - If $c_{ijk}^t = 1$, generate a triplet instance $(i, j, k)_t$.
     Else: generate a triplet instance $(k, j, i)_t$.

In the above generative process, $x_t$ and $y_i$ have von Mises-Fisher (vMF) [Mardia, 1975] priors, parameterized by mean unit vector $\mu$ and concentration $\kappa$. Higher $\kappa$ translates to greater concentration around $\mu$. $\kappa = 0$ models the uniform prior. In this paper, we assume that $\delta_t$ has a uniform prior.

**Aspect-Specific Probability Function**
Given the shared spherical representation, and the intention to maintain each aspect’s embodiment on a Euclidean space, a natural choice is to have the aspect-specific probability lie on the tangent hyperplane $\mathcal{S}_K^t$ at $x_t$, defined as: $T_{x_t} \mathcal{S}_K^t = \{v \in \mathbb{R}^{K+1} : (x_t)^T v = 0\}$. We define the aspect-specific representation for $t$ to be the projection of objects’ coordinates $\{y_i : i \in I\}$ onto the tangent hyperplane at $x_t$:

$$y_i^t = \text{Proj}_{x_t}(y_i) = \left[I - x_t(x_t)^T\right] y_i$$

where $I$ is $(K+1)$-dimensional identity matrix$^1$.

$^1$Though $\text{Proj}_{x_t}(y_i)$ is a $(K + 1)$-dimensional vector, it still effectively lies on a $K$-dimensional tangent hyperplane in the $(K + 1)$-dimensional space. In the appendix, we describe in detail the $K$-dimensional coordinate transformation.

Fig. 2 illustrates an example of the representations $y_{i}, y_{j}, y_k$ of three objects $i, j, k$ (red points) and $x_{t}, x_{t'}$ of two aspects $t, t'$ (blue points) on the unit sphere. The left tangent hyperplane $T_{x_t} \mathcal{S}_K^t$ corresponds to the representation map of aspect $t$. On this map, $y_i^t$ is closer to $y_k^t$ than to $y_j^t$ through the projection $\text{Proj}_{x_t}$. The right tangent hyperplane $T_{x_{t'}} \mathcal{S}_K^t$ is the representation map of aspect $t'$. There, $y_i^{t'}$ is closer to $y_k^{t'}$ than to $y_j^{t'}$. These are “conflicting” ordinal relationships between $t$ and $t'$, yet they arise from the same spherical coordinates of objects, indicating the role of aspects’ tangent hyperplanes in accommodating different similarity perceptions. There are also triplets which both $t$ and $t'$ agree on.

We now express $P_t$ in terms of such projected distances. Let us denote the $d_{ij}^t$ between two objects $i, j$ on aspect $t$’s map, i.e., $d_{ij}^t = || \text{Proj}_{x_t}(y_j - y_i) ||$. We express $P_t(c_{ijk}^t = 1 | y_i^t, y_j^t, y_k^t)$ in terms of difference between $d_{jk}^t$ and $d_{ij}^t$ (Eq. 3). The smaller is $d_{ij}^t$ relative to $d_{jk}^t$, the higher is the probability. $\alpha$ is the scaling factor.

$$\sigma_{ijk}^t = P_t(c_{ijk}^t = 1 | y_i^t, y_j^t, y_k^t) = \frac{1}{1 + e^{-\alpha (d_{jk}^t - d_{ij}^t)}}$$

**Global Probability Function**
We now describe the “global” probability $P_s$ - the likelihood of observing the triple $(i, j, k)_t$ based on the objects’ spherical coordinates. On the unit sphere, the distance between $y_i$ and $y_j$ is the geodesic distance [Ferreira et al., 2014]:

$$\text{gd}(y_i, y_j) = \cos^{-1}(y_i^T y_j)$$

Given $y_i, y_j, y_k \in \mathcal{S}_K^t$, the following relation holds:

$$\text{gd}(y_i, y_j) < \text{gd}(y_i, y_j) \iff y_i^T y_j > y_k^T y_j$$

On one hand, Eq.4 implies that the inner product yields the same ordering as the geodesic distance. On the other hand, inner product computation is more computationally efficient compared to the geodesic distance. Therefore, the global probability is defined as follows:

$$\sigma_{ijk} = P_s(c_{ijk}^t = 1 | y_i, y_j, y_k) = \frac{1}{1 + e^{-\alpha (y_i^T y_j - y_k^T y_j)}}$$

\[ \text{(5)} \]
Objective Function
The likelihood of observing the triplet \((i, j, k)\) is the normalized weighted sum of \(P_t\) and \(P_s\). The formula is described in Eq. 6 below.

\[
I'_{i,j,k} = \delta_t \sigma_{i,j,k} + (1 - \delta_t) \sigma_{i,j,k}
\]  

(6)

The model’s parameters are learnt to maximize the joint probability \(P(N(X, Y|K_T, \mu_T, \kappa_T, \mu_Z))\) of the model across the observed triplets (Eq. 7), which can be factorized as product of \(P(N_i|X, Y) = \prod_{(i, j, k) \in N_i} I'_{i,j,k}\) - the likelihood, and \(P(X|K_T, \mu_T)\) and \(P(Y|K_Z, \mu_Z)\) - the priors.

\[
\arg \max_{X, Y} \prod_{i \in T} P(N_i|X, Y) \times P(X|K_T, \mu_T) \times P(Y|K_Z, \mu_Z)
\]

(7)

Maximizing the joint probability in Eq. 7 is equivalent to maximizing its logarithm \(\mathcal{L}\) (to simplify the parameters, we tie \(K_T = K_Z = \kappa\) and \(\mu_T = \mu_Z = \mu\):

\[
\mathcal{L} = \sum_{i \in T} \ln P(N_i|X, Y) + \ln P(X|\kappa, \mu) + \ln P(Y|\kappa, \mu)
\]

\[
\propto \sum_{i \in T} \sum_{(i, j, k) \in N_i} \ln I'_{i,j,k} + \sum_{i \in T} \kappa \cdot \mu^T x_i + \sum_{i \in T} \kappa \cdot \mu^T y_i.
\]

4.2 Parameter Learning
Line Search on Manifold
The learning requires solving an optimization problem on the spherical manifold. [Absil et al., 2009] presents the line-search method on a manifold. The update formula is:

\[
x_{k+1} = \mathcal{R}_{x_k}(t_k \eta_k), \quad \kappa_{x_k} = \kappa_{x_k} t_k, \quad \eta_k \in \mathcal{T}_{x_k} M \text{ are the retraction map at } x_k, \text{ the step size, and the search direction respectively. Retraction map ensures the update process to be performed on the manifold. Here, we consider the following map [Bonnabel, 2013]:}
\]

\[
\mathcal{R}_x(\eta) = \arg \min_{y \in \mathbb{S}^K} \|x + \eta - y\| = \frac{x + \eta}{\|x + \eta\|}
\]

(8)

For parameter learning, we adopt the stochastic gradient descent strategy for functions defined on a Riemannian manifold [Bonnabel, 2013], which requires the computation of the Riemannian gradient. According to [Ferreira et al., 2014], the gradient on the sphere of a differentiable function \(f: \Omega \to \mathbb{R}\) (let \(\Omega \in \mathbb{S}^K\) be an open set, at \(x \in \Omega\) is defined by:

\[
\nabla f(x) = [I - xx^T] \nabla f(x),
\]

(9)

where \(\nabla f(x)\) is the usual gradient of \(f(x)\) in \(x \in \Omega\).

Learning Algorithm
Algorithm 1 shows that in each iteration, a triplet \((i, j, k)\) is randomly selected, and the parameters are updated using the line-search optimization technique on the unit sphere. Specifically, we first compute the partial derivatives with respect to \(x_i, y_i, y_j, y_k\) (line 7). The gradients on the spherical surface are computed through the project map \(\text{Proj}()\). Then we update the model parameters using the retraction map as described earlier (line 9). Learning rate \(\epsilon\) is decayed over time. The last update in line 10 guarantees that \(\delta_t \in [0, 1]\). The complexity is linear to the size of \(N\)-the set of all triplets, which is bounded by \(O(|T| \times |\mathbb{I}|^3)\).

Algorithm 1 SCORE

1: Initialize \(x_i\) for \(t \in T\) and \(y_i\) for \(i \in I\).
2: While not converged:
3: Draw a triplet \((i, j, k)\) randomly from \(N\).
4: Compute the likelihood:
5: \(I'_{i,j,k} = \delta_t \sigma_{i,j,k} + (1 - \delta_t) \sigma_{i,j,k}\).
6: Compute the partial derivatives:
7: \(\delta_{t} \leftarrow \arg \min_{\delta_t} \|\delta_t - \delta\|\) for each \(z \in \{x_i, y_i, y_j, y_k\}\);
8: Update the model parameters:
9: \(t \leftarrow t + \epsilon (\sigma_{i,j,k} - \sigma_{i,j,k})\); \(t \leftarrow \arg \min_{\delta_t} \|\delta_t - \delta\|\);
10: Return \(\{x_i\}_{i \in T}\) and \(\{y_i\}_{i \in I}\).

5 Experiments
Our objective is primarily to investigate the effectiveness of multiple maps for conditional or dinal embedding.

5.1 Experimental Setup
Datasets
We experiment with three public datasets that could model varying perceptions of similarity.

- **Zoo** contains 17 attributes of 101 animals (excluding animal name). We model each attribute as a similarity aspect. For attribute \(t\), we form the triplet \(\langle i, j, k \rangle\) if \(i\) and \(j\) have the same attribute value, which is different from \(k\). There are \(3.24 \times 10^7\) triplets.

- **Congressional Voting Records (or HouseVote)** contains 435 instances (congressmen) and 16 attributes (voting issues). After excluding instances with missing values, we get 232 fully-observed instances of 16 attributes. We generate triplets in the same way as we do with Zoo dataset. That induces totally \(2.4 \times 10^7\) triplets.

- **Paris Attractions** contains 237 users organizing 250 Paris attractions into clusters. With user as aspect, we induce \(3.48 \times 10^6\) triplets, each involves two attractions \(i\) and \(j\) that the user puts into the same cluster, and another attraction \(k\) in a different cluster. As in [Yue et al., 2014], we exclude attractions uninteresting to users.

Comparative Methods
We compare SCORE to several baselines. The disjoint learning approach (Section 3) learns a distinct map from the triplets of an aspect. We use two recent ordinal embedding methods: SOE [Terada and Luxburg, 2014] and iSTE [Van der Maaten and Weinberger, 2012].

Multiview Triplet Embedding (MVTE) [Amit and Ukkonen, 2015] divides one pool of triplets into clusters. The number of views is set to the number of aspects, i.e., \(|T|\). Since the aspect-triplet associations are unknown, we match each
view with a ground-truth attribute using Hungarian maximum bipartite matching algorithm, so as to maximize the accuracy. Multiview Multidimensional Scaling (MVMDS) [Bai et al., 2017] performs MDS on multi-view data, learns the weights of these views, and produces one consensus map. This is akin to learning a single map by consolidating multiple views. Since MVMDS expects distance matrices, we feed it feature vectors learnt by SOE from the ordinal triplets.

For visualization purpose, we set the dimensionality of the embedding space $K = 2$. We tune the parameters of all methods for their best performances on the training data. For SCORE, the setting is $\kappa = 10^{-3}$ for Paris Attractions, and 0 for Zoo and HouseVote, VMF mean vector $\mu = (0, 0, 1)$, the learning rate $\epsilon = 0.05$, and the scaling factor $\alpha = 30$. For SOE, the scaling factor is 0.1 for all the datasets. For $t$STE, the learning rate and regularization parameter are 2 and 0 respectively for all datasets. For MVTE, the learning rate is 1 for all datasets. For MVMDS, $\gamma = 5$ for all datasets.

### Evaluation Measure

The preservation accuracy for an aspect $t$ is the fraction of its ordinal triplets $N_t$ for which $t$’s coordinates reflect the correct direction. The fewer the violated triplets, the higher is the accuracy. The overall accuracy is the average of aspects’ preservation accuracies (Eq. 10):

$$\frac{1}{|T|} \sum_{t \in T} \frac{|\{(i,j,k) \in N_t : |y^t_i - y^t_j| < |y^t_j - y^t_k|\}|}{|N_t|}, \quad (10)$$

$y^t_i, y^t_j, y^t_k$ are $t$’s embedding coordinates of objects $i, j, k$.

Since in practice we may not observe all triplets or even all objects beforehand, we sample a fraction $r$ (split ratio) of objects for each aspect, then evaluate the coordinates against the full set of triplets. As the default for this study, we set $r = 0.5$, which has a relative balance between the information that an aspect sees and the information that it could learn from others. Later we will also investigate the effects of different $r$ values. We average the results across 30 random samples.

The running times are reasonable. For the Paris Attractions, including all aspects, SCORE takes 5 minutes on a PC with Intel Core i5 3.2 GHz CPU and 12 GB RAM. The learning times for the Zoo and HouseVote are less than 10 minutes.

#### 5.2 Comparison to Baselines

We vary the number of aspects by randomly sampling aspects. Fig. 3 shows the preservation accuracies of all models.

### Overall Preservation Accuracy

SCORE shows significantly higher performance than SOE, $t$STE (Fig. 3(a,b,c)). In the latter, an aspect cannot collaborate with other aspects, leading to poor performance on unseen triplets. This highlights the benefit of collaborative learning, as it helps aspects fill in each other’s missing information. MVTE and MVMDS show even weaker performances. For MVTE, the likely reason is the lack of information about the associations between aspects and triplets. If this information is provided, MVTE reduces to $t$STE, which is showing relatively higher performance than MVTE. For MVMDS, the likely reason is the consolidation into a single map, which, though learnt from multiple distance matrices, cannot fit conflicting triplets.

### Performance on Predicting Hidden Triplets

The earlier accuracies are evaluated for the full set of triplets, which is the combination of the observed subset (from the $r$ fraction) and the hidden subset (the unseen triplets). To see how well the models generalize to the unseen data, we now investigate the preservation accuracy measured on the hidden alone (Fig. 3(d,e,f)). We observe the same picture as before but with generally lower accuracies than that for full sets (Fig. 3(a,b,c)), which are expected as these are unseen triplets. The reduction is more dramatic for SOE and $t$STE, which tend to overfit the observed, and generalize poorly to the hidden triplets. SCORE does commendably well on the hidden set, showing greater robustness in generalizing to the unseen triplets.

For an alternative measure of generalization, we test the learnt coordinates as features to classify the hidden objects by attribute values associated with the aspect. An object is assigned the majority label among its 10-nearest neighbors. Fig. 4 shows the 10-NN classification accuracy, averaged across aspects. Only Zoo and HouseVote have “labels” and are involved in this experiment. SCORE has better results than the baselines (statistically significant at 0.05) in pre-
dicting the labels of unseen instances. Interestingly, MVTE performs better than disjoint learning baselines in this task. Since triplets are learned jointly, some triplets may have been assigned to clusters correlated with the class labels, though the clusters may not reflect the aspect-specific view perfectly.

**Exploration on the Split Ratio**

To better understand the benefits of multi-aspect modeling, we show the accuracies with varying $r$ for the complete set of aspects in Fig. 5.

The disjoint learning baselines perform poorly for low value of $r$. This is expected since the amount of observed data is insufficient for a single task to learn its own map effectively. For extremely high $r$, e.g., 0.7, the disjoint learning baselines tend to do well. For Zoo, HouseVote, and Paris Attractions, $r = 0.7$ respectively corresponds to approximately 1.1M, 16.1M, and 155K triplets in training, which are 34.23%, 34.17%, and 44.42% of all possible triplets. With sufficiently large data that cover majority of objects, each aspect has more flexibility to specialize, with little risk in missing out information. Also in Fig. 5, SCORE shows significantly better performances than MVTE and MVMDS for the same reasons that have been discussed in the first experiment.

Importantly, SCORE is robust across values of $r$. It is the best around 0.2-0.6, and never the worst. This result has two implications. First, it reiterates the benefit of collaborative approach when the data is under-sampled, yet sufficient to learn the relatedness and specialization of tasks. Second, in practice it is often unclear whether the data is sufficient. Upon such ambiguity, multi-aspect modeling ameliorates the risk of performing badly, while providing reasonable performance.

### 5.3 Aspect Relatedness

Two similar aspects would be expected to be closer on the hypersphere than two dissimilar aspects. For Zoo and HouseVote, each aspect corresponds to an attribute, whose values effectively define a clustering of objects. We define the attribute-based similarity between two aspects as the Normalized Mutual Information or NMI [Estévez et al., 2009] between the two clusterings. We also define the proximity between two aspects on the shared hypersphere as their angular similarity. For each aspect $t \in T$, we measure the similarities between two aspects on the shared hypersphere as their angular similarity. For each aspect $t \in T$, we measure the

**Exploration on the Split Ratio**

To better understand the benefits of multi-aspect modeling, we show the accuracies with varying $r$ for the complete set of aspects in Fig. 5.

The disjoint learning baselines perform poorly for low value of $r$. This is expected since the amount of observed data is insufficient for a single task to learn its own map effectively. For extremely high $r$, e.g., 0.7, the disjoint learning baselines tend to do well. For Zoo, HouseVote, and Paris Attractions, $r = 0.7$ respectively corresponds to approximately 1.1M, 16.1M, and 155K triplets in training, which are 34.23%, 34.17%, and 44.42% of all possible triplets. With sufficiently large data that cover majority of objects, each aspect has more flexibility to specialize, with little risk in missing out information. Also in Fig. 5, SCORE shows significantly better performances than MVTE and MVMDS for the same reasons that have been discussed in the first experiment.

Importantly, SCORE is robust across values of $r$. It is the best around 0.2-0.6, and never the worst. This result has two implications. First, it reiterates the benefit of collaborative approach when the data is under-sampled, yet sufficient to learn the relatedness and specialization of tasks. Second, in practice it is often unclear whether the data is sufficient. Upon such ambiguity, multi-aspect modeling ameliorates the risk of performing badly, while providing reasonable performance.

**Exploration on the Split Ratio**

To better understand the benefits of multi-aspect modeling, we show the accuracies with varying $r$ for the complete set of aspects in Fig. 5.

The disjoint learning baselines perform poorly for low value of $r$. This is expected since the amount of observed data is insufficient for a single task to learn its own map effectively. For extremely high $r$, e.g., 0.7, the disjoint learning baselines tend to do well. For Zoo, HouseVote, and Paris Attractions, $r = 0.7$ respectively corresponds to approximately 1.1M, 16.1M, and 155K triplets in training, which are 34.23%, 34.17%, and 44.42% of all possible triplets. With sufficiently large data that cover majority of objects, each aspect has more flexibility to specialize, with little risk in missing out information. Also in Fig. 5, SCORE shows significantly better performances than MVTE and MVMDS for the same reasons that have been discussed in the first experiment.

Importantly, SCORE is robust across values of $r$. It is the best around 0.2-0.6, and never the worst. This result has two implications. First, it reiterates the benefit of collaborative approach when the data is under-sampled, yet sufficient to learn the relatedness and specialization of tasks. Second, in practice it is often unclear whether the data is sufficient. Upon such ambiguity, multi-aspect modeling ameliorates the risk of performing badly, while providing reasonable performance.
same number of legs tend to be found together, e.g., 6 legs on the top left, 2 legs on the bottom right. Fig. 7(c) visualizes animals based on whether they are predator (shape). The binary separation of predators (triangles) on the top right and non-predators (circles) on the lower left is evident. A single map cannot capture diverse perceptions of similarity. Fig. 7(d) depicts the single map mode for the same three attributes. It could only represent separation by predator (shape), but is unable to represent type or #legs well.

6 Conclusion

In this work, we formulate the problem of ordinal embedding involving ordinal comparisons from multiple aspects as conditional ordinal embedding. Our proposed geometric approach seeks to represent aspects and objects on a shared hypersphere, as well as on aspect-specific tangent hyperplanes. Experiments on public datasets show that the proposed framework is robust, and particularly beneficial when there is variance across tasks yet insufficient data to learn each task separately, thus collaboration across tasks is helpful.

Acknowledgements

This research is supported by the National Research Foundation, Prime Minister’s Office, Singapore under its NRF Fellowship Programme (Award No. NRF-NRFF2016-07).

<table>
<thead>
<tr>
<th>SCORE</th>
<th>Zoo</th>
<th>HouseVote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Map</td>
<td>0.82</td>
<td>0.70</td>
</tr>
<tr>
<td>Multiple Maps</td>
<td>0.98</td>
<td>0.89</td>
</tr>
</tbody>
</table>

Table 1: Performance of SCORE: multi-maps vs. single-map

Figure 8: Transformation of objects’ coordinates from 3D to 2D

A K-dimensional Coordinate Transformation

A necessary step is to transform the \((K + 1)\)-dimensional coordinate of objects on \(t\)’s tangent hyperplane, i.e., \(\{\text{Proj}_{x_{t}}(y_{i})\}_{i \in I}\), to their corresponding \(K\)-dimensional coordinates, i.e., \(\{y_{t}^{i}\}_{i \in I}\). For the purpose of visualizing the embedding for an aspect \(t\) on a scatterplot, we describe how to transform the 3D coordinates of objects on \(t\)’s tangent hyperplane to their corresponding 2D coordinates in the following. However, the below analysis is also applicable to high-dimensional embedding space, i.e., when \(K > 2\).

Since \(x_{t}, \text{Proj}_{x_{t}}(y_{i}), \text{Proj}_{x_{t}}(y_{j}), \text{Proj}_{x_{t}}(y_{k})\) lie on the tangent hyperplane \(T_{x_{t}}, S^{K}\) of the task \(t\), the three vectors: \(u = \text{Proj}_{x_{t}}(y_{i}) - x_{t}; v = \text{Proj}_{x_{t}}(y_{j}) - x_{t}; w = \text{Proj}_{x_{t}}(y_{k}) - x_{t}\) are on \(T_{x_{t}}, S^{K}\) as well.

As illustrated in Fig. 8, the cross product \(x_{t} \times u\) is a vector on \(T_{x_{t}}, S^{K}\) and perpendicular to \(x_{t}, u\). Let’s denote:

\[ e_{1} = \frac{u}{||u||}, e_{2} = \frac{x_{t} \times u}{||x_{t} \times u||}. \]

We can see that \(e_{1}, e_{2}\) form a basis of \(T_{x_{t}}, S^{K}\) (since \(||e_{1}|| = ||e_{2}|| = 1, e_{1}^T e_{2} = 0\)). From linear algebra, for each point \(y \in T_{x_{t}}, S^{K}\), there exists unique \(a_{y}, b_{y} \in \mathbb{R}\) such as:

\[ (y - x_{t}) = a_{y} e_{1} + b_{y} e_{2}. \]

Consider the following transformation map where \(a_{y}, b_{y} \in \mathbb{R}\) are defined as above:

\[ \text{Tr}_{t} : T_{x_{t}}, S^{K} \rightarrow \mathbb{R}^{2}, \quad y \mapsto \text{Tr}_{t}(y) = (a_{y}, b_{y}) \quad (11) \]

Let \((a_{j}, b_{j})\) and \((a_{k}, b_{k})\) be the transformation of \(\text{Proj}_{x_{t}}(y_{j})\) and \(\text{Proj}_{x_{t}}(y_{k})\) respectively:

\[ ||\text{Proj}_{x_{t}}(y_{j}) - \text{Proj}_{x_{t}}(y_{k})|| = ||v - w|| = ||(a_{j} e_{1} + b_{j} e_{2}) - (a_{k} e_{1} + b_{k} e_{2})|| = ||(a_{j} - a_{k}) e_{1} + (b_{j} - b_{k})e_{2}|| = \sqrt{(a_{j} - a_{k})^{2} + (b_{j} - b_{k})^{2}} = ||\text{Tr}_{t}(\text{Proj}_{x_{t}}(y_{j})) - \text{Tr}_{t}(\text{Proj}_{x_{t}}(y_{k}))||. \quad (12) \]

Equation 12 implies that the \(L_{2}\)-norm between points on \(T_{x_{t}}, S^{K}\) are preserved through the transformation map \(\text{Tr}_{t}\). Therefore, the ordinal relations between points are also preserved through the transformation. Hence, we express \(y_{t}^{i} = \text{Tr}_{t}(\text{Proj}_{x_{t}}(y_{i}))\), for all \(i \in I\).
References


