Multiview Triplet Embedding:
Learning Attributes in Multiple Maps

Ehsan Amid   EHSAN.AMID@AALTO.FI
Aalto University and Helsinki Institute for Information Technology HIIT, Finland

Antti Ukkonen   ANTTI.UKKONEN@TTL.FI
Finnish Institute of Occupational Health, Helsinki, Finland

Abstract

For humans, it is usually easier to make statements about the similarity of objects in relative, rather than absolute terms. Moreover, subjective comparisons of objects can be based on a number of different and independent attributes. For example, objects can be compared based on their shape, color, etc. In this paper, we consider the problem of uncovering these hidden attributes given a set of relative distance judgments in the form of triplets. The attribute that was used to generate a particular triplet in this set is unknown. Such data occurs, e.g., in crowdsourcing applications where the triplets are collected from a large group of workers.

We propose the Multiview Triplet Embedding (MVTE) algorithm that produces a number of low-dimensional maps, each corresponding to one of the hidden attributes. The method can be used to assess how many different attributes were used to create the triplets, as well as to assess the difficulty of a distance comparison task, and find objects that have multiple interpretations in relation to the other objects.

1. Introduction

High-dimensional data can be analyzed by first embedding it into a low-dimensional space (Kruskal, 1964; Tenenbaum et al., 2000; Belkin & Niyogi, 2003; Saul & Roweis, 2003). A usual input to such methods is a distance matrix of the items, and the objective is to create an embedding that aims to preserve these distances as well as possible. However, eliciting absolute distance informa-

tion can be hard in a number of cases. This is especially true if the judgments must be collected from human evaluators. Therefore, a number of recent methods, such as Generalized Non-metric Multidimensional Scaling (GNMDS) (Agarwal et al., 2007), the Crowd Kernel algorithm (Tamuz et al., 2011), and Stochastic Triplet Embedding (STE) (van der Maaten & Weinberger, 2012), only use relative distance judgments, or some other type of qualitative information (Gomes et al., 2011). These are more amenable to applications in, e.g., crowdsourcing and human computation.

Relative distances are often collected in the form of *triplets*, where the evaluator must answer the following task:

> *"Which of the items A and B is closer to item X?"*

A common problem when collecting such data is that the evaluators may provide inconsistent answers. Someone might say that A is closer to X, while somebody else might says that B is closer. A lot of research on human computation tends to make the assumption that the tasks have a single correct solution, and all other solutions are incorrect. This is clearly a good approach in some applications, e.g., labeling tasks where the items unambiguously either do or do not satisfy some property. In such cases, it is important to aggregate the solutions of a single task to provide the most probable correct answer (Dawid & Skene, 1979; Whitehill et al., 2009; Raykar & Yu, 2012).

However, with some tasks, the situation can be more ambiguous. Consider the following toy example in the context of the comparison task given above. We are given a set of objects, each having two attributes: shape ($\circ$ or $\times$) and color ('red' or 'green'). The user is asked to compare item X = 'a red $\circ$', with item A = 'a red $\times$' and item B = 'a green $\circ$'. We argue that A and B are both correct answers depending on the point of view taken by the evaluator. In the absence of more precise instructions, the decision can
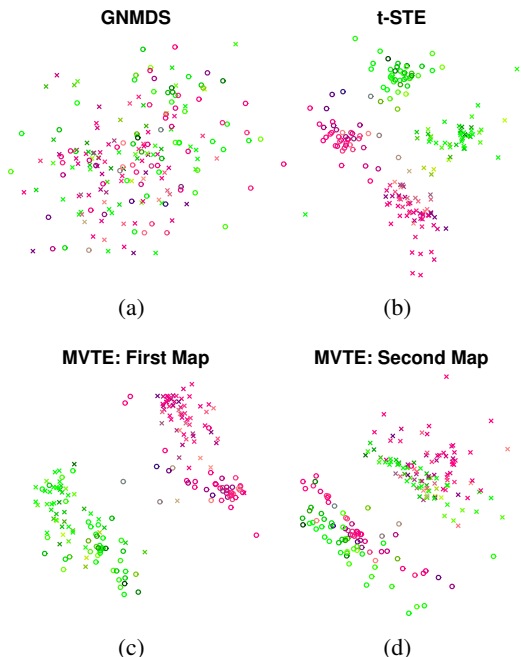
Figure 1. Maps produced by GNMDS, t-STE and MVTE (proposed in this paper) given a toy data with objects having two attributes (shape, color).

be made based on either of the two attributes. If the evaluator uses color as a deciding factor, A is the correct choice, while if the evaluator uses shape, B is an appropriate answer.

In general, our input thus contains a mixture of triplets where the workers may have used different attributes of the items when making their comparisons. Figure 1 shows embeddings produced by the GNMDS and t-STE methods[1], as well as the MVTE method that we propose, given such triplets from the simple 'x○' toy data described above. GN-MDS and t-STE collapse all items into a single map, and consequently may fail to identify the original attributes. For instance, the t-STE method appropriately divides the objects into four subclusters, but neither of the dimensions of the embedding corresponds to shape or color. However, MVTE successfully separates the underlying attributes using *two different maps*. Distances in the first map are based on color, while distances in the second map reflect the shape.

In this paper, we take thus the position that inconsistent answers to individual tasks should not necessarily be aggregated into a single consensus solution. Instead, we consider all solutions as potentially correct. Rather than trying to learn a single low-dimensional map for the items, we propose to *simultaneously learn a number of maps that all*

aim to represent one possible attribute of the input space, as shown in Figure 1. A similar problem was considered by (Changpinyo et al., 2013) in the context of metric learning using pairwise similarity comparisons. Observe that this problem, in general, cannot be solved simply by increasing the dimensionality of the output space. In the example above, no matter what dimensionality we use, one of the solutions (A vs. B) would always be unsatisfied in terms of any distance metric.

Our contributions: We propose the Multiview Triplet Embedding (MVTE) algorithm for learning multiple maps from a given set of triplets. We propose a number of applications of the algorithm, and conduct experiments that show how the method can be used to *identify tasks and items that are confusing to the workers*, as well as to *identify the attribute each worker mainly uses* when comparing the items.

2. Multiview Triplet Embedding

In this section, we define the problem of finding multiple embeddings[2] given a set of triplets that originate from a number of different views, as well as describe the MVTE algorithm.

2.1. Problem Formulation

We define a *query* as the triad $(i | j, k)$ of items where i is called the *probe* item and j and k are called the *test* items. The query is a question of the form: "Is i more similar to j or k?". An answer to the query is called a *triplet*. We denote a triplet by the ordered tuple (i, j, k), meaning that "i is closer to j than k". Let $\mathcal{T} = \{(i, j, k)\}$ denote the set of triplets provided for a set of N different items in M different *views* $\mathcal{V} = \{\mathbf{V}^m\}_{m=1}^M$. Let $\mathcal{X}^m = \{\mathbf{x}_1^m, \mathbf{x}_2^m, \ldots, \mathbf{x}_N^m\}$ denote the representation of the items in the mth view. Each view $\mathbf{V}^m$ represents a particular attribute (or aspect) of the data, e.g., shape, orientation, color, semantics, etc. and $\mathcal{X}^m$ denotes the placements of the items with respect to that attribute. Each triplet (i, j, k) specifies the relative distances of the query items in (at least) one of the views $\mathbf{V}^m \in \mathcal{V}$. That is, the inequality

$$d^m(\mathbf{x}_i^m, \mathbf{x}_j^m) < d^m(\mathbf{x}_i^m, \mathbf{x}_k^m) \qquad (1)$$

is satisfied with respect to the distance function d^m for some $m \in \{1, 2, \ldots, M\}$. However, the same triplet may also be satisfied in some of the other spaces, as there might exist some correlation among different attributes of the objects. Therefore, a particular triplet might happen to be satisfied in more than one, only one, or none of the provided metric spaces (if there is noise).

---

[1]We only consider GNMDS and t-STE because other approaches are similar to one of these two techniques.

[2]Please note that we use the words embedding and map interchangeably throughout this paper.
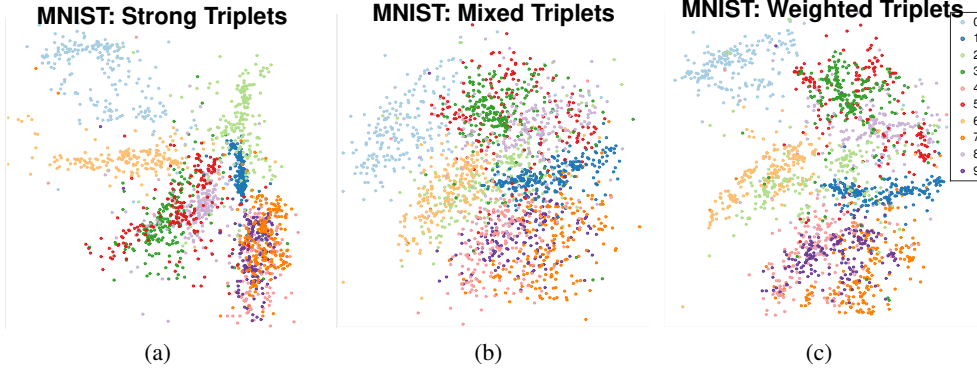
Figure 2. Embedding results of the MNIST dataset using different sets of triplets: results of the t-STE method using (a) sharp triplets only, (b) sharp triplets mixed with weak triplets, and (c) the MVTE method ($M = 1$) using the same set of mixed triplets weighted by the satisfiability ratios.

Problem: Given the set of triplets $\mathcal{T}$, our goal is to find M different embeddings of the items $\mathcal{Y}^m = \{\mathbf{y}_1^m, \mathbf{y}_2^m \ldots, \mathbf{y}_N^m\}$ such that for each triplet (i, j, k), the distance constraint is satisfied with respect to the Euclidean norm in the corresponding map[3]. In other words,

$$d^m(\mathbf{x}_i^m, \mathbf{x}_j^m) < d^m(\mathbf{x}_i^m, \mathbf{x}_k^m)$$
$$\Updownarrow \qquad (2)$$
$$\|\mathbf{y}_i^m - \mathbf{y}_j^m\| < \|\mathbf{y}_i^m - \mathbf{y}_k^m\|$$

for all $(i, j, k) \in \mathcal{T}$.

2.2. The MVTE algorithm

To overcome the problem of having triplets from different views, we consider *a mixture of maps* as follows. For each triplet (i, j, k), we define p_{ijk}^m as the probability that it is satisfied in map m. We adopt the formulation similar to (van der Maaten & Weinberger, 2012), that is

$$p_{ijk}^m = \frac{\exp(-\|y_i^m - y_j^m\|^2)}{\exp(-\|y_i^m - y_j^m\|^2) + \exp(-\|y_i^m - y_k^m\|^2)}. \qquad (3)$$

We denote by $\mathbf{z}_{ijk}$ the binary indicator vector of length M for the triplet (i, j, k) having all values equal to zero except one, specifying the corresponding view that it originates from. Thus, the probability that the triplet (i, j, k) is satisfied in the corresponding view can be written as

$$p_{ijk} = \prod_{m=1}^M (p_{ijk}^m)^{z_{ijk}^m}, \qquad (4)$$

where z_{ijk}^m is the mth component of the binary vector $\mathbf{z}_{ijk}$. Now, our objective becomes to maximize the sum of the

---

[3]Note that in general, the correspondence between the maps and the views might be randomly permuted since the comparisons are provided as a set of unlabeled triplets. However, we use identical indices for notational simplicity.
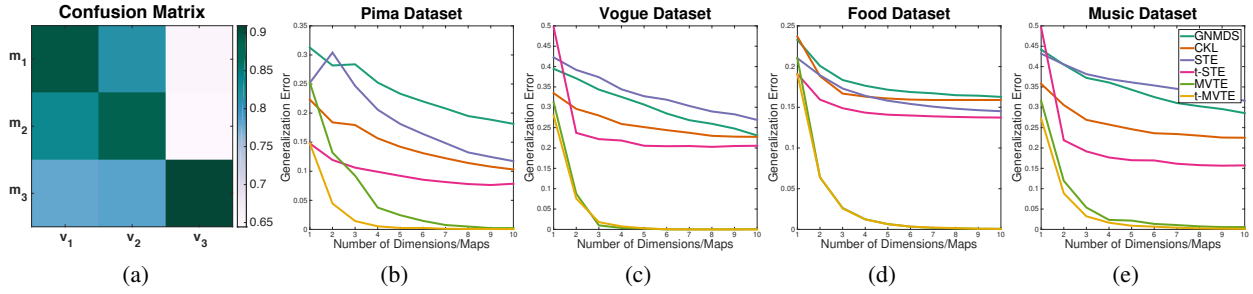
log-probabilities over all triplets, that is

$$\max_{\mathcal{Y}} \sum_{(i,j,k) \in \mathcal{T}} \log p_{ijk} = \max_{\mathcal{Y}} \sum_{(i,j,k) \in \mathcal{T}} \sum_{m=1}^M z_{ijk}^m \log p_{ijk}^m. \qquad (5)$$

The objective function can be optimized using a standard iterative gradient ascent algorithm on the map points y_i^m.

Now, there still remains the problem of estimating the latent indicator variables $\mathbf{z}_{ijk}$ in (5). The naïve approach to maximize (5) w.r.t. $\mathcal{Z} = \{\mathbf{z}_{ijk}\}$ would be to set each $\mathbf{z}_{ijk}$ as the indicator of the probability p_{ijk}^m which has the largest value among all the maps. However, as we stated earlier, each triplet may be satisfied in more than one view. Therefore, restraining the indicator variable to a single map prevents using the triplet information when forming the maps that correspond to other views. We consider a triplet to be more informative for $\mathbf{V}^m$ if it is strongly satisfied in $\mathbf{V}^m$, *and* it is only weakly satisfied or entirely unsatisfied in the other views. Therefore, it must be given more emphasis when finding the map for view $\mathbf{V}^m$. To formulate the importance of a triplet, we define the *satisfiability ratio* for triplet (i, j, k) in view $\mathbf{V}^m$ as

$$\Gamma_{ijk}^m = \frac{d^m(\mathbf{x}_i^m, \mathbf{x}_k^m)}{d^m(\mathbf{x}_i^m, \mathbf{x}_j^m)}. \qquad (6)$$

A value of $\Gamma_{ijk}^m > 1$ ($\Gamma_{ijk}^m \leq 1$) indicates that the triplet is satisfied (unsatisfied) in view $\mathbf{V}^m$. We similarly define

$$\gamma_{ijk}^m = \frac{\|\mathbf{y}_i^m - \mathbf{y}_k^m\|}{\|\mathbf{y}_i^m - \mathbf{y}_j^m\|} \qquad (7)$$

as the satisfiability ratio in the corresponding map.

We consider a simple example that illustrates how the satisfiability ratio can be used to assess the importance of a triplet. We use a subset of 2000 datapoints from the MNIST

Figure 3. Triplet satisfiability results: (a) confusion matrix for Pima dataset using MVTE with $M = 3$, and the generalization error of different algorithms on (b) Pima, (c) Vogue, (d) Food, and (e) Music datasets.

Algorithm 1 Multiview Triplet Embedding (MVTE)

Input: set of triplets $\mathcal{T}$, number of views M
Output: embeddings of datapoints $\mathcal{Y} = \{\mathcal{Y}^m\}_{m=1}^M$, indicator variables of the triplets $\mathcal{Z}$
Initialize $\mathcal{Y}$ and $\mathcal{Z}$.
repeat
  1) Update (5) w.r.t $\mathcal{Y}$ with $\mathcal{Z}$ fixed
  2) Update $\mathcal{Z}$ using (9)
until (5) does not change significantly

---

dataset (LeCun & Cortes, 1999). We first build a map using the t-STE algorithm by considering a set of 20,000 strongly satisfied synthetic triplets (see Figure 2(a)). The quality of the same map reduces significantly when we append another set of 20,000 weakly satisfied triplets and re-compute the embedding (Figure 2(b)). To increase the effect of strong triplets, we calculate the satisfiability ratios of the triplets in map (b), and reconstruct the map by *weighting the triplets by the estimated satisfiability ratios* (using the MVTE algorithm). As can be seen, the result is improved by assigning a higher weight to the strongly satisfied triplets (Figure 2(c)). The satisfiability ratio is thus a reasonable way to assign weights to the triplets. This result also suggests that the satisfiability ratios in the learned map may provide an appropriate estimate of the true satisfiability ratios in the original view.

We make use of these findings when estimating the indicator variables for the triplets in different views as follows. First, we define the *biased satisfiability ratio* for triplet (i, j, k) as

$$\tilde{\gamma}_{ijk}^m = \begin{cases} \gamma_{ijk}^m & \text{if } \gamma_{ijk}^m > 1 \\ 0 & \text{otherwise} \end{cases} . \tag{8}$$

The biased ratio assigns no importance to the unsatisfied triplets while it preserves the satisfiability ratio if the triplet is satisfied. We define the indicator variable z_{ijk}^m so that it reflects the extent to which the triplet (i, j, k) is satisfied in

the mth map. In particular, we let

$$z_{ijk}^m = \frac{\tilde{\gamma}_{ijk}^m}{\max\left(\sum_{l=1}^M \tilde{\gamma}_{ijk}^l, 1\right)} . \tag{9}$$

In this way, we assign higher weights to the informative triplets while neglecting the unimportant (and possibly unsatisfied) ones. Therefore, the triplets are distributed among the maps with different weights, where the total weights for each sum up to one. The algorithm for finding the maps proceeds by alternatively maximizing (5) w.r.t. $\mathcal{Y}$ and then, updating the indicator variables using (9). The pseudocode for the algorithm is shown in Algorithm 1. We refer to our proposed method as Multiview Triplet Embedding (MVTE). The algorithm can be extended to distributions having heavier tail than Gaussian, e.g., Student t-distribution which leads to the t-distributed MVTE algorithm, or t-MVTE, in short.

3. Applications

Next, we describe a number of novel applications where we can use the MVTE algorithm. We assume that the method is applied in a crowdsourcing context where the triplets are elicited from a number of *workers*.

Number of attributes: As the first result, our method can be used to estimate the true *number of attributes in a set of triplets*. We want to point out that this is not the same as the dimensionality of a single input space, as each attribute induces an independent view of the items. The effect of having several attributes can be investigated by increasing the dimensionality of a single map and comparing the results with those obtained from multiple maps that each have a fixed number of dimensions. As we will see in the next section, the limiting factor for satisfying the triplets in most cases is the assumption of a single view (or distance function) for the items rather than the number of dimensions of the resulting map.
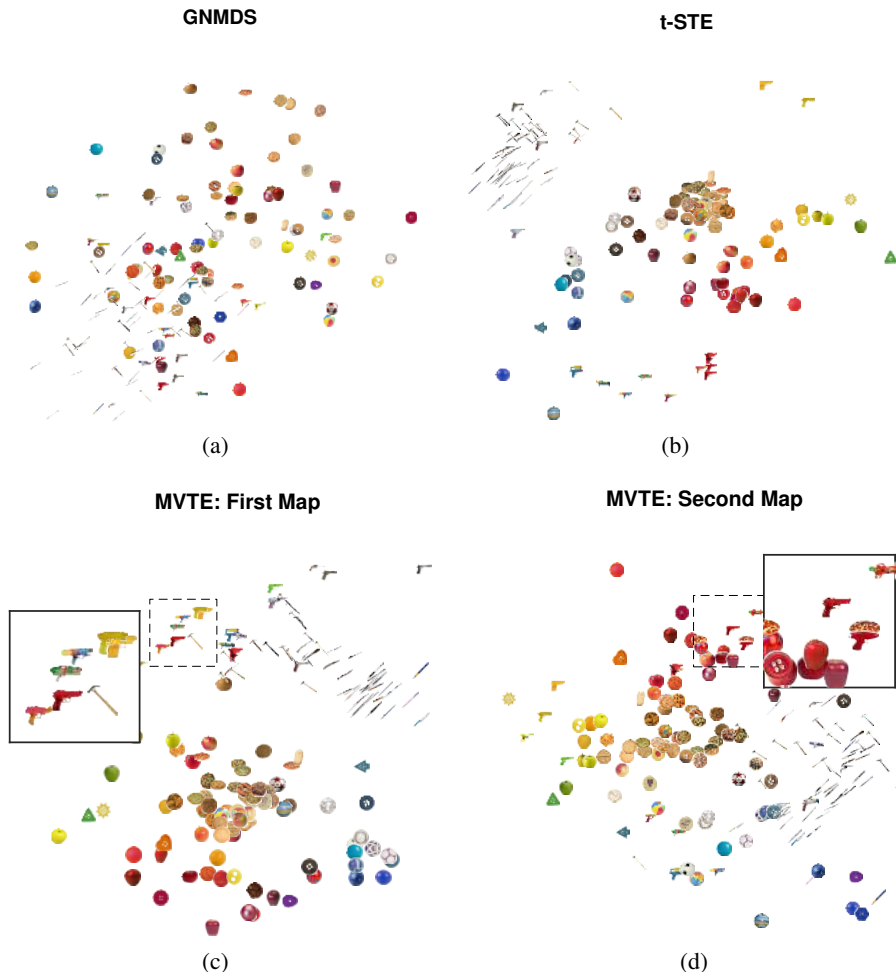
Figure 4. Results of the different algorithms on the subset of objects images dataset.

Query difficulty: Using multiple different maps we can estimate *the difficulty of a query* $(i|j,k)$; a property which might not be possible to measure using the methods producing a single map only. By difficulty, we mean the possibility of having different interpretations for a given query. As outlined in the Introduction, in certain situations, given a probe i and the test items j and k, both answers may be equally correct. Therefore, the workers might face difficulties when answering these queries. This property can be roughly formulated as follows. The query $(i|j,k)$ is considered *easy* if it is strongly satisfied in a consistent way in *all* the views, i.e, if the solution (i,j,k) is the correct answer no matter what attribute is considered. However, a query becomes *hard* if the solution (i,j,k) is strongly satisfied in at least one view while the (opposite) solution (i,k,j) it is strongly satisfied in the other view(s). Finally, a query is *ambiguous* if it is only (un)satisfied weakly in most of the views, making the answers only marginally different by any sense.

Item ambiguity: The availability of different views enables us to roughly estimate the ambiguity of the items, i.e., *the level of different interpretations that each item might have in distinct spaces*. We perform this by considering the *neighborhood* of each item in different maps. An item should be considered highly ambiguous if it has entirely dissimilar neighboring items in the maps. (See also (Cook et al., 2007).) In order to measure this quality, we calculate the *mean average precision* (MAP) (Manning et al., 2008) of each item with respect to the pairs of different views. MAP is calculated by first averaging the precision over different neighborhood sizes in a pair of maps, that is,

$$\text{avePrec} = \frac{\sum_{k=1}^{N} \text{prec}(k)}{N}, \qquad (10)$$

and then, finding the mean average precision of all pairs of maps. Here, $\text{prec}(k)$ denotes the precision by considering only the first k-nearest neighbors of the item in each map. A large MAP amounts to similar neighborhood structure in different maps and therefore, a less ambiguous item while a small MAP indicates a highly ambiguous item.
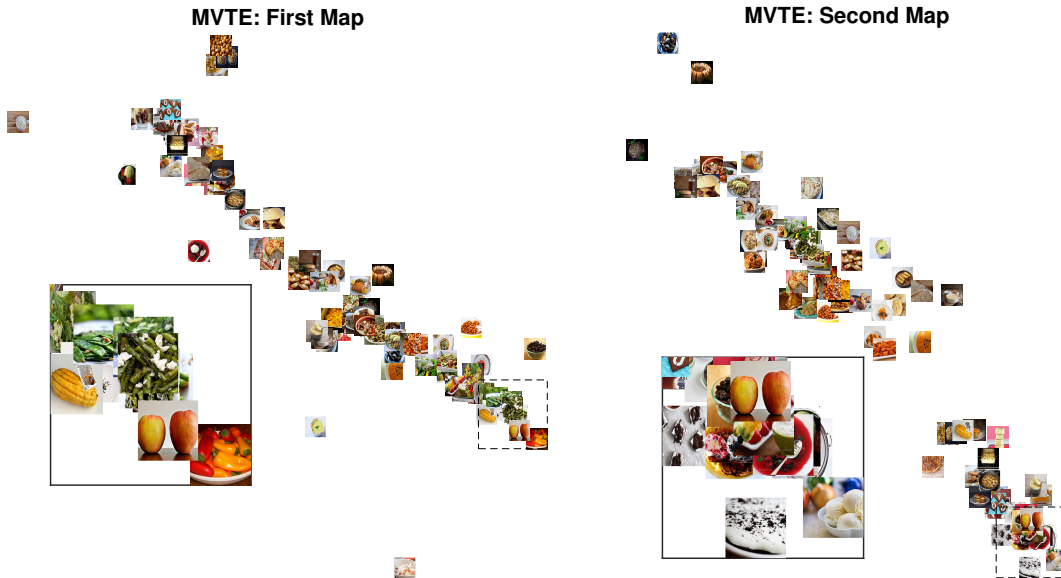
Figure 5. Results of the MVTE algorithm on the Food dataset with $M = 2$. The insets show that the picture of two apples can be placed both among naturally occurring things (e.g. vegetables, peppers, beans) as well as desserts that require preparation (e.g. ice cream, muffins). It is thus an *ambiguous item*.
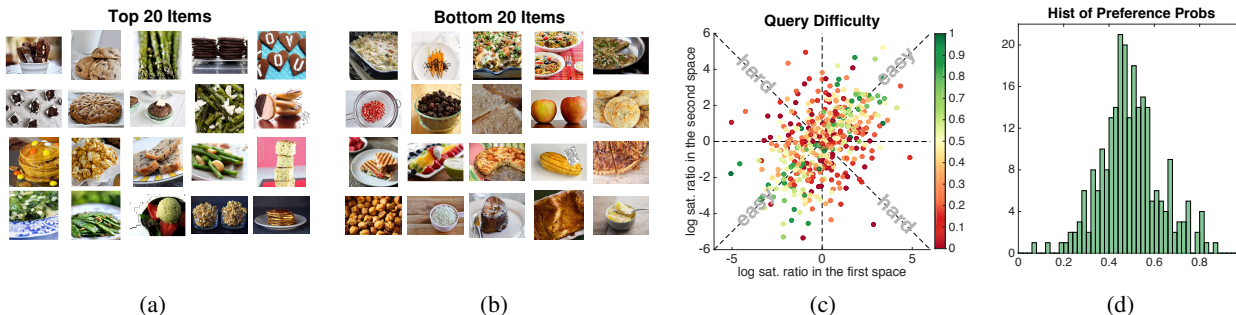

| (a) | (b) | (c) | (d) |

Figure 6. (a) Top and (b) bottom 20 items in the Food dataset in sense of MAP value in two maps, (c) query difficulty, (d) histogram of the preference probabilities of the workers towards the first map in the Food dataset.

Workers preferences towards different attributes:
Clearly, the workers may also have different levels of preferences towards the perceived attributes. It is reasonable to expect that all the workers provide consistent answers to the set of easy queries[4], since all the attributes tend to favor the same answer. However, they might provide different answers to hard queries based on their respective preferences.

4. Experimental Results

In this section, we compare the MVTE algorithm with methods that produce only a single map. In particular, we study the four applications outlined in the previous section. We conduct the experiments on a set of artificial as well as real-world datasets. Our MATLAB implementation of the

algorithm is publicly available online[5].

Multiple embeddings vs. higher dimensionality: We start by comparing the effect of having multiple maps with that of a single map having large number of dimensions. As an example, we consider triplets generated over a subspace of the Pima Indians Diabetes dataset (Smith et al., 1988). The original dataset contains 768 instances, each having 8 different measurements which belong to one of two different classes: ill or not-ill. We select three features, corresponding to three different measurements in the data: plasma glucose concentration, diastolic blood pressure, and 2-hour serum insulin. Each feature represents a different view for each instance. We generate 100 triplets for each datapoint in each view. The confusion matrix on the training signal and the generalization error[6] on a 10-fold cross-

---

[4]Unless the worker is spammer or malicious.

[5]https://github.com/eamid/mvte

[6]The generalization error is the ratio of the new triplets that are unsatisfied.

validation are shown in Figure 3(a) and Figure 3(b), respectively. The MVTE algorithm successfully divides the training triplets among three different maps, each corresponding to one of the views. Note that because of correlation between the views, each map also satisfies a portion of the triplets that originate from other views. Additionally, over 98% of the held-out triplets are satisfied using t-MVTE algorithm in at least one of the three maps (vs. 92% in a single map solution).

We perform similar experiments on three real-world datasets, namely Vogue Magazine Covers (Heikinheimo & Ukkonen, 2013), Food Images (Wilber et al., 2014), and Music Artists (Ellis et al., 2002), for all of which, the true numbers of views are unknown. For all the datasets, we do not make any assumptions about the provided triplets and thus, consider all of them as potentially correct. The results, illustrated in Figure 3(c) to 3(e), indicate that using multiple maps results in a lower generalisation error. Furthermore, the generalization error provides clues about the true number of views in each dataset. For instance, in the Vogue dataset, almost all the triplets are satisfied using three maps while the Music dataset requires a larger number of maps to obtain similar accuracy. This suggests that the users indeed considered a large number of different, subjective attributes when comparing the artists (Ellis et al., 2002).

Separating attributes from a mixture of triplets: Next, we consider the visualization results on two datasets. We first perform a visualization using a set of synthetic triplets, generated on a dataset of objects images (Konkle et al., 2010). The dataset contains 3400 images of objects from 200 different categories on a white background. We consider a subset of 156 objects out of 12 object categories. We form $M = 2$ spaces, corresponding to shapes and colors of the objects. We use Fourier descriptors (Gonzalez & Woods, 2006) for shape representation. For the color space, we concatenate the color histograms in the RGB channels (each with 16 bins in each channel) to form the color feature vectors. The results are shown in Figure 4. It can be seen that our method successfully projects the objects into two different maps corresponding to shape and color, while the other methods at best mix both attributes into a single map. The 'red toy-gun' is an example of an object with different neighborhood structures in two maps; in the first map, it appears among other toy-guns having different colors while in the second map, it is located among red objects, irrespective of their shape.

Finding ambiguous items: We continue by describing an experiment to find items that may have different interpretations. The Food dataset (Wilber et al., 2014) contains 250,320 triplets, queried on 100 images of different dishes of food. Figure 5 shows the results of the MVTE method

with $M = 2$. The first map represents 'types' of the dishes in which the neighborhood structure mainly corresponds to the attributes such as vegetable vs. non-vegetable, natural vs. processed, etc. On the other hand, the second map seems to be based on the 'taste', rather than type of the food. For instance, whether a food is sweet or not. The image of *two apples* is a good example of an item with very different neighborhood structures in the two maps. (See the insets in Figure 5.)

Figures 6(a) and 6(b) show the items having the top and bottom 20 MAP values in the Food dataset, respectively. The items having large values of MAP are those, such as green beans, muffins, pancakes, etc., which have a unique structural and semantic interpretations. However, the items at the bottom, e.g., apples, sweet corn, berries, etc., can be easily grouped in different categories, depending on the property considered.

Are queries with ambiguous items difficult? The level of ambiguity of items in a query also affects the difficulty of the query for workers. A query containing ambiguous items leads to several different interpretations, making it hard for the workers to decide on the answer. We show this by roughly estimating the difficulty of the queries by considering the *minimum of the MAP values of the items in each query as the level of difficulty*. Figure 6(c) illustrates the distribution of a number of 500 different triplets in the Food dataset, based on their respective satisfiability ratios in two different maps[7]. The color, on the other hand, indicates the minimum of the MAP values of the items in the query. Clearly, the difficulty level of the items, estimated using MAP values, is highly correlated with the difficulty of the queries, found by considering the satisfiability ratios in different maps. In other words, triplets having ambiguous items tend to appear mainly in regions which are expected to comprise hard triplets. Similar results hold for the easy queries having unambiguous items.

Do some workers prefer one attribute over another? To asses the performance of our approach for estimating the preferences of the workers, we conduct the following experiment. We first generate 10,000 queries on the xo-objects dataset (see Section 1) in each view (20,000 queries in total). To filter out the easy queries, we first compute the satisfiability ratios in the original spaces for each query. We mark the query $(i \,|\, j, k)$ as *hard* if: 1) the triplet (i, j, k) has a high satisfiability ratio in one view, the triplet (i, k, j) as a high satisfiability ratio in the other view, and 2) the difference between the satisfiability ratios in two views is comparatively small. The rest of the queries are marked as *easy*. Next, we generate a set of

---

[7]The triplets are plotted by reversing the order of the test items in a randomly selected subset. Please note that this procedure would not affect the difficulty of the query.

Table 1. Workers preferences probabilities in the synthetic example.

ID	1	2	3	4	5
p	0.004	0.274	0.471	0.483	0.490
p̃	0.038	0.318	0.500	0.472	0.545
ID	6	7	8	9	10
p	0.551	0.580	0.585	0.945	0.981
p̃	0.538	0.604	0.631	0.926	0.981

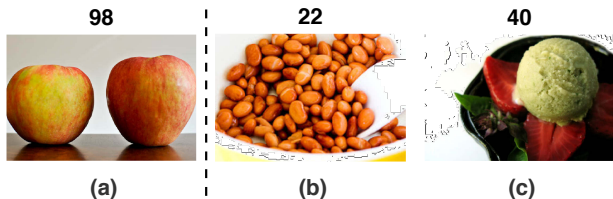**98**   **22**   **40**

(a)   (b)   (c)

Figure 7. An example of a hard query $(a \mid b, c)$ which has been answered differently by the workers having opposite preferences.

10 random workers, each having a different probability to prefer the shape space, $\{p_i\}_{i=1}^{10}$ (therefore, $\{(1 - p_i)\}_{i=1}^{10}$ for the color). That is, the ith worker provides her answer with reference to the shape space with probability p_i. For each query, 5 workers are chosen randomly to answer the task. All the easy queries are answered consistently by every worker. The hard queries are answered according to the preference probabilities.

We then run the MVTE with $M = 2$ on the set of triplets generated by the above workers. We repeat the same procedure to filter out the easy triplets by finding the satisfiability ratios in the maps. Finally, for each worker, we estimate the preference probabilities by finding the number of triplets satisfied in each view and normalizing the counts. Table 1 shows the true probabilities $\{p_i\}_{i=1}^{10}$ along with the estimated probabilities $\{\tilde{p}_i\}_{i=1}^{10}$ using our method. The estimated probabilities are very close to the true probabilities.

As a real-world example, we consider the workers of the Food dataset. We repeat the procedure for finding the hard triplets using the satisfiability ratios in the embeddings, as above and then, calculate the preference probabilities of the workers for each map. The histogram of the estimated preference probabilities of the workers towards the first map is shown in Figure 6(d). Most workers are neutral, having preference probabilities around 0.5, as expected. Figure 7 illustrates an example of a conflicting query, provided by the workers having highly different preferences. The first worker (#463 with $\tilde{p} = 0.64$), who is inclined towards the first map (types of items), selected the beans (b) as more similar to the apples, while the other worker (#188 with $\tilde{p} = 0.34$), who prefers the second view (taste), has chosen the ice cream (c). Note that the probe item (apples) appears among the ambiguous items in Figure 6(b), while the test items are relatively less ambiguous.

5. Conclusions

In this paper, we introduced a method to uncover multiple hidden attributes that can be used independently for making relative distance comparisons. We propose the MVTE method that successfully represents these attributes by finding a number of maps that correspond to the underlying attributes. The method provides a framework to estimate the true number of attributes, and to evaluate the difficulty of distance comparison tasks as well as the ambiguity of an object. Finally, it enables estimating the preferences of each worker towards different attributes based on the solutions provided by that worker. In general, the method can be seen as a means to learn a number of independent distance functions from a set of relative distance judgements. This may have applications beyond the crowdsourcing example considered in this paper.

References

Agarwal, Sameer, Wills, Josh, Cayton, Lawrence, Lanckriet, Gert, Kriegman, David, and Belongie, Serge. Generalized non-metric multidimensional scaling. In *AISTATS*, San Juan, Puerto Rico, 2007.

Belkin, Mikhail and Niyogi, Partha. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

Changpinyo, Soravit, Liu, Kuan, and Sha, Fei. Similarity component analysis. In *Advances in Neural Information Processing Systems 26*, pp. 1511–1519. 2013.

Cook, James, Sutskever, Ilya, Mnih, Andriy, and Hinton, Geoffrey E. Visualizing similarity data with a mixture of maps. In *International Conference on Artificial Intelligence and Statistics*, pp. 67–74, 2007.

Dawid, Alexander Philip and Skene, Allan M. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pp. 20–28, 1979.

Ellis, Daniel P. W., Whitman, Brian, Berenzweig, Adam, and Lawrence, Steve. The Quest for Ground Truth in Musical Artist Similarity. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR '02)*, pp. 170–177, Paris, France, October 2002.

Gomes, Ryan G, Welinder, Peter, Krause, Andreas, and Perona, Pietro. Crowdclustering. In *Advances in Neural Information Processing Systems*, pp. 558–566, 2011.

Gonzalez, Rafael C. and Woods, Richard E. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006. ISBN 013168728X.

Heikinheimo, Hannes and Ukkonen, Antti. The crowd-median algorithm. In *Conference on Human Computation and Crowdsourcing*. AAAI, 2013.

Konkle, Talia, Brady, Timothy F., Alvarez, George A., Oliva, Aude, Konkle, Talia, Brady, Timothy F., Oliva, Aude, and Brain, Department Of. Conceptual distinctiveness supports detailed visual long-term memory for realworld objects. *JEP:G*, pp. 149, 2010.

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

LeCun, Yann and Cortes, Corinna. The MNIST database of handwritten digits. 1999. URL http://yann.lecun.com/exdb/mnist/.

Manning, Christopher D., Raghavan, Prabhakar, and Schütze, Hinrich. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

Raykar, Vikas C. and Yu, Shipeng. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *J. Mach. Learn. Res.*, 13(1):491–518, February 2012. ISSN 1532-4435.

Saul, Lawrence K and Roweis, Sam T. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, 4: 119–155, 2003.

Smith, J W, Everhart, J E, Dickson, W C, Knowler, W C, and Johannes, R S. Using the adap learning algorithm to forecast the onset of diabetes mellitus. *Johns Hopkins APL Technical Digest*, 10:262–266, 1988.

Tamuz, Omer, Liu, Ce, Belongie, Serge, Shamir, Ohad, and Kalai, Adam T. Adaptively Learning the Crowd Kernel. In *ICML*, 2011.

Tenenbaum, Joshua B, De Silva, Vin, and Langford, John C. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

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

Whitehill, Jacob, Ruvolo, Paul, Wu, Tingfan, Bergsma, Jacob, and Movellan, Javier R. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems 22*, pp. 2035–2043, 2009.

Wilber, Michael, Kwak, Sam, and Belongie, Serge. Cost-effective hits for relative similarity comparisons. In *Human Computation and Crowdsourcing (HCOMP)*, Pittsburgh, November 2014.